

Benutzer- und Referenzhandbuch



Copyright © 1998–2012, Altova GmbH. All rights reserved. Use of this software is governed by and subject to an Altova software license agreement. XMLSpy, MapForce, StyleVision, SemanticWorks, SchemaAgent, UModel, DatabaseSpy, DiffDog, Authentic, AltovaXML, MissionKit, and ALTOVA as well as their logos are trademarks and/or registered trademarks of Altova GmbH. Protected by US Patent 7,200,816.

XML, XSL, XHTML, and W3C are trademarks (registered in numerous countries) of the World Wide Web Consortium; marks of the W3C are registered and held by its host institutions, MIT, INRIA, and Keio. UNICODE and the Unicode Logo are trademarks of Unicode Inc. This software contains 3rd party software or material that is protected by copyright and subject to other terms and conditions as detailed on the Altova website at http://www.altova.com/legal_3rdparty.html

ALTOVA®

Altova StyleVision 2012 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: 2012

© 2012 Altova GmbH

Inhaltsverzeichnis

1	Altova StyleVision 2012	3
2	Informationen zu dieser Dokumentation	6
3	Neue Funktionalitäten	10
3.1	Version 2011	11
3.2	Version 2010	13
4	Einführung	18
4.1	Was ist ein SPS?	19
4.2	Produktfeatures	22
4.3	Terminologie	24
4.4	Einrichten von StyleVision	28
4.5	Authentic-Ansicht in Altova-Produkten	29
5	Benutzeroberfläche	32
5.1	Hauptfenster	33
5.1.1	Design-Ansicht	35
5.1.2	Authentic-Ansicht	36
5.1.3	Ausgabeansichten	38
5.2	Seitenleisten	39
5.2.1	Design-Übersicht	43
5.2.2	Schema-Struktur	46
5.2.3	Design-Struktur	48
5.2.4	Globale Stile	52
5.2.5	Stil	55
5.2.6	Eigenschaften	57
5.2.7	Projekt	62
6	Schnellstart-Tutorial	66

6.1	Erstellen und Einrichten eines neuen SPS	67
6.2	Einfügen von dynamischem Inhalt (aus der XML-Quelle)	71
6.3	Einfügen von statischem Inhalt	78
6.4	Formatieren des Inhalts	82
6.5	Verwenden von automatischen Berechnungen	88
6.6	Verwendung von Bedingungen	92
6.7	Verwendung von globalen Vorlagen und "Restlicher Inhalt"	99
6.8	Fertig!	103

7 Übersicht über die Verwendung 106

7.1	SPS und Quellen	107
7.2	Erstellen des Designs	108
7.3	XSLT- und XPath-Versionen	109
7.4	Kompatibilität mit Internet Explorer	110
7.5	SPS und Authentic-Ansicht	112
7.6	Synchronisieren von StyleVision und Authentic	114
7.7	Generierte Dateien	115
7.8	Projekte in StyleVision	117
7.9	Kataloge in StyleVision	123

8 SPS-Dateien: Inhalt 128

8.1	Einfügen von XML-Inhalt als Text	129
8.1.1	Einfügen von Inhalt mit einem vordefinierten Format	132
8.1.2	Hinzufügen von Elementen in der Authentic-Ansicht	133
8.1.3	Restlicher Inhalt	136
8.2	Benutzerdefinierte Vorlagen	137
8.3	Benutzerdefinierte Elemente, XML-Textblöcke	140
8.3.1	Benutzerdefinierte Elemente	141
8.3.2	Benutzerdefinierte XML-Textblöcke	143
8.4	Tabellen	144
8.4.1	Statische Tabellen	146
8.4.2	Dynamische Tabellen	148
8.4.3	Tabellen in der Design-Ansicht	153
8.4.4	Formatieren von Tabellen	155
8.4.5	Anzeigen von Zeilen und Spalten	159
8.4.6	CALS/HTML-Tabellen	161
8.5	Listen	165

8.5.1	Statische Listen	166
8.5.2	Dynamische Listen	168
8.6	Grafiken	171
8.6.1	Bilder: URIs und Inline-Daten	172
8.6.2	Bildtypen und Ausgabe	175
8.6.3	Beispiel: Eine Vorlage für Grafiken	178
8.7	Formularsteuerelemente	179
8.7.1	Eingabefelder, mehrzeilige Eingabefelder	182
8.7.2	Kontrollkästchen	183
8.7.3	Auswahllisten	185
8.7.4	Optionsfelder, Schaltflächen	189
8.8	Links	190
8.9	Barcodes	191
8.10	Layout-Module	195
8.10.1	Layout Container	196
8.10.2	Layout-Felder	200
8.10.3	Linien	204
8.11	Die Funktion "Ändern in"	206

9 SPS-Datei: Struktur 210

9.1	Schemaquellen	212
9.1.1	DTDs und XML-Schemas	215
9.1.2	Datenbankschemas	220
9.1.3	Benutzerdefinierte Schemas	221
9.2	Zusammenführen von XML-Daten aus mehreren Quellen	223
9.3	Modulare SPSs	226
9.3.1	Verfügbare Modulobjekte	228
9.3.2	Erstellen eines modularen SPS	231
9.3.3	Beispiel: Ein Adressbuch	236
9.4	Vorlagen und Design-Fragmente	241
9.4.1	Hauptvorlage	242
9.4.2	Globale Vorlagen	243
9.4.3	Benutzerdefinierte Vorlagen	248
9.4.4	Variablenvorlagen	251
9.4.5	Node-Vorlagenoperationen	252
9.4.6	Design-Fragmente	256
9.5	XSLT-Vorlagen	259
9.6	Mehrfach-Dokumentausgabe	261
9.6.1	Einfügen einer Vorlage für ein neues Dokument	262

9.6.2	Vorlagen für neue Dokumente und Design-Struktur	263
9.6.3	URLs von Vorlagen für neue Dokumente	264
9.6.4	Vorschaudateien und Ausgabedokumentdateien	267
9.6.5	Dokumenteigenschaften und Stile	269

10 SPS-Datei: Komplexe Funktionen 272

10.1	Automatische Berechnungen	273
10.1.1	Bearbeiten und Verschieben von automatischen Berechnungen	274
10.1.2	Aktualisieren von Nodes mit automatischen Berechnungen	277
10.1.3	Automatische Berechnungen auf Basis von aktualisierten Nodes	280
10.1.4	Beispiel: Eine Rechnung	282
10.2	Bedingungen	286
10.2.1	Definieren der Bedingungen	287
10.2.2	Bearbeiten von Bedingungen	290
10.2.3	Ausgabe-basierte Bedingungen	291
10.2.4	Bedingungen und automatische Berechnungen	293
10.3	Bedingte Präsenz	294
10.4	Gruppieren	296
10.4.1	Beispiel: Group-By (Persons.sps)	299
10.4.2	Beispiel: Group-By (Scores.sps)	301
10.5	Sortieren	305
10.5.1	Funktionsweise	306
10.5.2	Beispiel: Sortieren nach mehreren Sortierschlüsseln	308
10.6	Parameter und Variablen	311
10.6.1	Vom Benutzer deklarierte Parameter	312
10.6.2	Parameter für Design-Fragmente	314
10.6.3	SPS Parameter für Quelldokumente	317
10.6.4	Variablen	318
10.7	Inhaltsverzeichnis, Referenzierung, Lesezeichen	320
10.7.1	Markieren von Elementen für das Inhaltsverzeichnis	323
	<i>Strukturieren des Designs in Ebenen.....</i>	325
	<i>Erstellen von Inhaltsverzeichnismarkierungen.....</i>	328
10.7.2	Erstellen der Inhaltsverzeichnis-Vorlage	331
	<i>Ebenenreferenzen in der Inhaltsverzeichnis-Vorlage.....</i>	333
	<i>Inhaltsverzeichnisreferenz: Name, Geltungsbereich, Hyperlink.....</i>	334
	<i>Formatieren von Inhaltsverzeichniseinträgen.....</i>	335
10.7.3	Beispiel: Einfaches Inhaltsverzeichnis	336
10.7.4	Beispiel: Hierarchische und sequenzielle Inhaltsverzeichnisse	340
10.7.5	Automatische Nummerierung	344

10.7.6	Textreferenzen	348
10.7.7	Lesezeichen und Hyperlinks	350
	<i>Einfügen von Lesezeichen</i>	351
	<i>Definieren von Hyperlinks</i>	354
10.7.8	Beispiel: Mehrere Sprachen	358
11	SPS-Datei: Darstellung	362
11.1	Vordefinierte Formate	363
11.2	Output Escaping	365
11.3	Werteformatierung (Formatieren von numerischen Datentypen)	367
11.3.1	Funktionsweise der Werteformatierung	368
11.3.2	Syntax der Werteformatierung	371
11.4	Arbeiten mit CSS-Stilen	378
11.4.1	Externe Stylesheets	380
11.4.2	Globale Stile	384
11.4.3	Lokale Stile	388
11.4.4	Definieren von Stilwerten	390
11.4.5	Stileigenschaften über XPath	393
11.4.6	Kombistile	396
11.5	Flexibles Zuweisen von Textstilen in Authentic	399
11.5.1	Kombistile	400
11.5.2	RichEdit	402
11.5.3	Textstatus-Symbole	405
11.6	Erstellen der Druckausgabe	408
11.6.1	Dokumentabschnitte	410
	<i>Dokumentanfangsabschnitt</i>	412
	<i>Kopf- und Fußzeilen: Teil 1</i>	415
	<i>Kopf- und Fußzeilen: Teil 2</i>	418
11.6.2	Zusammenhalten und Umbrechen	421
11.6.3	Pixel-Auflösung	422
12	SPS-Datei: Zusätzliche Funktionen	426
12.1	Globale Altova-Ressourcen	427
12.1.1	Definieren globaler Ressourcen	428
	<i>Dateien</i>	431
	<i>Ordner</i>	434
	<i>Datenbanken</i>	435
	<i>Kopieren von Konfigurationen</i>	437

12.1.2	Verwendung globaler Ressourcen	438
	<i>Zuweisen von Dateien und Ordnern</i>	439
	<i>Zuweisen von Datenbanken</i>	442
	<i>Wechseln von Konfigurationen</i>	443
12.2	Authentic Node-Eigenschaften	444
12.3	Parent Node bei Klick ersetzen durch	446
12.4	Zusätzliche Validierung	449
12.5	Neu von XSLT-, XSL-FO- oder FO-Datei	451
12.6	Benutzerdefinierte XPath-Funktionen	454
12.6.1	Definieren einer XPath-Funktion	457
12.6.2	Funktionen zum Auffinden von Nodes wiederverwenden	460
12.6.3	Parameter in XPath-Funktionen	461
	<i>Parameter und Sequenzen</i>	463
	<i>Parameter und Nodes</i>	468
12.7	Arbeiten mit Datumswerten	470
12.7.1	Verwendung der Datumswahl	471
12.7.2	Formatieren von Datumswerten	473
12.8	Verwenden von Scripts	476
12.8.1	Definieren von JavaScript-Funktionen	478
12.8.2	Zuweisen von Funktionen als Event Handler	479
12.8.3	Externe JavaScript-Dateien	480
12.9	HTML-Import	482
12.9.1	Erstellen eines neuen SPS anhand einer HTML-Datei	483
12.9.2	Erstellen des Schemas und SPS Designs	485
12.9.3	Erstellen von Tabellen und Listen als Elemente/Attribute	487
12.9.4	Generieren der Ausgabedateien	489
12.10	ASPX-Schnittstelle für Web-Applikationen	490
12.10.1	Beispiel: Lokaler Host auf Windows 7	492
12.11	PXF-Datei: Behälter für SPS und dazugehörige Dateien	494
12.11.1	Erstellen einer PXF-Datei	495
12.11.2	Bearbeiten einer PXF-Datei	498
12.11.3	Bereitstellen einer PXF-Datei	500

13 SPS-Dateien und Datenbanken 502

13.1	Datenbanken und StyleVision	503
13.2	Herstellen einer Verbindung zu einer Datenbank	505
13.2.1	Verbindungsassistent	507
13.2.2	ADO-Verbindungen	511
13.2.3	ODBC-Verbindungen	518

13.2.4	JDBC-Verbindungen	521
13.2.5	Globale Ressourcen	525
13.3	Auswählen der Datenbankdaten	526
13.3.1	Nicht-XML-Datenbanken	527
13.3.2	XML-Datenbanken	534
13.4	Das DB-Schema und DB-XML-Dateien	537
13.5	DB-Filter: Filtern von Datenbankdaten	540
13.6	SPS Design Features für Datenbanken	546
13.7	Generieren von Ausgabedateien	550
13.8	Datenbank abfragen	552
13.8.1	Datenquellen	554
13.8.2	Browser-Fenster: Anzeigen der DB-Objekte	556
13.8.3	Abfragefenster: Beschreibung und Funktionen	560
13.8.4	Abfragefenster: Arbeiten mit Abfragen	564
13.8.5	Ergebnisse und Meldungen	565

14 Authentic-Ansicht 568

14.1	Authentic-Ansicht	569
14.1.1	Übersicht über die Benutzeroberfläche	570
14.1.2	Authentic-Ansicht Symbolleistenschaltflächen	571
14.1.3	Authentic-Ansicht Hauptfenster	573
14.1.4	Authentic-Ansicht Kontextmenüs	576
14.2	Bearbeitung in der Authentic-Ansicht	579
14.2.1	Grundlegendes zur Bearbeitung	580
14.2.2	Tabellen in der Authentic-Ansicht	585
	<i>SPS-Tabellen.....</i>	<i>586</i>
	<i>CALS/HTML-Tabellen.....</i>	<i>588</i>
	<i>Schaltflächen zum Bearbeiten von CALS/HTML-Tabellen.....</i>	<i>592</i>
14.2.3	Bearbeiten einer DB	595
	<i>Navigieren in einer DB-Tabelle.....</i>	<i>596</i>
	<i>DB-Abfragen.....</i>	<i>597</i>
	<i>Ändern einer DB-Tabelle.....</i>	<i>602</i>
14.2.4	Arbeiten mit Datumsangaben	604
	<i>Datumswahl.....</i>	<i>605</i>
	<i>Texteintrag.....</i>	<i>606</i>
14.2.5	Definieren von Entities	607
14.2.6	Grafiken in der Authentic-Ansicht	609
14.2.7	Verwendung von Tasten in der Authentic-Ansicht	610

15	Authentic-Skripterstellung	612
15.1	Skript-Editor	615
15.2	Makros	616
15.2.1	Macros an Design-Elementen	617
15.2.2	Makros an Symbolleisten-Schaltflächen	619
15.2.3	Makros an Kontextmenübefehlen	621
15.3	Event Handler	622
15.4	Optionen für die Skripterstellung	623
16	Automatisierte Verarbeitung	626
16.1	Befehlszeilenschnittstelle: StyleVisionBatch	628
16.1.1	StyleVisionBatch Syntax	629
16.1.2	StyleVisionBatch Beispiele	632
16.2	Verwendung von AltovaXML	636
16.2.1	XSLT 1.0 CLI-Transformationen	637
16.2.2	XSLT 2.0 CLI-Transformationen	639
16.3	Automatisieren mit geplanten Tasks (Windows XP)	641
16.3.1	Erstellen von Batch-Dateien	642
16.3.2	Automatisieren mit geplanten Tasks (Windows XP)	643
16.3.3	Automatisieren mit geplanten Tasks (Windows Vista)	646
17	StyleVision in Visual Studio	654
17.1	Installieren des StyleVision Plugin	655
17.2	Unterschiede zur StyleVision Standalone-Version	656
18	StyleVision in Eclipse	658
18.1	Installation des StyleVision Plugin für Eclipse	659
18.2	Einstiegspunkte für Stylevision in Eclipse	663
19	Referenz	670
19.1	Symbolleisten	671
19.1.1	Tabelle	674
19.1.2	Authentic	676
19.1.3	RichEdit	678

19.1.4	Design-Elemente einfügen	679
19.1.5	Design-Filter	682
19.1.6	Globale Ressourcen	683
19.1.7	Standard	684
19.2	Design-Ansicht	686
19.2.1	Symbole	687
19.2.2	XPath-Ausdruck bearbeiten	691
19.3	Menü "Datei"	694
19.3.1	Neu	695
19.3.2	Öffnen, Neu laden, Schließen, Alle schließen	702
19.3.3	Design speichern, Alles speichern	703
19.3.4	Speichern unter	708
19.3.5	XML-Authentic-Daten speichern, Speichern unter	711
19.3.6	Generierte Dateien speichern	712
19.3.7	Web Design	714
19.3.8	Eigenschaften	715
19.3.9	Druckvorschau, Drucken	718
19.3.10	Zuletzt verwendete Dateien, Beenden	719
19.4	Menü "Bearbeiten"	720
19.4.1	Rückgängig, Wiederherstellen, Alles markieren	721
19.4.2	Suchen, Weitersuchen, Ersetzen	722
19.4.3	Stylesheet-Parameter	724
19.4.4	Markup reduzieren/erweitern	725
19.5	Menü "Projekt"	726
19.5.1	Neues Projekt, Projekt öffnen, Projekt neu laden	728
19.5.2	Projekt schließen, Projekt speichern	729
19.5.3	Dateien / globale Ressource / URL zu Projekt hinzufügen	730
19.5.4	Aktive (und verwandte) Dateien zu Projekt hinzufügen	732
19.5.5	Projekt und externe Verzeichnisse zu Projekt hinzufügen	733
19.6	Menü "Ansicht"	736
19.6.1	Symbolleisten und Statusleiste	737
19.6.2	Design-Seitenleisten	738
19.6.3	Design-Filter, Vergrößern/Verkleinern	739
19.7	Menü "Einfügen"	740
19.7.1	Inhalt	741
19.7.2	Restlicher Inhalt	742
19.7.3	RichEdit	743
19.7.4	Formular-Steuerelemente	744
19.7.5	DB-Steuerelement	745
19.7.6	Automatische Berechnung	746

19.7.7	Datumswahl	748
19.7.8	Absatz, Spezieller Absatz	749
19.7.9	Barcode	750
19.7.10	Bild	751
19.7.11	Horizontale Linie	752
19.7.12	Tabelle	753
19.7.13	Nummerierung und Aufzählungszeichen	754
19.7.14	Lesezeichen	757
19.7.15	Hyperlink	758
19.7.16	Bedingung, Ausgabe-basierte Bedingung	760
19.7.17	Vorlage	762
19.7.18	Benutzerdefinierte Vorlage	763
19.7.19	Variablenvorlage	764
19.7.20	Design Fragment	765
19.7.21	Layout Container, Layout-Feld, Linie	766
19.7.22	Inhaltsverzeichnis	767
19.7.23	Neues Dokument	768
19.7.24	Seite / Spalte / Dokumentabschnitt	769
19.7.25	Benutzerdefiniertes Element	770
19.8	Menü "Einschließen in"	771
19.8.1	Vorlage	772
19.8.2	Benutzerdefinierte Vorlage	773
19.8.3	Variablenvorlagen	774
19.8.4	Absatz, Spezieller Absatz	775
19.8.5	Nummerierung und Aufzählungszeichen	776
19.8.6	Lesezeichen und Hyperlinks	777
19.8.7	Bedingung, ausgabebasierte Bedingung	778
19.8.8	Inhaltsverz.-Markierungen und Inhaltsverz.-Ebenen	780
19.8.9	Neues Dokument	781
19.8.10	Benutzerdefiniertes Element	782
19.9	Menü "Tabelle"	783
19.9.1	Tabelle einfügen, Tabelle löschen	784
19.9.2	Kopf-/Fußzeilen für Tabelle einfügen	785
19.9.3	Zeile/Spalten anhängen, einfügen	786
19.9.4	Zeile, Spalte löschen	787
19.9.5	Zelle links, rechts, unten, oben verbinden	788
19.9.6	Zelle horizontal oder vertikal teilen	789
19.9.7	Zellumrandung / Tabellen-Markup-Code anzeigen	790
19.9.8	Tabelleneigenschaften	791
19.9.9	CALS/HTML-Tabellen bearbeiten	792
19.9.10	Vertikale Ausrichtung des Zelleninhalts	793

19.10 Menü "Authentic"	794
19.10.1 Authentic Skripts bearbeiten	795
19.10.2 Benutzerdefinierte Symbolleisten-Schaltflächen	796
19.10.3 Makro-Referenzen überprüfen	800
19.10.4 Datumswahl automatisch hinzufügen	801
19.10.5 DB-Steuerelemente automatisch hinzufügen	802
19.10.6 Authentic-Ansicht neu laden, XML validieren	803
19.10.7 Neue Zeile mit XML-Daten für Bearbeitung auswählen	804
19.10.8 XML Entities definieren	805
19.10.9 Markup-Code anzeigen	806
19.10.10 RichEdit	807
19.10.11 (Dynamische Tabelle) Zeilenbefehle	808
19.11 Datenbank	809
19.11.1 Datenbank abfragen	810
19.11.2 DB-Filter bearbeiten, DB-Filter löschen	811
19.12 Menü "Eigenschaften"	812
19.12.1 Nummerierung und Aufzählungszeichen bearbeiten	813
19.12.2 Vordefinierte Wertformatierungsstrings	814
19.13 Menü "Extras"	816
19.13.1 Rechtschreibung	817
19.13.2 Rechtschreibung - Optionen	819
19.13.3 Globale Ressourcen	822
19.13.4 Aktive Konfiguration	823
19.13.5 Anpassen	824
19.13.6 Symbolleisten und Fenster wiederherstellen	829
19.13.7 Optionen	830
19.14 Menü "Fenster"	832
19.15 Menü "Hilfe"	833
19.15.1 Inhaltsverzeichnis, Index, Suchen	834
19.15.2 Aktivierung, Bestellformular, Registrieren, Updates	835
19.15.3 Weitere Befehle	837

20 Referenz für Programmierer 840

20.1 Skripterstellung	841
20.1.1 Übersicht	843
<i>Die Benutzeroberfläche des Skript-Editors</i>	845
<i>Komponenten eines Skripting-Projekts</i>	849
20.1.2 Globale Deklarationen	851
20.1.3 Formulare	852

	<i>Erstellung eines neuen Formulars</i>	853
	<i>Formulardesign und Formularobjekte</i>	855
	<i>Formular-Events</i>	858
20.1.4	Events	860
20.1.5	Makros	861
	<i>Erstellen und Bearbeiten eines Makros</i>	862
	<i>Ausführen eines Makros</i>	863
	<i>Debuggen eines Makros</i>	864
20.1.6	Anmerkungen zur Programmierung	865
	<i>Vordefinierte Befehle</i>	867
20.2	Applikations-API	874
20.2.1	Übersicht	876
	<i>Objektmodell</i>	877
	<i>Programmiersprachen</i>	878
	<i>DOM und XML Data</i>	898
	<i>Nicht mehr verwendete Zeilenoperationen in der Authentic-Ansicht</i>	901
20.2.2	Schnittstellen	902
	<i>Application</i>	903
	<i>AppOutputLine</i>	908
	<i>AppOutputLines</i>	913
	<i>AppOutputLineSymbol</i>	915
	<i>AuthenticContextMenu</i>	917
	<i>AuthenticEventContext</i>	919
	<i>AuthenticRange</i>	922
	<i>AuthenticView</i>	950
	<i>Document</i>	968
	<i>Documents</i>	976
	<i>Parameter</i>	978
	<i>Parameters</i>	979
	<i>SchemaSource</i>	980
	<i>SchemaSources</i>	983
	<i>StyleSheet</i>	985
	<i>XMLData</i>	986
20.2.3	Enumerationen	997
	<i>ENUMApplicationStatus</i>	998
	<i>ENUMAppOutputLine_Severity</i>	999
	<i>ENUMAppOutputLine_TextDecoration</i>	1000
	<i>ENUMSchemaSourceType</i>	1001
	<i>ENUMSchemaType</i>	1002
20.2.4	Applikations-API für Java	1003
	<i>Beispielquellcode</i>	1005

<i>SpyApplication</i>	1007
<i>SpyCodeGeneratorDlg</i>	1008
<i>SpyDatabaseConnection</i>	1009
<i>SpyDialogs</i>	1010
<i>SpyDoc</i>	1011
<i>SpyDocuments</i>	1013
<i>SpyDTDSchemaGeneratorDlg</i>	1014
<i>SpyElementList</i>	1015
<i>SpyElementListItem</i>	1016
<i>SpyExportSettings</i>	1017
<i>SpyFileSelectionDlg</i>	1018
<i>SpyFindInFilesDlg</i>	1019
<i>SpyFindInFilesMatch</i>	1020
<i>SpyFindInFilesResult</i>	1021
<i>SpyFindInFilesResults</i>	1022
<i>SpyGenerateSampleXMLDlg</i>	1023
<i>SpyGridView</i>	1024
<i>SpyProject</i>	1025
<i>SpyProjectItem</i>	1026
<i>SpyProjectItems</i>	1027
<i>SpySchemaDocumentationDlg</i>	1028
<i>SpyTextImportExportSettings</i>	1030
<i>SpyTextView</i>	1031
<i>SpyWSDL20DocumentationDlg</i>	1032
<i>SpyWSDLDocumentationDlg</i>	1034
<i>SpyXBRLDocumentationDlg</i>	1036
<i>SpyXMLData</i>	1038
<i>Authentic</i>	1039
<i>Vordefinierte Konstanten</i>	1042
20.3 ActiveX Integration	1051
20.3.1 Integration auf Applikationsebene	1052
<i>Beispiel: HTML</i>	1053
20.3.2 Integration auf Dokumentebene	1056
<i>Verwendung von StyleVisionControl</i>	1057
<i>Verwendung von StyleVisionControlDocument</i>	1058
<i>Verwendung von StyleVisionControlPlaceHolder</i>	1059
<i>Abfragen von StyleVision Befehlen</i>	1060
<i>Beispiele</i>	1061
20.3.3 Befehlstabelle für StyleVision	1064
<i>Menü "Datei"</i>	1065
<i>Menü "Bearbeiten"</i>	1067

	Menü "Projekt"	1068
	Menü "Ansicht"	1069
	Menü "Einfügen"	1070
	Menü "Einschließen in"	1073
	Menü "Tabelle"	1075
	Menü "Authentic"	1076
	Menü "Datenbank"	1077
	Menü "Eigenschaften"	1078
	Menü "Extras"	1079
	Menü "Fenster"	1080
	Menü "Hilfe"	1081
	Diverse andere Menübefehle	1082
20.3.4	Aufrufen von StyleVisionAPI	1101
20.3.5	Objektreferenz	1102
	StyleVisionCommand	1103
	StyleVisionCommands	1105
	StyleVisionControl	1106
	StyleVisionControlDocument	1112
	StyleVisionControlPlaceHolder	1118
	Enumerationen	1121

21 Anhänge 1124

21.1	XSLT-Prozessorinformationen	1125
21.1.1	XSLT 1.0-Prozessor: Implementierungsinformationen	1126
21.1.2	XSLT 2.0-Prozessor: Implementierungsinformationen	1128
	Allgemeine Informationen	1129
	XSLT 2.0-Elemente und -Funktionen	1131
21.1.3	XPath 2.0- und XQuery 1.0-Funktionen	1132
	Allgemeine Informationen	1133
	Unterstützung von Funktionen	1135
21.1.4	Erweiterungen	1139
	Java-Erweiterungsfunktionen	1140
	.NET-Erweiterungsfunktionen	1149
	MSXSL Scripts für XSLT	1155
	Altova Erweiterungsfunktionen	1158
21.2	Datentypen in XML-Schemas, die von DB generiert wurden	1164
21.2.1	MS Access	1165
21.2.2	MS SQL Server	1166
21.2.3	MySQL	1167
21.2.4	Oracle	1168

21.2.5	ODBC	1169
21.2.6	ADO	1170
21.2.7	Sybase	1171
21.3	Technische Daten	1172
21.3.1	OS- und Arbeitsspeicheranforderungen	1173
21.3.2	Altova XML Parser	1174
21.3.3	Altova XSLT- und XQuery-Prozessor	1175
21.3.4	Unicode-Unterstützung	1176
	<i>Windows XP</i>	1177
	<i>Rechts nach Links-Schreibrichtung</i>	1178
21.3.5	Internet-Verwendung	1179
21.4	Lizenzinformationen	1180
21.4.1	Electronic Software Distribution	1181
21.4.2	Software-Aktivierung und Lizenzüberwachung	1182
21.4.3	Rechte am geistigen Eigentum	1183
21.4.4	Altova Endbenutzer-Lizenzvereinbarung	1184

Index	1199

Kapitel 1

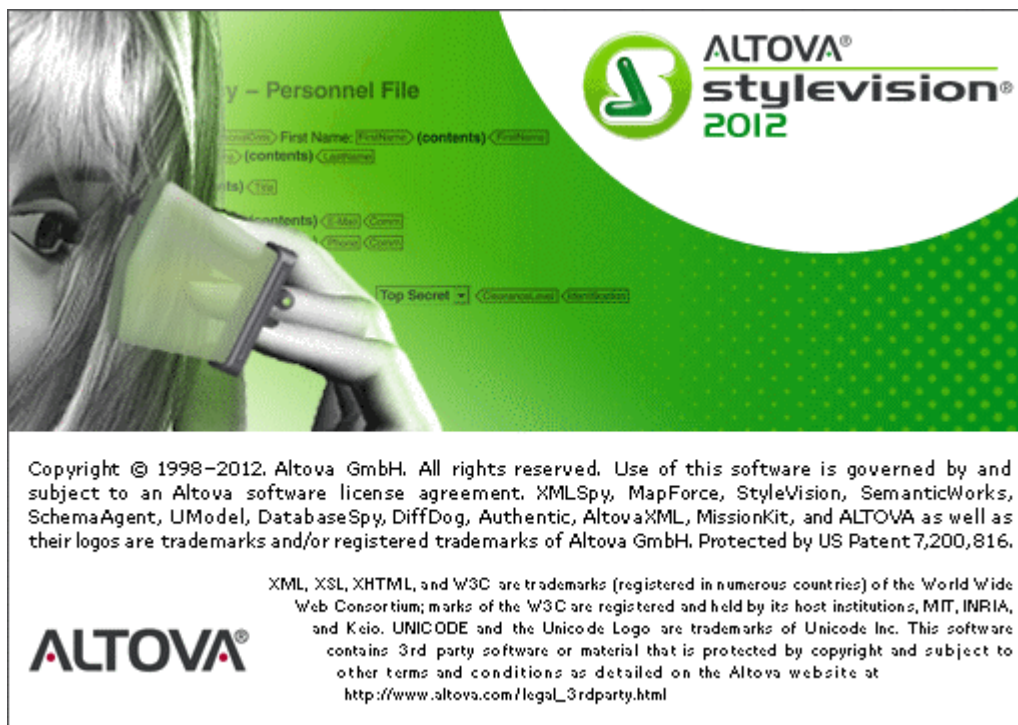
Altova StyleVision 2012

1 Altova StyleVision 2012

Die **Altova® StyleVision® 2012 Professional Edition** ist eine Applikation zum grafischen Erstellen und Bearbeiten von StyleVision Power Stylesheets und steht sowohl für die 64-Bit- als auch für die 32-Bit-Version zur Verfügung. 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 Bearbeiten von **Datenbanken (DB) über die Authentic-Ansicht** und zum Generieren von Datenbankberichten im HTML- und RTF-Format.
- zum Generieren von **XSLT Stylesheets** auf Basis des SPS Designs. (es wird sowohl XSLT 1.0 als auch XSLT 2.0 unterstützt.) Mit Hilfe der XSLT Stylesheets können Sie XML-Dokumente außerhalb von StyleVision in Ausgabedokumente wie z.B. HTML und RTF (von Textverarbeitungsprogrammen wie MS Word verwendetes Rich Text-Format) transformieren.
- zum Generieren von **HTML- und RTF- Dokumenten** anhand eines XML-Dokuments - und zwar direkt von StyleVision aus. Bei Datenbank-basierten SPSs kann StyleVision zusätzlich für jedes SPS ein XML-Schema generieren, das auf der Datenbank und einem XML-Instanzdokument basiert, die/das diesem Schema entspricht und Daten aus der Datenbank enthält.

Außerdem können Sie mit Hilfe von StyleVision ein HTML-Dokument importieren und anhand dessen ein XML-Dokument erstellen.



Altova Website:  [Stylesheet Designer](#), [XSLT Designer](#)

Letzte Aktualisierung: 02/17/2012

Kapitel 2

Informationen zu dieser Dokumentation

2 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](#), in der erklärt wird, was ein SPS ist und in der die wichtigsten Funktionen und Begriffe in StyleVision vorgestellt werden.
- eine [Beschreibung der Benutzeroberfläche](#) von StyleVision.
- ein [Tutorial](#), in dem die Funktionen von StyleVision anhand von praktischen Beispielen erklärt werden.
- [Übersicht über die Verwendung](#). 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](#). Hier wird beschrieben, wie statische (Stylesheet-basierte) und dynamische (XML-Dokument-basierte) Komponenten im SPS erstellt und bearbeitet werden.
- [Struktur der SPS-Datei](#). Hier wird beschrieben, wie ein SPS strukturiert und modularisiert werden kann. Außerdem wird die Behandlung von StyleVision-Vorlagen beschrieben.
- [SPS-Datei - Komplexe Funktionen](#). 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](#). Hier wird erklärt, wie Sie SPS Komponenten formatieren und ihnen ein Layout zuweisen.
- [Zusätzliche Funktionen für die SPS-Datei](#). 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, Scripts und Variablen und Parameter.
- [SPS-Dateien und Datenbanken](#). Hier wird erklärt, wie Sie Datenbanken mit SPSs verwenden können.
- [Authentic-Ansicht](#). In diesem Abschnitt wird beschrieben, wie XML-Dokumente in der Authentic-Ansicht bearbeitet werden. Die StyleVision Benutzeroberfläche enthält ein Register Authentic, auf dem Sie eine Vorschau der Ausgabe der Authentic-Ansicht erhalten.
- [Automatisierte Verarbeitung](#): Hier wird beschrieben, wie die Generierung von Ausgabedateien automatisiert werden kann.
- [Die Integrationsfunktionen von StyleVision](#). Hier finden Sie eine Dokumentation zur Integration von StyleVision in andere Applikationen. Außerdem finden Sie hier Informationen zur Verwendung von StyleVision in [Visual Studio](#) und [Eclipse](#).
- einen Abschnitt [Referenz](#) mit einer Beschreibung aller in StyleVision verwendeter Symbole und Befehle.
- [Anhänge](#), die Informationen über den Altova XSLT-Prozessor und die Konvertierung von DB-Datentypen in XML-Schema-Datentypen enthalten; technische Daten zu StyleVision; und Lizenzinformationen.

Anleitung zur Verwendung

Lesen Sie am besten zuerst die Abschnitte [Einführung](#), [Benutzeroberfläche](#) und [Übersicht über die Verwendung](#) um einen ersten Eindruck von den Funktionen und der Verwendung von

StyleVision zu gewinnen. Als nächstes finden Sie im [Tutorial](#) praktische Beispiele zur Erstellung eines SPS. In den folgenden Abschnitten zur SPS-Datei ([Inhalt von SPS-Dateien](#), [Struktur der SPS-Datei](#), [SPS-Datei - Komplexe Funktionen](#), [SPS-Datei - Darstellung](#), [Zusätzliche Funktionen für die SPS-Datei](#), [SPS-Dateien und Datenbanken](#)) wird ausführlich beschrieben, wie die verschiedenen Funktionen von StyleVision verwendet werden. Im Anschluss daran finden Sie im Abschnitt [Referenz](#) eine genaue Beschreibung aller Symbolleistenflächen, Design-Symbole und Menübefehle, geordnet nach Symbolleisten und Menü. In den Abschnitten [Authentic-Ansicht](#) und [Befehlszeilenschnittstelle: StyleVisionBatch](#) finden Sie Informationen über das Bearbeiten in der Authentic-Ansicht und das Aufrufen von StyleVision über die Befehlszeile.

Dateipfade in Windows XP, Windows Vista und Windows 7

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 in Windows XP entspricht in Windows Vista und Windows 7 dem Verzeichnis Dateien. Es befindet sich standardmäßig an den nachstehend angeführten Orten. Beispieldateien befinden sich im Normalfall in einem Unterverzeichnis des Verzeichnisses (Eigene) Dateien.

Windows XP	C: /Dokumente und Einstellungen/<Benutzername> /Eigene Dateien
Windows Vista, Windows 7	C: /Benutzer/<Benutzername>/Dateien

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

Windows XP	C: /Programme/Altova
Windows Vista, Windows 7	C: /Programme/Altova
32-Bit-Paket unter 64-Bit Windows Betriebssystem (XP, Vista, 7)	C: /Programme (x86) /Altova/

Support-Optionen

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

1. Lesen Sie in der [Hilfe](#) (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
- **DB:** Datenbank
- **CSS:** Cascading Style Sheets
- **FAQ:** (Frequently Asked Questions) häufig gestellte Fragen

Kapitel 3

Neue Funktionalitäten

3 Neue Funktionalitäten

Version 2012 Release 1

Im Folgenden sind die neuen Funktionen in [StyleVision](#) **Version 2012 Release 1** aufgelistet.

- [Kombistile](#): Mit Hilfe von Kombistilen können Sie mehrere CSS-Stildeklarationen zu einer einzigen Stilregel kombinieren und diese Kombistilregel auf die am häufigsten verwendeten Designkomponenten anwenden.
- [Wechseln zwischen externen CSS Stylesheets](#): Sie können wählen, ob alle in allen externen mit dem SPS verknüpften CSS Stylesheets (in kaskadierender Reihenfolge) vorhandenen Regeln verwendet werden sollen oder nur die Regeln in einem der externen CSS Stylesheets.
- [Neues SPS auf Basis eines SPS-Moduls](#): Eine neue SPS-Datei kann erstellt werden, die eine leere Hauptvorlage und die ausgewählte SPS-Datei als Modul enthält.
- [Suchen in Projekten](#): Ermöglicht die einfache Suche von Projektdateien und -ordnern mit Hilfe von im Namen enthaltenen Text-Strings.
- Unterstützung für [JDBC-Datenbankverbindungen](#).
- Neue [Application API-Schnittstelle für Java](#).

Version 2012 Release 2

Im Folgenden sind die neuen Funktionen in [StyleVision](#) **Version 2012 Release 2** aufgelistet.

- [RichEdit für die Authentic-Ansicht](#): Elemente im Design können als RichEdit-Komponenten erstellt werden, sodass der Benutzer Textfragmente in diesem Element in der Authentic-Ansicht markieren und mit Hilfe der RichEdit-Stileigenschaften der Authentic-Ansicht mit Stilen versehen kann. Mit Hilfe von RichEdit können die Schriftart (font), die Schriftbreite (font-style), der Schriftschnitt (font-style), die Schriftartverzierung (font-decoration), die Farbe (color), die Hintergrundfarbe und die Textausrichtung in der Authentic-Ansicht definieren.
 - [Parent Node bei Klick ersetzen durch](#). Der Wert des übergeordneten Nodes einer Schaltfläche oder eines Hyperlink kann vom Benutzer der Authentic-Ansicht ausgewählt werden. Das SPS kann so konzipiert werden, dass die Darstellung sich ändert, je nachdem, welchen Wert der Benutzer der Authentic-Ansicht auswählt.
- Fehlerbehebungen

3.1 Version 2011

Version 2011 Release 1

Im Folgenden sind die neuen Funktionen in [StyleVision](#) **Version 2011 Release 1** aufgelistet.

- [CALS/HTML-Tabellen](#): Die XML-Tabellenfunktion aus früheren Versionen wurde verbessert und in [CALS/HTML-Tabellen](#) umbenannt. Wenn eine CALS- oder HTML-Tabellenstruktur in der DTD bzw. dem Schema des SPS-Designs definiert ist, können Sie im Design festlegen, dass CALS/HTML-Tabellen verarbeitet werden sollen. Diese Tabellenstrukturen werden dann, falls sie in der XML-Instanzdatei vorkommen, korrekt als Tabellen an die Ausgabe gesendet.
- [Neu von XSLT-Datei](#): 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.
- [Integration in Microsoft Visual Studio 2010](#): Damit wird die Unterstützung für Visual Studio zusätzlich zu den Versionen 2005 und 2008 auf die neueste Version von Visual Studio ausgeweitet. Visual Studio 2003 wird nicht mehr unterstützt.
- [Integration in Eclipse 3.6](#): Damit wird die Unterstützung für Eclipse auf die neueste Eclipse Version ausweitert. Unterstützt werden zusätzlich dazu die Versionen 3.4 und 3.5. Eclipse 3.3 wird nicht mehr unterstützt.
- [Authentic Skripterstellung](#): Ermöglicht Flexibilität und Interaktivität in der Authentic-Ansicht.

Version 2011 Release 2

Im Folgenden sind die neuen Funktionen in [StyleVision](#) **Version 2011 Release 2** aufgelistet.

- [Mehrfach-Dokumentausrabe](#): 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 XPath-Funktionen](#): Benutzer können XPath-Funktionen definieren, die an jeder Stelle im Dokument, an der XPath-Funktionen erlaubt sind, verwendet werden können.
- [Bilder aus Inline-Daten](#): 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.
- [ASPX-Schnittstelle für Web-Applikationen](#): 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.
- [Auswahllisten](#): Die Design-Möglichkeiten für Auswahllisten wurden erweitert. Die sichtbaren Werte in der Dropdown-Liste der Auswahlliste sowie die sich in der XML-Datei befindenden zugehörigen Werte können separat angegeben werden.
- [Barcodes](#): Mit Hilfe dieser neuen Design-Komponente können Barcodes einfach in das Design eingefügt werden. Barcode-Bilder werden on the fly generiert und in die Ausgabedokumente eingefügt.
- [Bedingte Präsenz](#): Bestimme Design-Komponenten haben bedingte Präsenz als Eigenschaft. Eine bedingte Design-Komponente wird in der Ausgabe nur dann erstellt, wenn die für sie angegebene Bedingung erfüllt wird.

Im Folgenden sind die neuen Funktionen in [StyleVision](#) **Version 2011 Release 3** aufgelistet.

- [Kompatibilität mit Internet Explorer 9](#): StyleVision bietet Unterstützung für Internet

Explorer 9 (IE 9). CSS-Stile und HTML5 Elemente, die von IE 9 unterstützt werden, werden auch auf der SPS Design-Oberfläche unterstützt. In der Authentic-Ansicht und der HTML-Vorschau sehen Sie die IE 9-Darstellung des Ausgabedokuments. Sie können die IE-Kompatibilität des SPS jederzeit im [Dialogfeld "Eigenschaften"](#) ändern..

- [Verbesserte Unterstützung von Bildformaten](#): Bessere Unterstützung für die Formate TIFF, SVG und JPEG XR.
- [Unterstützung für PXF-Dateien](#): Das PXF-Dateiformat wurde von Altova speziell zum Verpacken eines SPS-Designs mit dazugehörigen Dateien (wie z.B. Schema-Datei, XML-Quelldatei, im Design verwendeten Bilddateien und XSLT-Dateien für die Transformation der XML-Quelldatei in ein Ausgabeformat) entwickelt. Der Vorteil des PXF-Dateiformats ist, dass alle für die Bearbeitung in der Authentic-Ansicht und die Generierung der Ausgabedatei anhand der Authentic-Ansicht erforderlichen Dateien in einer einzigen Datei weitergeleitet werden können.

3.2 Version 2010

Version 2010 Release 1

Im Folgenden sind die neuen Funktionen in [StyleVision Version 2010 Release 1](#) aufgelistet. Aufgrund einiger dieser neuen Funktionen musste einige Änderungen an der Verwendungsweise älterer Funktionen vorgenommen werden. In diesen Fällen verhält sich die Funktion wie bisher, es kommen aber ein oder mehrere neue Mechanismen zum Einsatz. In der folgenden Liste ist aufgeführt, wie eine neue Funktion sich auf bestehende Funktionen auswirkt.

- [Layout Container](#): Ein Layout Container ist ein Block, in dem Design-Elemente mit einem Layout versehen und an einer absoluten Stelle im Block positioniert werden können.
- [Grafikvorlagen](#): Sie können einen Layout Container mit einem Bild als Grafikvorlage für das Design hinterlegen. Mit Hilfe einer Grafikvorlage kann ein vorhandenes Design genau reproduziert werden.
- [Dokumentabschnitte](#): Dokumente können in Abschnitte aufgeteilt werden, wobei jeder Abschnitt eigene Eigenschaften wie z.B. Seitenlayouteigenschaften haben kann. Dadurch können unterschiedliche Teile des Dokuments unterschiedlich dargestellt werden. *Ältere dadurch betroffene Funktionen*: In früheren Designs gab es keine Abschnitte. Diese Designs werden nun als Dokumente mit einem Abschnitt, nämlich dem Anfangsabschnitt geöffnet. Die Seiteneigenschaften und die Seitenlayouteigenschaften, die vormals für das Dokument als Ganzes definiert wurden, gelten nun für den Anfangsabschnitt. Die [Umschlagseite](#) für die Druckausgabe früherer Versionen wird nun in der neuen Version als Vorlage im Anfangsdokumentabschnitt erstellt.
- [Seitenspalten](#): Es können nun Spalten für Seiten definiert werden.
- [Benutzerdefinierte Vorlagen](#): Eine Vorlage kann durch einen XPath-Ausdruck für eine Reihenfolge von Elementen generiert werden. Bei diesen Elementen kann es sich um atomare Werte oder Nodes handeln. Mit Hilfe eines XPath-Ausdrucks kann eine genauere Auswahl von Nodes erfolgen, da dadurch für die Auswahl Bedingungen und Filter verwendet werden können. Außerdem können Vorlagen für atomare Werte erstellt werden, sodass Strukturen erstellt werden können, die unabhängig von der Schemastruktur sind. *Ältere dadurch betroffene Funktionen*: Variableniteratoren, die früher zum Erstellen einer Vorlage für eine Variable verwendet wurden, erstellen nun eine Variable in einer Node-Vorlage und anschließend eine benutzerdefinierte Vorlage für diese Variable.
- [Benutzerdefinierte Elemente](#): Hauptsächlich dient diese Funktion dazu, Elemente aus Darstellungssprachen (wie HTML, XSLT und XSL-FO) jederzeit an jeder Stelle im Design einfügen zu können.
- [Benutzerdefinierte XML-Textblöcke](#): XML-Textblöcke können an jeder beliebigen Stelle im Design eingefügt werden. Diese Blöcke werden an dieser Stelle im generierten XSLT-Stylesheet erstellt.
- [XSLT-Vorlagen](#): In die generierten Stylesheets können XSLT-Dateien importiert werden. Wenn ein Node im XML-Instanzdokument einer Vorlage in der importierten XSLT-Datei entspricht und keine andere Vorlage Vorrang vor der importierten Vorlage hat, so wird die importierte Vorlage verwendet. Zusätzlich dazu können vom Design aus benannte Vorlagen in der importierten XSLT-Datei aufgerufen werden.
- [Variablen](#): In einer Vorlage kann nun eine Variable deklariert werden und einen Wert erhalten, der durch einen XPath-Ausdruck definiert wird. Früher bezog sich der Wert einer Variablen nur auf die Auswahl des Node, in dem sie erstellt wurde. In der Version 2010 kann nun jeder beliebige XPath-Ausdruck als Wert der Variablen definiert werden. *Ältere davon betroffene Funktionen*: Variablen und Variableniteratoren. Variablen aus älteren Versionen werden nun in der entsprechenden Vorlage erstellt und erhalten einen Wert, der dieselbe Vorlage auswählt. Variableniteratoren werden durch eine Kombination einer Variablen und einer benutzerdefinierten Vorlage ersetzt; siehe benutzerdefinierte Vorlagen weiter unten.

- [Einfügen von Design-Elementen](#): Design-Elemente (Absätze, Listen, Bilder, usw.) können zuerst eingefügt werden, sodass dem Design-Element erst nachher ein XML-Node aus der Schema-Struktur zugewiesen wird. Diese Funktion dient als Ergänzung zum bestehenden Mechanismus, bei dem Schema-Nodes in das Design gezogen werden und ein Design-Element dafür erstellt wird.
- [Ausblenden von Markup-Tags in der Design-Ansicht](#): Markup-Tags können in der Design-Ansicht ausgeblendet und reduziert werden, um mehr Platz in der Ansicht zu schaffen.
- [Deaktivieren der Ausgabe mit Escape](#): Eine Einstellung, mit der definiert wird, ob die Textausgabe mit Escape versehen wird oder nicht. Ein Zeichen wird mit Escape versehen, wenn es als Zeicheneinheit (wie z.B. `& amp;` oder `& #65;`) ausgegeben wird. Diese Funktion ist nützlich, wenn Text ausgegeben wird, der Programmcode enthält.
- [Pixel-Auflösung](#): Die Pixel-Längeneinheiten im SPS werden für die Druckausgabe gemäß einem vom SPS-Designer definierten Faktor, in absolute Längeneinheiten konvertiert.
- [Standardlängeneinheiten](#): Kann im Dialogfeld "Optionen" (**Extras | Optionen**) definiert werden.
- [XHTML-Ausgabe](#): Wenn in den Dokumenteigenschaften (**Datei | Eigenschaften**) in den HTML-Ausgabeeinstellungen XHTML definiert ist, wird ein XHTML-Dokument für die HTML-Ausgabe generiert.
- [Druck des Designs](#): Das Design in der Design-Ansicht kann mit oder ohne Tags ausgedruckt werden.

Version 2010 Release 2

Im Folgenden sind die neuen Funktionen in [StyleVision Version 2010 Release 2](#) aufgelistet.

- Sowohl die Enterprise als auch die Professional Edition ist jetzt als separate 64-Bit und 32-Bit-Applikation verfügbar.
- [Dank Parametern für Design-Fragmente](#) können Design-Fragmente für jede Instanz, in der sie zum Einsatz kommen, mit unterschiedlichen Parameterwerten verwendet werden. An jeder Stelle, an der das Design-Fragment im SPS verwendet wird, kann dem Design-Fragment ein anderer Parameterwert zugewiesen werden.
- Definition [separater Kopf- und Fußzeilen für die erste und letzte Seite](#). Dies ist eine zusätzliche Option neben der Option, für gerade und ungerade nummerierte Seiten und unterschiedliche Dokumentabschnitte unterschiedliche Kopf- und Fußzeilen zu definieren.
- [Layout-Felder](#) und [Linien](#) können mit Hilfe von Tasten verschoben und in ihrer Größe angepasst werden.
- [Vorlagen rund um Tabellenzeilen oder -spalten](#) können hinzugefügt oder gelöscht werden, ohne dass der Inhalt oder die Formatierung der betroffenen Zeile oder Spalte geändert werden muss.
- [Text in Tabellen](#) und [in Layout-Feldern](#) kann im oder entgegen dem Uhrzeigersinn gedreht werden, sodass er vertikal angezeigt wird.
- Wo globale Vorlagen in der Hauptvorlage verwendet werden, [können Filter an diesen Vorlagen angewendet werden](#).
- Die Design-Fragmenten stehen nun nicht mehr nur in der [Design-Struktur](#) zur Verfügung, sondern können auch mit der Maus aus der [Schema-Struktur](#) gezogen werden.

Version 2010 Release 3

Im Folgenden sind die neuen Funktionen in [StyleVision Version 2010 Release 3](#) aufgelistet.

- [Werteformatierung \(Formatierung numerischer Datentypen\)](#): Der frühere Eingabeformatierungsmechanismus wurde erweitert, sodass nun - nur in der Enterprise Edition - Inline XBRL-Werte formatiert werden können, wenn diese in einem (X)HTML-Bericht ausgegeben werden. Die frühere Eingabeformatierungsfunktion bleibt

- unverändert, wurde aber in Werteformatierung umbenannt.
- [Globale Vorlagen](#) können nun für jeden beliebigen Node oder Typen im Schema erstellt werden. In früheren Versionen von StyleVision konnten globale Vorlagen nur für globale Elemente und globale Typen erstellt werden. Nun lassen sich globale Vorlagen zu jedem Node oder Typen und für jedes andere Datenelement, das von einem XPath-Ausdruck zurückgegeben wird, erstellen.
- [Integration in Microsoft Visual Studio 2010](#). Damit wird die Unterstützung für Visual Studio von Version 2005 und 2008 auf Visual Studio Version 2010 ausgeweitet. Visual Studio 2003 wird nicht mehr unterstützt.

Kapitel 4

Einführung

4 Einführung

In diesem Abschnitt wird **Altova® StyleVision® 2012** vorgestellt. Er besteht aus den folgenden Unterabschnitten:

- [Was ist ein SPS?](#) In diesem Abschnitt wird die Aufgabe eines SPS in einer XML-Umgebung hinsichtlich StyleVision erklärt.
- [Produktfeatures](#). Dieser Abschnitt enthält einen Überblick über die wichtigsten Funktionen von StyleVision.
- [Terminologie](#). Dieser Abschnitt enthält eine Liste der auf der StyleVision Benutzeroberfläche und in dieser Dokumentation verwendeten Begriffe.
- [Einrichten von StyleVision](#). In diesem Abschnitt wird beschrieben, wie Sie StyleVision korrekt einrichten.

4.1 Was ist ein SPS?

Ein StyleVision Power Stylesheet (oder SPS) ist ein erweitertes XSLT Stylesheet, welches verwendet wird.

- um festzulegen, wie Daten in der [Authentic](#)-Ansicht von XML-Dokumenten und Datenbanken (DB) angezeigt und eingegeben werden und
- um das Ausgabedesign einer XML-Dokumententransformation zu definieren.

Ein SPS hat die Dateierweiterung `.sps`.

Design des SPS

Ein SPS wird in StyleVision grafisch erstellt und basiert auf einem Schema (DTD oder XML-Schema); wenn das SPS mit einer DB verwendet werden soll, basiert es auf einem XML-Schema, das von StyleVision automatisch anhand der Datenbankstruktur erstellt wird. Das Design des SPS ist flexibel. Es kann dynamische und statische Inhalte enthalten. Bei [dynamischem Inhalt](#) handelt es sich um Daten aus einem XML-Dokument oder Datenbank. Bei [statischem Inhalt](#) 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 die Authentic-Ansicht

Wenn ein fertiges SPS mit einem XML-Dokument oder einer DB verknüpft wird, kann dieses XML-Dokument oder diese Datenbank in der [Authentic](#)-Ansicht bearbeitet werden. Die Authentic-Ansicht ist ideal dazu geeignet, um ein XML-Dokument oder eine Datenbank von mehreren Arbeitsplätzen aus grafisch zu editieren. Ein XML-Dokument oder eine DB kann über die grafische Benutzeroberfläche der Authentic-Ansicht von mehreren Benutzern editiert werden. Sie können das SPS, während Sie es erstellen, (auf dem Authentic-Register für dieses SPS) in StyleVision ansehen und testen. Eine ausführliche Beschreibung dazu, wie SPS in der Authentic-Ansicht verwendet werden, finden Sie unter [SPS und die Authentic-Ansicht](#).

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. Sie können anhand eines einzigen SPS XSLT Stylesheets für die Ausgabe im HTML-, RTF-, XSL-FO- und Word 2007 (und höher)-Format (*XSL-FO und Word 2007 (und höher) nur in der Enterprise Edition; RTF- 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](#).

Das SPS und die Ausgabe

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

Aufrufen von StyleVision [über die Befehlszeile](#).

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

4.2 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.
- Die StyleVision-Funktionalitäten können über die [Befehlszeile](#) aufgerufen werden.
- Die StyleVision-Funktionalitäten können in [externe Applikationen integriert werden](#) und StyleVision kann in [Visual Studio](#) und [Eclipse](#) integriert werden.

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](#) 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](#).

Benutzeroberfläche

Die folgenden allgemeinen GUI-Funktionen stehen zur Verfügung:

- Es können gleichzeitig [mehrere SPS Designs](#) geöffnet sein, wobei immer nur eines aktiv ist. Jedes SPS Design wird auf einem eigenen Register angezeigt.
- Mit Hilfe von [Vorlagenfiltern](#) 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](#): 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 [Authentic-Ansicht](#), die [Ausgabeansichten](#) und Stylesheets anzeigen und auf diese Weise schnell eine Vorschau des Resultats und des XSLT-Codes sehen und die Funktionen der Authentic-Ansicht testen.
- Wenn ein SPS mit einem [XML-Quelldokument](#) oder einer [Quelldatenbank](#) verknüpft wird, kann das Quelldokument direkt in der Authentic-Ansicht von StyleVision bearbeitet werden.

Datenbanken

Folgende Datenbankfunktionen stehen zur Verfügung:

- [DB-Berichte](#) können entweder in StyleVision angezeigt werden oder als HTML-Dokument gespeichert werden.
- [IBM DB2-Datenbanken](#), die XML-Spalten enthalten, werden unterstützt.
- Eine Datenbank kann direkt von StyleVision [aus abgefragt werden](#).

Ausgabe

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

- Es werden beide [XSLT-Versionen \(1.0 und 2.0\)](#) unterstützt. XSLT 2.0 bietet umfangreiche Funktionen zum Zugriff auf und zur Bearbeitung von Daten.

4.3 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](#)
- [Allgemeine XML-Termini und Begriffe](#)
- [XSLT- und XPath-Begriffe](#)
- [StyleVision-spezifische Begriffe](#)

Anmerkung: Wenn einer der unten angeführten Links auf einen Begriff verweist, der in der Ansicht bereits zu sehen ist, ändert sich die Anzeige nicht, wenn Sie auf den Link klicken; suchen Sie den Begriff in diesem Fall im angezeigten Bereich.

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 <code>.sps</code> gespeichert wird. Eine ausführliche Beschreibung finden Sie unter Was ist ein SPS? .
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 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 .
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 <code>http</code> -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](#) in XPath 2.0 bezeichnet. In dieser Dokumentation wird die XSLT 2.0- und XPath 2.0-Terminologie verwendet.

Absoluter XPath	Ein Pfadausdruck, der am Root Node der Baumstruktur beginnt, die den Kontext-Node 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 .
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 . Bei Unterausdrücken kann sich der Kontext-Node allerdings ändern. Beachten Sie, dass es sich bei der <code>current()</code> Funktion um eine XSLT-Funktion und nicht um eine XPath-Funktion handelt und dass sie daher in den automatischen Berechnungen und Bedingungsvorlagen von StyleVision nicht verwendet werden kann. Um den aktuellen Node in einem Ausdruck auszuwählen, verwenden Sie den <code>for</code> Ausdruck von XPath 2.0.
Dokument-Element	In einem wohlgeformten XML-Dokument wird das äußerste Element als Dokumentelement bezeichnet. Es ist ein Child des Dokument-Node . 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: <code>' / '</code> dargestellt. Im Schema-Struktur-Fenster von StyleVision wird er folgendermaßen dargestellt: <code>' / Root elements'.</code>

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 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 können nur für globale Elemente definiert werden.
globale Vorlage	Eine globale Vorlage kann für ein globales Element 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 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 verarbeitet wird. Die lokale Vorlage wird auf dieses eine Element an dieser bestimmten Stelle angewendet, an der das Element in der Hauptvorlage vorkommt. Anstelle der lokalen Vorlage kann eine globale Vorlage auf eine bestimmte Instanz eines Elements in der Hauptvorlage angewendet werden.
Hauptschema	Eine der zugewiesenen Schemaquellen wird als das Hauptschema definiert; der Dokument-Node der XML-Arbeitsdatei , die mit dem Hauptschema verknüpft ist, wird als Ausgangspunkt für die Hauptvorlage verwendet.
Hauptvorlage	Die Hauptvorlage dient als Ausgangspunkt. In StyleVision entspricht diese Vorlage dem Dokumentelement . Sie ist das erste Objekt, das vom XSLT-Prozessor verarbeitet wird. Im Fenster "Schema-Struktur" wird sie als Child des Dokument-Node aufgelistet. In der Hauptvorlage ist die grundlegende Struktur des Ausgabedokuments definiert und wie ein oder mehrere Eingabedokumente verarbeitet werden sollen. Sie kann lokale Vorlagen enthalten und globale Vorlagen referenzieren.
Ausgabe	Das Ausgabedokument, das durch die Verarbeitung eines XML-Dokuments mit einem XSLT Stylesheet erzeugt wird. Sie können mit StyleVision Dokumente im HTML- und RTF-Format generieren. Die Authentic-Ansicht wird nicht als Ausgabe betrachtet sondern separat als Authentic-Ansicht bezeichnet. 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 oder einen Text-String; oder (iii) ein vordefiniertes Format (das im SPS durch Start- und End-Tags dargestellt wird).
Vorlage	Wird grob als eine Gruppe von Anweisungen zur Verarbeitung eines Node oder einer 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 Authentic-Ansicht und 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 Dokumentsymbol ist. Durch Zuweisung einer XML/XBRL-Arbeitsdatei wird dieses Problem gelöst (da ein gültiges XML-Dokument definitionsgemäß nur ein Dokumentsymbol 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 einschließlich Datenbankquellen (anhand derer XML-Datendokumente für die Verwendung mit einem SPS generiert werden. Die Bedeutung ergibt sich jeweils aus dem Zusammenhang.

4.4 Einrichten von StyleVision

Altova StyleVision läuft auf Windows 200, Windows XP, Windows Vista und Windows 7. 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 Authentic-Ansicht, die HTML-Vorschau und die Design-Ansicht. Internet Explorer 6.0 und höher bietet eine bessere XML-Unterstützung. Es wird die Verwendung von Internet Explorer 9 empfohlen.
- Microsoft Word 2000 oder höher für eine RTF-Vorschau. Empfohlen wird Microsoft Word 2003.
- Microsoft Word 2007 (und höher) oder das Word 2007 Ansichtsprogramm von Microsoft für die Anzeige von Word 2007 (und höher) in der Word 2007-Vorschau. Zur Anzeige der Word 2007 (und höher)-Ausgabe kann Microsoft Word 2003 mit dem Kompatibilitätspaket verwendet werden. Manche Word 2007 (und höher)-Dateien können damit allerdings nicht ordnungsgemäß dargestellt werden.

Anmerkung: In dieser Dokumentation verwenden wir für Microsoft Word 2007 oder höhere Programmversionen die Abkürzung **Word 2007+**.

Anmerkung: Falls es ein Problem mit der eingebetteten Vorschau gibt, versucht StyleVision das Vorschaudokument in einer externen Applikation (normalerweise in MS Word oder Adobe Reader) anzuzeigen. In StyleVision wird eine Fehlermeldung über die eingebettete Vorschau angezeigt. Wenn das Vorschaudokument in einer externen Applikation angezeigt wird, müssen Sie die externe Applikation schließen, bevor Sie das temporäre Ausgabedokument neu generieren, da sonst eine Fehlermeldung ausgegeben wird, die besagt, dass die Datei von einem anderem Programm verwendet wird. Sie sollten die externe Applikation außerdem vor dem Schließen des SPS-Designs schließen, da StyleVision die temporäre Ausgabedatei sonst nicht schließen kann, da sie von der externen Applikation gesperrt wurde.

Befehlszeilendienstprogramm: StyleVisionBatch

Die Installation schließt auch ein [Befehlszeilendienstprogramm](#), nämlich `StyleVisionBatch.exe` ein. Mit Hilfe dieses Dienstprogramms können Sie die Dateigenerierungsfunktion von StyleVision über die Befehlszeile aufrufen. StyleVisionBatch befindet sich im Ordner der StyleVision Applikation.

4.5 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

In StyleVision kann die Authentic-Ansicht auf dem Register "Elektronisches Authentic-Formular" des [Hauptfensters](#) angezeigt werden.

Enterprise Edition von Applikationen, die die Authentic-Ansicht enthalten

Die folgenden SPS Funktionalitäten sind nur in der jeweiligen **Enterprise Edition** der Altova Applikation [Authentic-Ansicht](#) aktiv:

- [Absolute Positionierung \(Layout Container\)](#)
- [Aufruf von Java- und .Net-Funktionen über XPath-Ausdrücke in automatischen Berechnungen](#)
- [Variablen](#)
- [Benutzerdefinierte Elemente und XML-Textblöcke](#)

Wenn eine dieser Funktionen in einem SPS enthalten ist, das mit einer anderen als der Enterprise Edition für die Authentic-Ansicht geöffnet wird (z.B. mit der XMLSpy Professional Edition), so wird eine Meldung angezeigt, die besagt, dass diese Funktionalität nur in der Enterprise Edition der Applikation zur Verfügung steht.

Anmerkung: Die StyleVision Enterprise Edition unterstützt die Enterprise Edition der Authentic-Ansicht, während die StyleVision Professional Edition die Community Edition der Authentic-Ansicht unterstützt.

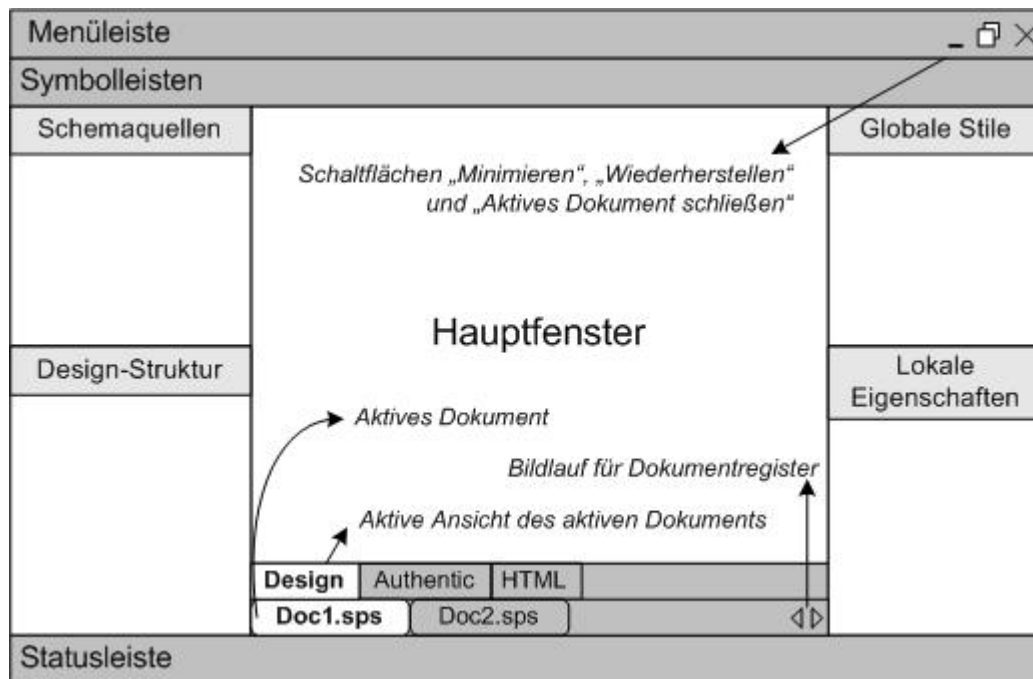
Kapitel 5

Benutzeroberfläche

5 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](#) beschrieben. Die Menüleiste enthält außerdem die Schaltflächen zum Minimieren, Wiederherstellen und Schließen des aktiven Dokuments.
- einem **Symbolleistenbereich**. Die verschiedenen [Symbolleisten](#) und die Schaltflächen zum Aufrufen der Befehle in den einzelnen Symbolleisten sind im Abschnitt [Benutzerreferenz](#) beschrieben.
- einem aus mehreren Registern bestehenden **Hauptfenster**, in dem ein oder mehrere geöffnete SPS angezeigt werden. In diesem Fenster können Sie das [Design des SPS](#) und den Inhalt der [Authentic-Ansicht](#) (im Register "Elektronisches Authentic-Formular") bearbeiten und eine [Vorschau der XSLT Stylesheets und Ausgabedokumente](#) anzeigen.
- die **Design-Seitenleisten**—die Fenster [Design-Übersicht](#), [Schema-Struktur](#), [Design-Struktur](#), [Globale Stile](#), [Stile](#), [Eigenschaften](#) und [Projekt](#) - welche angedockt oder freischwebend angezeigt werden können.
- eine **Statusleiste**, in der Informationen zum Status der Applikation angezeigt werden. Wenn Sie die 64-Bit-Version von StyleVision verwenden, so wird dies in der Statusleiste durch das Suffix (x64) nach dem Applikationsnamen angezeigt. Für die 32-Bit-Version gibt es kein Suffix.

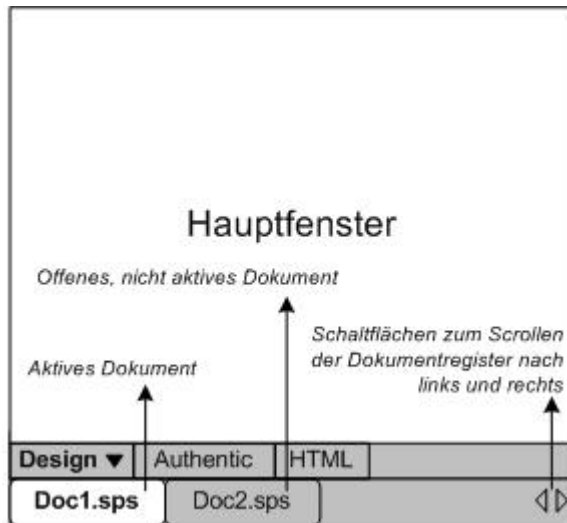


Das [Hauptfenster](#) und die [Design-Seitenleisten](#) werden in den Unterabschnitten dieses Abschnitts näher beschrieben.

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

5.1 Hauptfenster

Im **Hauptfenster** (siehe Abbildung unten) werden das SPS Design, die Authentic-Ansicht, 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](#)).

Dokumentansichten

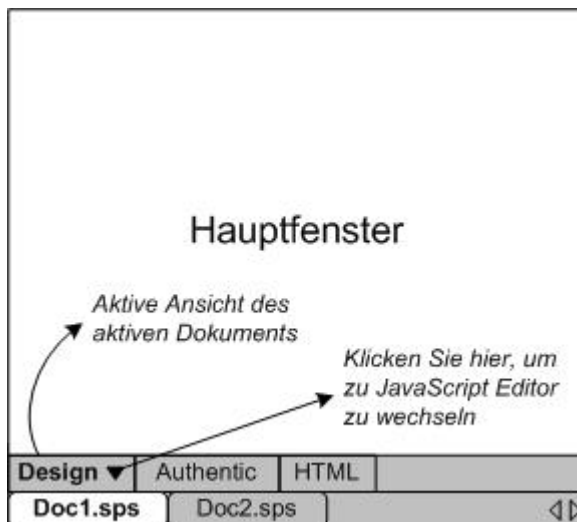
Ein Dokument wird in den folgenden Ansichten angezeigt, wobei immer nur eine davon aktiv sein kann:

- [Design-Ansicht](#). 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.
- [Authentic-Ansicht](#) (Elektronisches Authentic-Formular). In dieser Ansicht sehen Sie sofort, wie ein XML-Dokument (die [XML-Arbeitsdatei](#)) in der Authentic-Ansicht aussieht. Das SPS wird dynamisch auf die [XML-Arbeitsdatei](#) angewendet, so dass Sie das Design in der Authentic-Ansicht testen können.
- [Ausgabebansichten](#) (HTML- und RTF-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.

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.

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.

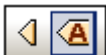


Sie können in der Design-Ansicht auch zu einem [JavaScript Editor](#) oder (Authentic) Skript-Editor wechseln. Im JavaScript-Editor können [JavaScript-Funktionen](#) erstellt und bearbeitet werden können. Diese Funktionen stehen dann auf der Benutzeroberfläche für die Verwendung im SPS zur Verfügung. Im Skript-Editor können Skripts für die Authentic-Ansicht erstellt und bearbeitet werden. Um zum [JavaScript Editor](#) oder zum (Authentic) Skript-Editor zu wechseln, klicken Sie auf dem Register "Design" auf die Dropdown-Schaltfläche (siehe Abbildung oben) und wählen Sie im Dropdown-Menü die Option JavaScript bzw. (Authentic) Skript-Editor. Um zurück zur Design-Ansicht zu wechseln, klicken Sie auf dem Register "JavaScript" bzw. (Authentic) Skript-Editor 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](#), die in Form von [Symbolleisten-Schaltflächen](#) 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).

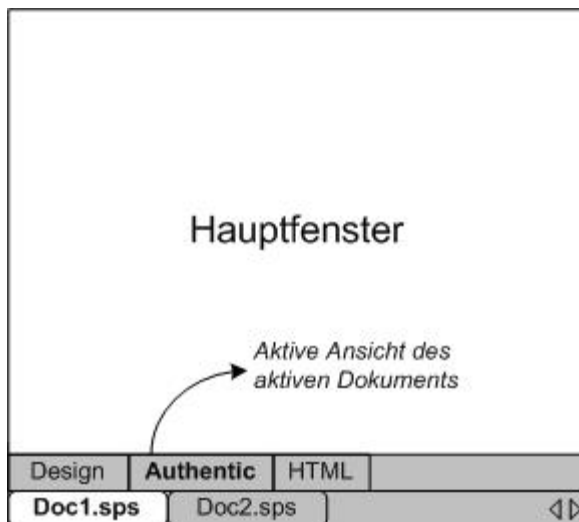


Die oben gezeigten Schaltflächen dienen zum Ein- und Ausschalten von Tags. Die Schaltflächen von links nach rechts sind: (i) Kleine Design Markup-Symbole einblenden (Tags ohne Namen); (ii) Große Design Markup-Symbole einblenden (Tags mit Namen). Wenn kleine Markup-Symbole eingeblendet werden, wird der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

Authentic-Ansicht

In der **Authentic**-Ansicht des Hauptfensters (das Elektronische Authentic-Formular) können Sie die [XML-Arbeitsdatei](#) in ihrer Authentic-Ansicht anzeigen und bearbeiten. In dieser Ansicht (i) sehen Sie, wie Ihr Authentic-XML-Dokument aussehen wird und (ii) Sie können die anhand des SPS erstellte Authentic-Ansicht testen. Dies ist vor allem nützlich, um die dynamischen Funktionen der Authentic-Ansicht zu testen. So können Sie z.B. testen, wie sich die Authentic-Ansicht verhält, wenn Sie:

- neue Elemente und Attribute hinzufügen
- neue Absätze oder Tabellenzeilen hinzufügen
- Werte ändern, die sich auf Bedingungsvorlagen auswirken



Die Authentic-Ansicht und die XML-Arbeitsdatei

Damit die Authentic-Ansicht angezeigt werden kann, muss dem aktiven SPS-Dokument eine [XML-Arbeitsdatei zugewiesen werden](#). Diese XML-Arbeitsdatei muss gemäß dem Schema, auf dem das SPS basiert, gültig sein.

StyleVision erstellt eine temporäre XML-Datei, die auf der XML-Arbeitsdatei basiert, und die im Hauptfenster auf dem Register Authentic angezeigt wird. Wenn Sie in der Authentic-Ansicht Änderungen vornehmen, wird dadurch die temporäre XML-Datei geändert. Die XML-Arbeitsdatei selbst wird erst geändert, wenn Sie die Änderungen explizit speichern (über den Menübefehl [Datei | XML-Authentic-Daten speichern](#)).

Wurde keine XML-Arbeitsdatei zugewiesen, werden Sie aufgefordert, eine XML-Arbeitsdatei zuzuweisen, wenn Sie auf das Register Authentic klicken.

Authentic-Einschränkungen

Die Authentic-Ansicht im Hauptfenster ähnelt der Vollversion der Authentic-Ansicht, die in XMLSpy und Authentic Desktop zur Verfügung steht. Ausnahmen bilden die folgenden Aspekte:

- Die Eingabehilfen der Authentic-Ansicht stehen auf der Benutzeroberfläche nicht zur Verfügung. Zum Einfügen oder Anhängen von Nodes müssen Sie das Kontextmenü verwenden, das Sie durch Rechtsklick aufrufen können.
- Es stehen keine CALS/HTML-Tabellen zum Einfügen zur Verfügung.
- Es stehen keine Textstatussymbole zur Verfügung.

Um diese Funktionen zu testen, verwenden Sie die Vollversion der Authentic-Ansicht in XMLSpy oder Authentic Desktop.
Eine ausführliche Beschreibung der Verwendung der Authentic-Ansicht finden Sie im Abschnitt [Bearbeitung in der Authentic-Ansicht](#). Zusätzliche Informationen finden Sie im Benutzerhandbuch zu XMLSpy oder Authentic Desktop im Tutorial zur Authentic-Ansicht.

Enterprise Editionen von Applikationen mit der Authentic-Ansicht

Die folgenden SPS Funktionalitäten sind nur in der jeweiligen **Enterprise Edition** der Altova Applikation [Authentic-Ansicht](#) aktiv:

- [Absolute Positionierung \(Layout Container\)](#)
- [Aufruf von Java- und .Net-Funktionen über XPath-Ausdrücke in automatischen Berechnungen](#)
- [Variablen](#)
- [Benutzerdefinierte Elemente und XML-Textblöcke](#)

Wenn eine dieser Funktionen in einem SPS enthalten ist, das mit einer anderen als der Enterprise Edition für die Authentic-Ansicht geöffnet wird (z.B. mit der XMLSpy Professional Edition), so wird eine Meldung angezeigt, die besagt, dass diese Funktionalität nur in der Enterprise Edition der Applikation zur Verfügung steht.

Anmerkung: Die StyleVision Enterprise Edition unterstützt die Enterprise Edition der Authentic-Ansicht, während die StyleVision Professional Edition die Community Edition der Authentic-Ansicht unterstützt.

Ausgabeansichten

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](#)) und Sie können daraus Text in die Zwischenablage kopieren (mit Hilfe von [Bearbeiten | Kopieren](#)).

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

Ausgabe-Vorschau

In der Ausgabe-Vorschau wird die durch Transformation der [XML-Arbeitsdatei](#) mit dem XSLT Stylesheet für dieses Ausgabeformatgenerierte Ausgabe angezeigt. Die Ausgabe wird jedesmal neu generiert, wenn Sie auf das Register die Ausgabe-Vorschau klicken. Beachten Sie, dass die gespeicherte Version der XML-Arbeitsdatei transformiert wird - nicht die temporäre Version, die in der Authentic-Ansicht bearbeitet wird. Das bedeutet, dass in der Authentic-Ansicht vorgenommene Änderungen in der Ausgabe-Vorschau nur dann berücksichtigt werden, nachdem diese Änderungen in der [XML-Arbeitsdatei](#) gespeichert wurden ([Datei | XML-Authentic-Daten speichern](#)).

Wenn bei Auswahl der Ausgabe-Vorschau auf dem Ausgabe-Ansichtsregister keine [XML-Arbeitsdatei](#) 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](#) verwendet wird.

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

5.2 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](#)
 - [Schema-Struktur](#)
 - [Design-Struktur](#)
 - [Globale Stile](#)
 - [Stile](#)
 - [Eigenschaften](#)
 - [Projekt](#)

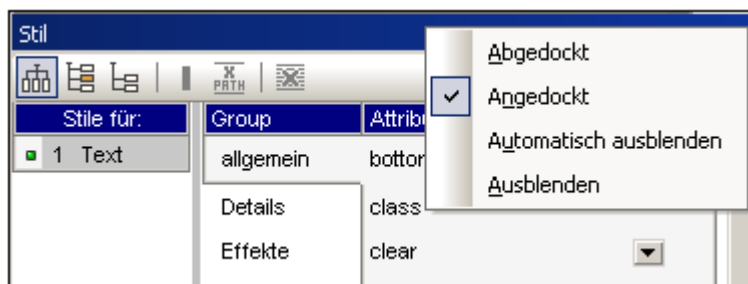
Layout der Ansichten

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. Beachten Sie: Das Layout, das für eine Ausgabevorschau definiert wurde (HTML- und RTF-Vorschau) wird auf alle Ausgabevorschauen angewendet. Die Ansichtskategorien sind: (i) kein offenes Dokument; (ii) Design-Ansicht; (iii) Ausgabevorschauen; und (iv) Authentic-Ansicht.

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.

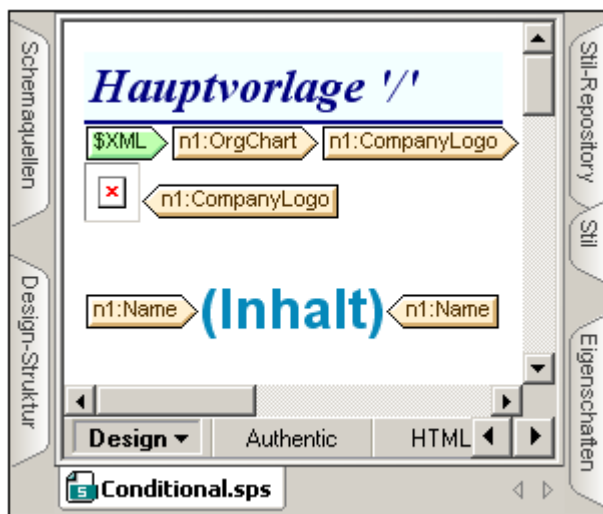
Alternativ dazu können Sie auch die folgenden Methoden verwenden. Um ein angedocktes Fenster freischwebend anzuzeigen, klicken Sie auf die Schaltfläche **"Menü"** in der rechten oberen Ecke eines angedockten Fensters (*siehe Abbildung unten*) und wählen Sie die Option "Abgedockt". Sie können dieses Menü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



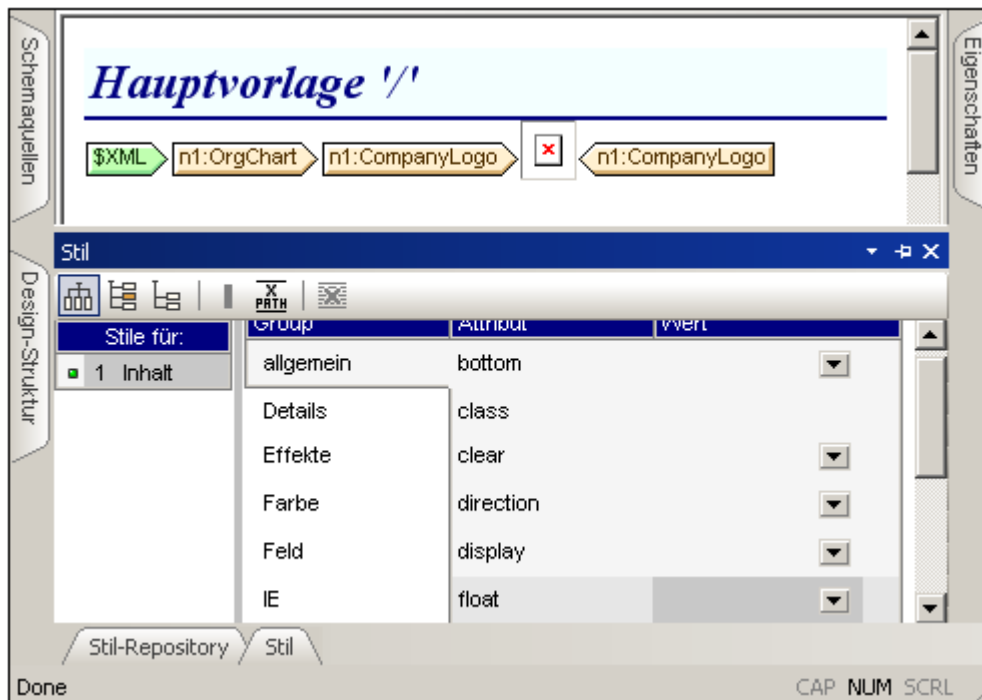
Um ein freischwebendes Fenster anzudocken, rechtsklicken Sie auf die Titelleiste des freischwebenden Fensters und wählen Sie im Menü das daraufhin angezeigt wird, die Option "Angedockt"; das Fenster wird in der Position andockt, an der es sich befand, als es zuletzt andockt war.

Automatisches Ausblenden der Seitenleistenfenster

Ein andocktes Fenster kann automatisch ausgeblendet werden. In diesem Fall wird es in Form eines minimierten Registers am Rand der Benutzeroberfläche angezeigt. In der Abbildung unten wurden fünf Seitenleisten automatisch ausgeblendet: zwei am linken Rand der Benutzeroberfläche, zwei am unteren Rand und eine am rechten Rand.



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.

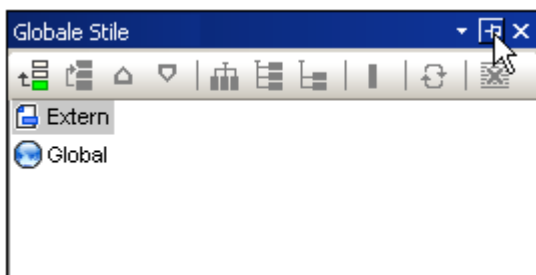
Die Funktion zum automatischen Ausblenden ist nützlich, wenn Sie selten gebrauchte Seitenleisten aus der Benutzeroberfläche entfernen und jederzeit wieder darauf Zugriff haben möchten, sollten Sie diese doch benötigen. Auf diese Art haben Sie im Hauptfenster mehr Platz zur Verfügung und trotzdem jederzeit 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](#) die Option "Automatisch ausblenden" auswählen; (zum Anzeigen des [Menüs](#) klicken Sie mit der rechten Maustaste auf die Titelleiste des Fensters oder klicken Sie in der Titelleiste des angedockten Fensters auf die [Schaltfläche Menü](#)).



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](#) die Option "Automatisch ausblenden" deaktivieren; (zum Anzeigen des [Menüs](#) klicken Sie mit der rechten Maustaste auf die Titelleiste des Fensters oder klicken Sie in der Titelleiste des angedockten Fensters auf die [Schaltfläche Menü](#)).



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 andockten oder freischwebenden Fenster auf die Schließen-Schaltfläche. Alternativ dazu wählen Sie im [Menü](#) die Option "Ausblenden"; (Klicken Sie zum Anzeigen des [Menüs](#) mit der rechten Maustaste auf die Titelleiste des Fensters oder klicken Sie in der Titelleiste des Fensters auf die [Schaltfläche Menü](#)).

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

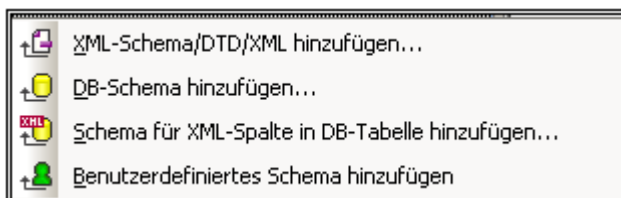
Design-Übersicht

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 (i) ein XML-Schema, eine DTD oder ein automatisch von StyleVision anhand einer XML-Datei generiertes XML-Schema; (ii) ein von StyleVision anhand einer Datenbank generiertes Schema; oder (iii) ein benutzerdefiniertes Schema oder hinzufügen können.



Die XML-Arbeitsdatei und die XML-Vorlagendatei

Wenn ein Schema hinzugefügt wird, wird es unter dem Eintrag "Schemaquellen" aufgelistet. Jedes Schema hat zwei Untereinträge:

- die [XML-Arbeitsdatei](#).
- die [XML-Vorlagendatei](#).

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.

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](#) 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 sowie die XML-Vorlagendatei die dem SPS zugewiesen wurden 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](#) 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](#), 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](#) (Objekte innerhalb von Modulen) werden jedoch in der [Seitenleiste "Design-Struktur"](#) 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](#), 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](#) werden jedoch in der [Seitenleiste "Globale Stile"](#) 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**.

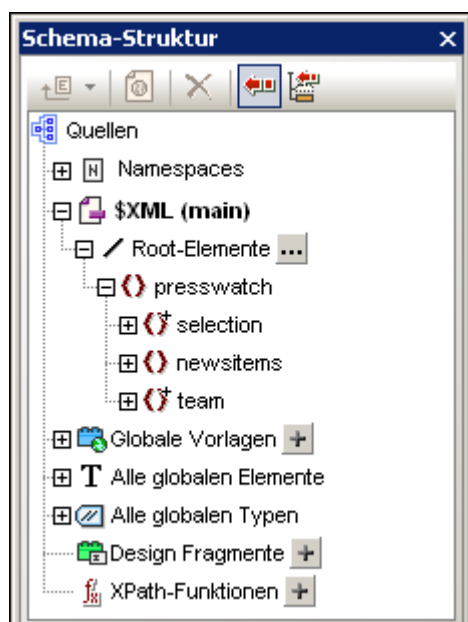
XSLT-Dateien

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](#).

Schema-Struktur

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

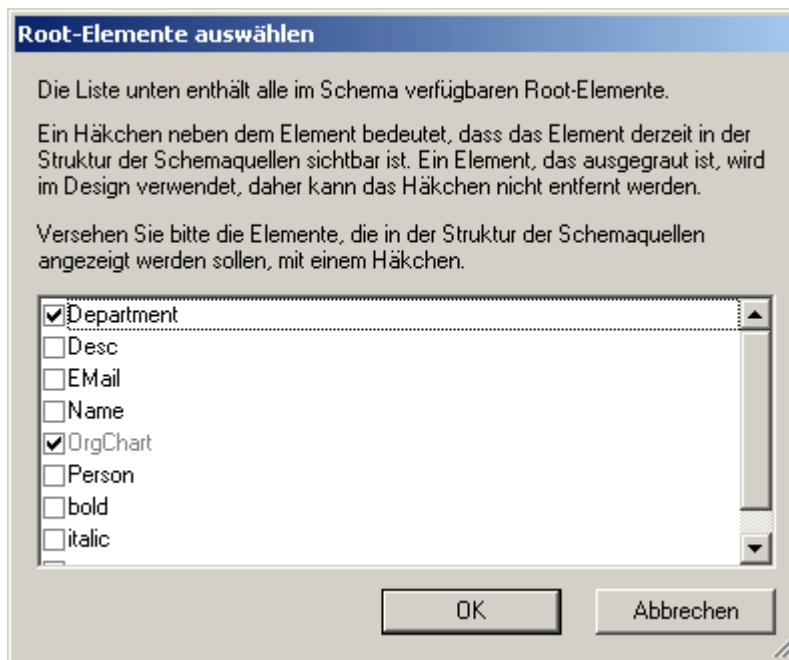
- Auswählen mehrerer Root-Elemente (Dokumentelemente) für ein Schema.
- Ziehen von Nodes aus einer Schema-Struktur in das Design. Diese Nodes stellen den XML-Inhalt dar, der in die Ausgabe inkludiert werden soll.
- Auflisten aller globalen Elemente und Typen in der Schemaquelle. Ein globales Element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



Root-Elemente

Für jedes Schema werden unter der Überschrift `$XML` die ausgewählten [Root-Elemente](#) (auch [Dokumentelemente](#) 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).

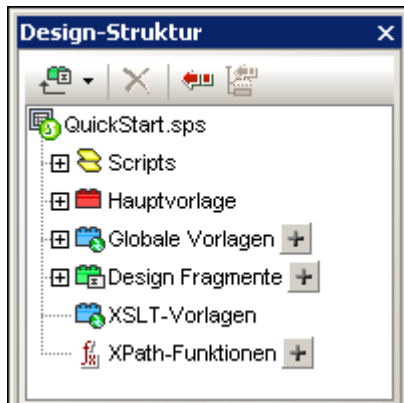
Um Root-Elemente für ein Schema auszuwählen, gehen Sie folgendermaßen vor: Klicken Sie rechts vom Eintrag `Root Elemente` auf die Auswählen-Schaltfläche . Daraufhin wird das Dialogfeld "Root-Elemente auswählen" (siehe Abbildung unten) angezeigt, in dem Sie auswählen können, welche/s der globalen Elemente 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](#).



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

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:

- [Scripts](#): Zeigt alle JavaScript-Funktionen an, die mit dem JavaScript Editor von StyleVision für das SPS definiert wurden.
- [Hauptvorlage](#): Zeigt die detaillierte Struktur der Hauptvorlage an.
- [Globale Vorlagen](#): Listet die globalen Vorlagen im SPS sowie die globalen Vorlagen in allen inkludierten SPS Modulen auf.
- [Design-Fragmente](#): Zeigt alle Design-Fragmente im Design und ermöglicht Ihnen sie zu erstellen, bearbeiten, umzubenennen und zu löschen.
- [XSLT-Vorlagen](#): Geben Ihnen die Möglichkeit, XSLT-Vorlagen in importierten XSLT-Dateien anzuzeigen.
- [Benutzerdefinierte XPath-Funktionen](#): 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

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.



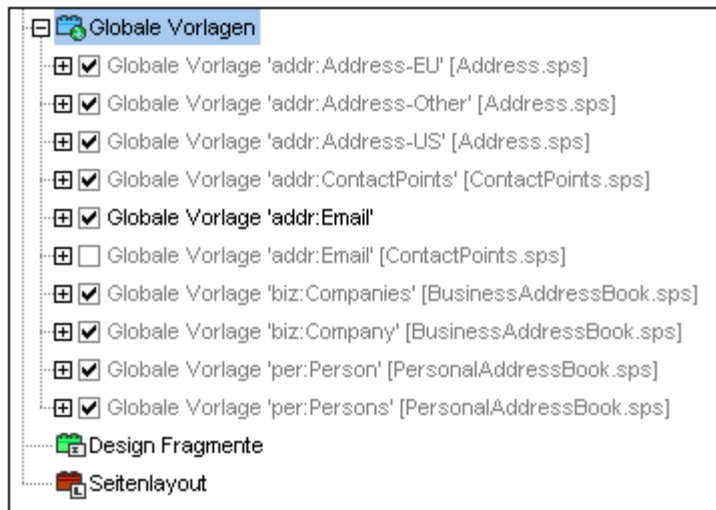
Mit dem Befehl **Entfernen** entfernen Sie den ausgewählten Eintrag aus der Design-Struktur. Mit dem Befehl **Als Design-Fragment definieren** erstellen Sie ein [Design-Fragment](#) im Design und fügen dafür einen Eintrag in der Design-Struktur hinzu. Mit dem Befehl **Alle erweitern** werden alle Einträge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

Scripts und Hauptvorlagen

Unter der Überschrift "Scripts" werden alle Scripts 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

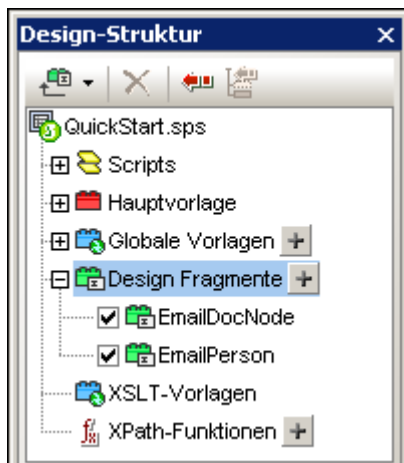
Unter dem Eintrag "[Globale Vorlagen](#)" werden alle globalen Vorlagen im aktuellen SPS und in alle hinzugefügten SPS Modulen aufgelistet. Im aktuellen SPS definierte globale Vorlagen werden schwarz angezeigt, globale Vorlagen, die in hinzugefügten Modulen definiert wurden, werden grau angezeigt (*siehe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



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

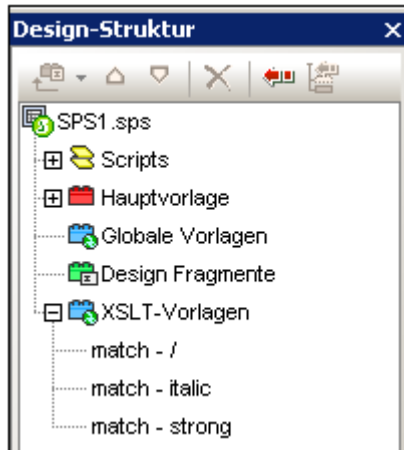
Unter dem Eintrag [Design-Fragmente](#) werden alle Design-Fragmente im aktuellen SPS und in allen hinzugefügten SPS Modulen aufgelistet. Im aktuellen SPS definierte Design Fragmente werden schwarz angezeigt, Design-Fragmente, die in hinzugefügten Modulen definiert wurden, werden grau angezeigt (*siehe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

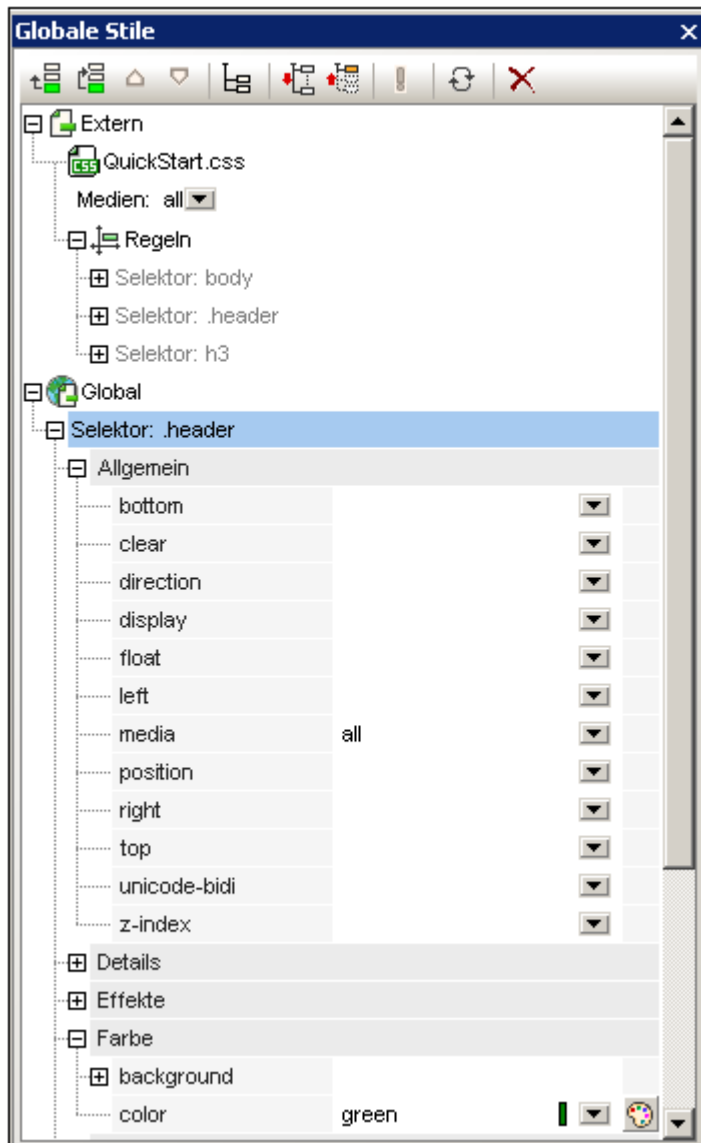
In der Seitenleiste "Design-Struktur" (*Abbildung unten*) werden die in der importierten XSLT-Datei enthaltenen XSLT-Vorlagen unter der Überschrift "XSLT-Vorlagen" angezeigt.



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](#).

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 die Authentic-Ansicht und 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 Globale Stile

Im Fenster "Globale 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](#). Beachten Sie, dass auch ein externes CSS-Stylesheet oder ein über die [Seitenleiste "Design-Übersicht"](#) 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](#).

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

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****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.

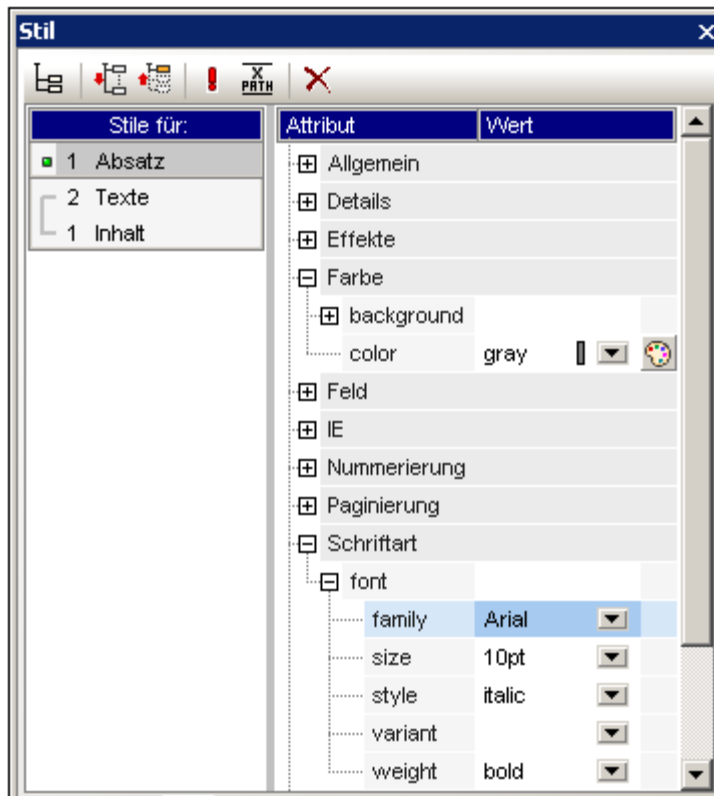
Bearbeiten von CSS-Stilen im Fenster Globale Stile

Im Fenster Globale 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](#).
- Die Selektoren und Eigenschaften globaler Stile können direkt im Fenster Globale Stile bearbeitet werden. Eine Beschreibung dazu finden Sie im Abschnitt [Globale Definition von CSS-Stilen](#).

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](#).
- die Spalte 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](#). 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 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.

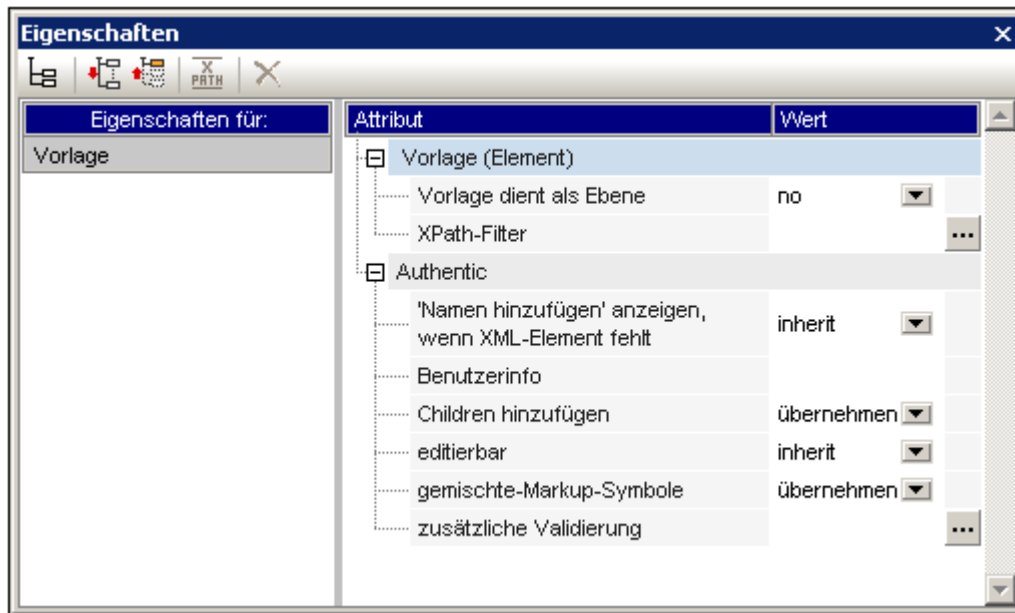
- **Nicht leere auflisten** : Wenn Sie diese Einstellung einschalten, werden für die in der Spalte auf der linken Seite ausgewählten Komponententypen nur die Stileigenschaften (in alphabetischer Reihenfolge) aufgelistet, für die ein Wert definiert ist. Andernfalls werden alle verfügbaren Stileigenschaften 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.

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:

- der Spalte **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](#).
- der Spalte für die **Eigenschaftsdefinitionen:** Hier werden die Komponenteneigenschaften für den Komponententyp definiert, der in der Spalte "Eigenschaften für" ausgewählt ist. Die Spalte 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](#).

Einstellungen für die Definitionsansicht

Sie können die 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
Vorlage	Vorlagen; Authentic
Inhalt	Inhalt; Authentic; Allgemein; Ereignis
Text	Text; Allgemein; Ereignis
Automatische Berechnung	XPath; Allgemein; Ereignis
Bedingungsverzweigung	When
Dateneingabe -element	Authentic; Allgemein; [Dateneingabeelement]; Ereignis; HTML
Bild	Bild; Authentic; Allgemein; Ereignis; HTML
Link	Link; Authentic; Allgemein; Ereignis; HTML
Tabelle	Tabelle; Authentic; Allgemein; Ereignis; HTML; Interaktiv
Absatz	Absatz; Authentic; Allgemein; Ereignis; HTML

Beachten Sie bei Komponententypen bitte folgende Dinge:

- Vorlagenkomponenten sind die Hauptvorlage, globale Vorlagen und alle Schemaknoten im Design.
- 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.
- Dateneingabeelemente 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.

Eigenschaftsgruppe	Beschreibung
XPath	Diese Eigenschaften sind aktiviert, wenn eine automatische Berechnung ausgewählt wurde. Mit der Eigenschaft <code>Werteformatierung</code> wird die Formatierung einer automatischen Berechnung, bei der es sich um einen numerischen oder einen Datums-Datentyp handelt, festgelegt. Mit der Eigenschaft <code>XPath</code> wird der XPath-Ausdruck festgelegt, der für die automatische Berechnung verwendet wird.
Authentic	Dies sind SPS-spezifische Eigenschaften, die für Vorlagen, Inhalt, automatische Berechnungen, Dateneingabeelemente, Bilder, Links, Tabellen und Absätze zur Verfügung stehen. Welche Eigenschaften in der Gruppe zur Verfügung stehen, ist komponentenspezifisch. Nähere Informationen dazu siehe Authentic Node-Eigenschaften .
Allgemein	Die Eigenschaftsgruppe <i>Allgemein</i> steht für alle Komponententypen mit Ausnahme der Komponententypen "Vorlage" und "XPath" zur Verfügung. Sie enthält die folgenden Eigenschaften: <code>class</code> (einen Klassennamen), <code>dir</code> (Schreibrichtung), <code>id</code> (eine eindeutige ID), <code>lang</code> (die Sprache) und <code>title</code> (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 definieren können: <code>onclick</code> , <code>ondblclick</code> , <code>onkeydown</code> , <code>onkeypress</code> , <code>onkeyup</code> , <code>onmousedown</code> , <code>onmousemove</code> , <code>onmouseout</code> , <code>onmouseover</code> , <code>onmouseup</code> .
HTML	Steht für die folgenden Komponententypen zur Verfügung: Dateneingabeelemente ; Bild ; Link ; Tabelle ; Absätze . Beachten Sie, dass es verschiedene Arten von Dateneingabeelementen und Absätzen gibt und dass Tabellen Unterkomponenten haben. Diese Eigenschaften sind HTML-Eigenschaften, die bei den entsprechenden HTML-Elementen (<code>img</code> , <code>table</code> , <code>p</code> , <code>div</code> , 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](#), [Links](#), [Absätze](#) und [andere vordefinierte Formate](#) und [Bedingungsverzweigungen](#). 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.

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](#).*

Ändern oder Löschen eines Eigenschaftswerts

Zum Ändern eines Eigenschaftswerts verwenden Sie eine der im vorigen Absatz [Einstellen von Eigenschaftswerten](#) beschriebenen verfügbaren Methoden. Um einen Eigenschaftswert zu löschen, wählen Sie die Eigenschaft aus und klicken Sie in der Symbolleiste der Seitenleiste

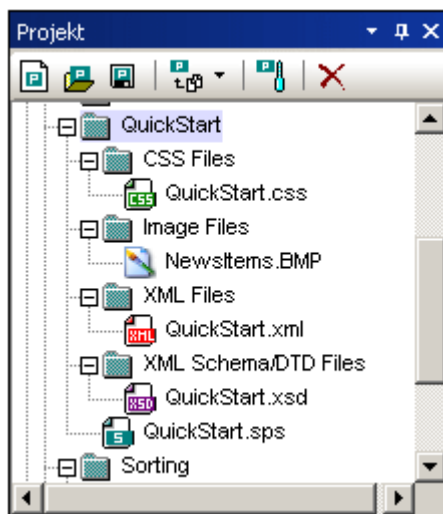
"Eigenschaften" auf das Symbol "Zurücksetzen"  .

Projekt

StyleVision-Projekte sind eine effiziente Methode, um miteinander in Zusammenhang stehende Dateien zu gruppieren, zu verwalten und damit zu arbeiten. Sobald die Dateien in einem Projekt gesammelt wurden, können Sie jederzeit über die Projekt-Seitenleiste aufgerufen werden, wenn Sie ein SPS erstellen. So können Sie z.B. eine SPS-Datei aus der Projekt-Seitenleiste in die Design-Struktur-Seitenleiste ziehen und dort als Modul erstellen; oder Sie ziehen eine Bilddatei als statisches Bild in das Design; oder Sie ziehen ein CSS Stylesheet als externes Stylesheet in die Seitenleiste "Globale Stile".

Eine ausführliche Beschreibung zum Arbeiten mit Projekten finden Sie im Abschnitt [Projekte in StyleVision](#).

In der Seitenleiste **Projekt** (*Abbildung unten*) wird das derzeit aktive Projekt angezeigt. Alle Befehle im [Menü "Projekt"](#) werden auf das derzeit aktive Projekt angewendet. Das derzeit aktive Projekt kann entweder durch Erstellen eines neuen Projekts ([Projekt | Neues Projekt](#)) oder durch Öffnen eines vorhandenen Projekts ([Projekt | Projekt öffnen](#)) geändert werden. Diese beiden Befehle stehen auch in Form von Schaltflächen (erste und zweite Schaltfläche in der Seitenleiste "Projekt") (*siehe Abbildung unten*) zur Verfügung. Sie können einem neu erstellten Projekt beim Speichern ([Projekt | Projekt speichern](#); dritte Schaltfläche in der Symbolleiste in der Seitenleiste "Projekt") einen Namen geben.



Die Symbolleiste-Schaltfläche **Dateien hinzufügen** (*vierte Schaltfläche in der Abbildung unten*) enthält die in der Abbildung unten gezeigten Befehle. Mit Hilfe dieser Befehle können die Dateien und Ordner zum aktiven Projekt hinzugefügt werden. Eine Beschreibung dazu finden Sie in der [Benutzerreferenz](#). Ein nützlicher Befehl ist der Befehl "Aktive und verwandte Dateien hinzufügen", mit dem alle Dateien, die mit dem derzeit aktiven SPS in Zusammenhang stehen, zum Projekt hinzugefügt werden. Wenn z.B. die Tutorial-Datei `QuickStart.sps`, aktiv ist, werden alle damit in Zusammenhang stehenden Dateien (*Abbildung oben*) hinzugefügt.

Kapitel 6

Schnellstart-Tutorial

6 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 [Verzeichnis \(Eigene\) Dateien](#) `C:/Documents and Settings/<username>/My Documents/Altova/`

`StyleVision2012/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.

6.1 Erstellen und Einrichten eines neuen SPS

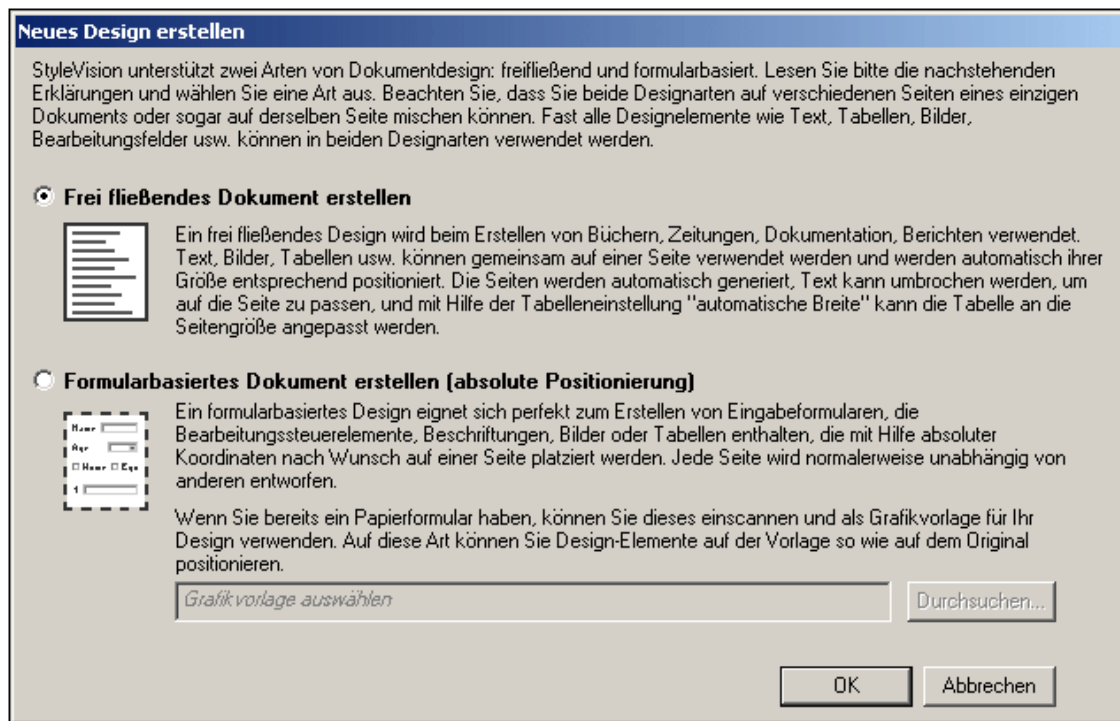
In diesem Abschnitt erfahren Sie Folgendes:

- [Wie man ein neues SPS Dokument erstellt.](#)
- [Wie man eine Schemaquelle für das SPS hinzufügt.](#)
- [Wie man die XSLT-Version des SPS auswählt.](#)
- [Wie man die XML-Arbeitsdatei zuweist.](#)
- [Wie man die Ausgabekodierung definiert.](#)
- [Wie man das SPS Dokument speichert.](#)

Erstellen eines neuen SPS Dokuments

Erstellen Sie ein neues SPS Dokument. Klicken Sie dazu auf **Datei | Neu | Neu (leer)**  oder wählen Sie in der Applikationssymbolleiste über die Dropdown-Liste des **Symbols "Neu"**  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).



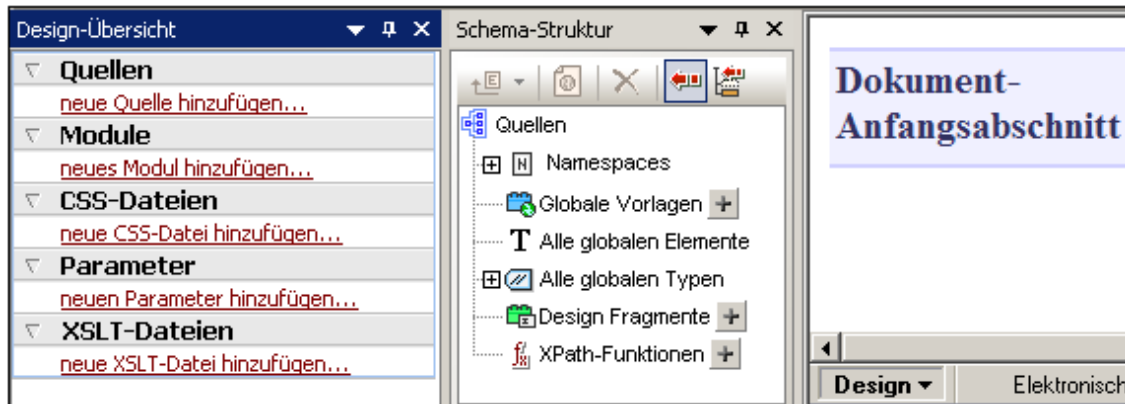
In einem frei fließenden Dokument ist das Dokument vom Layout her so angelegt, dass es den Maßen des Ausgabemediums entspricht (Papier oder Bildschirm). Elemente im Dokumentinhalt können nur relativ zueinander platziert werden, nicht aber an einer absoluten Position. Diese Art von Design eignet sich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

Ein in einem formularbasierten Dokument wird ein einziger [Layout Container](#) 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 orientieren soll, können Sie ein Originalformular als [Grafikvorlage](#) 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](#) angezeigt (*Abbildung unten*).

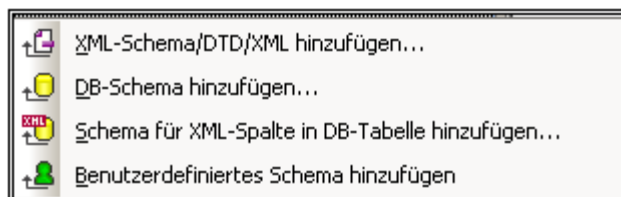


In der [Design-Ansicht](#) wird eine leere Vorlage angezeigt. Die Seitenleisten [Design-Übersicht](#) und [Schema-Struktur](#) enthalten keine Schemaeinträge.

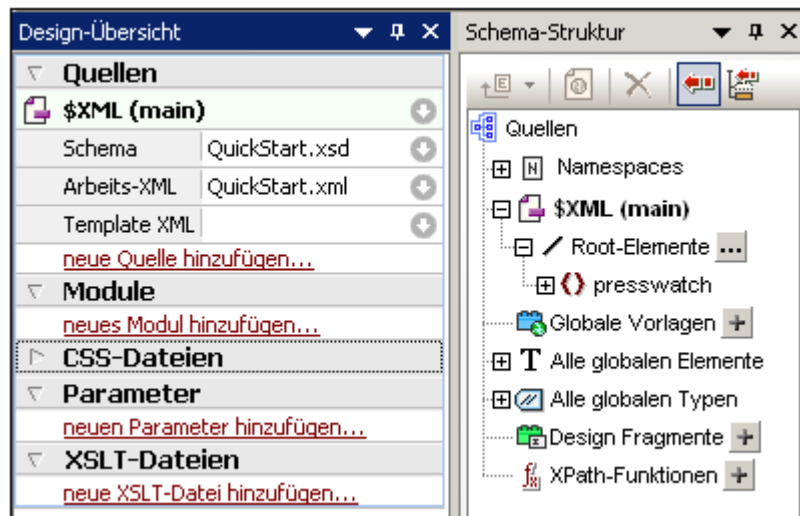
Hinzufügen einer Schemaquelle

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 [Verzeichnis \(Eigene\) Dateien](#) nach der Datei `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/QuickStart/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 \(Eigene\) Dateien](#) nach der Datei `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/QuickStart/QuickStart.xml` 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.



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

Zuweisen oder Ändern der XML-Arbeitsdatei

Als vorher ein XML-Schema zum SPS hinzugefügt wurde, wurde dem Schema auch eine [XML-Arbeitsdatei](#) zugewiesen. Eine XML-Arbeitsdatei liefert dem SPS eine XML-Datenquelle für die Verarbeitung. Um eine [XML-Arbeitsdatei](#) zuzuweisen oder zu ändern, klicken Sie auf die

Schaltfläche **XML-Arbeitsdatei hinzufügen**  direkt unterhalb des Schemanamens. Suchen Sie im Dialogfeld "Öffnen" nun die gewünschte XML-Datei und klicken Sie auf **Öffnen**. Die [XML-Arbeitsdatei](#) wurde nun zugewiesen und der Dateiname wird in die Zeile für die [XML-Arbeitsdatei](#) eingetragen (siehe *Abbildung unten*). Stellen Sie, bevor Sie fortfahren sicher, dass Sie die Datei `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/QuickStart/QuickStart.xml`, die sich im [Verzeichnis \(Eigene\) Dateien](#) befindet, korrekt als XML-Arbeitsdatei zugewiesen haben.

Definieren der Kodierung für die Ausgabe

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

Speichern des SPS Dokuments

Nachdem Sie das SPS wie oben beschrieben eingerichtet haben, speichern Sie es als `MyQuickStart.sps` in Ordner `C:/Documents and Settings/<username>/My Documents/`

Altova/StyleVision2012/StyleVisionExamples\Tutorial\QuickStart. Klicken Sie dazu auf den Menübefehl **Datei | Design speichern** oder drücken Sie **Strg+S** und geben Sie daraufhin im Dialogfeld "Design speichern" den Dateinamen ein.

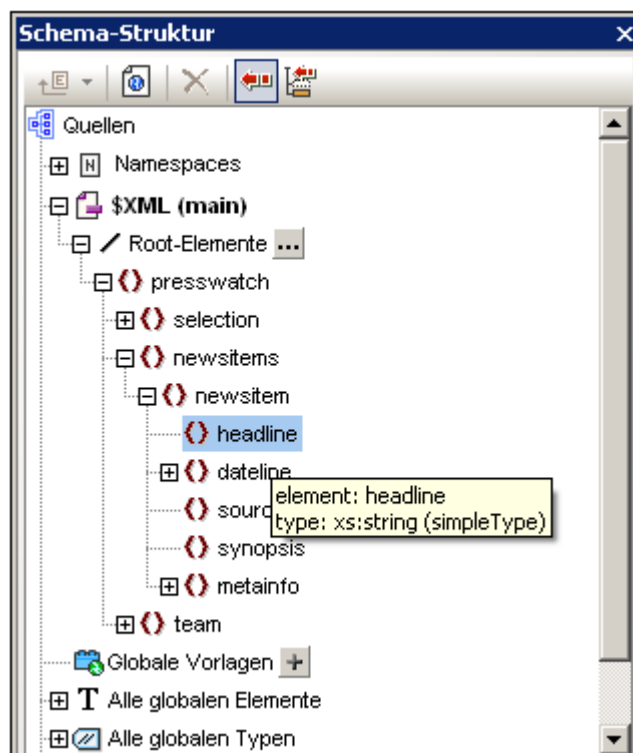
6.2 Einfügen von dynamischem Inhalt (aus der XML-Quelle)

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:

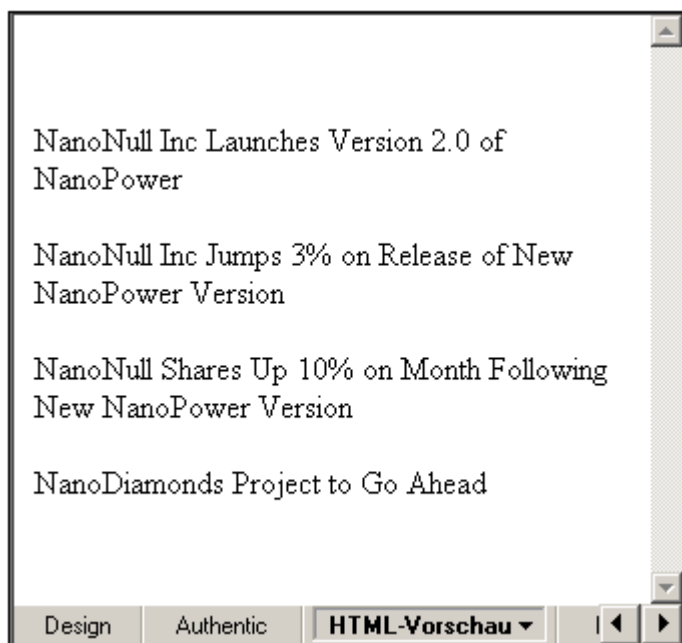
1. Erweitern Sie die Schemastruktur in der [Seitenleiste "Schema-Struktur"](#) bis zu den Child-Elementen des Elements `newsitem` (*Abbildung unten*).



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](#). Wenn sich der Pfeilcursor in einen Einfügekpunkt 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](#) zu sehen (Abbildung unten). In der HTML-Vorschau sehen Sie den Inhalt der `headline` Child-Elemente von `newsitem`, jedes als Textstring.



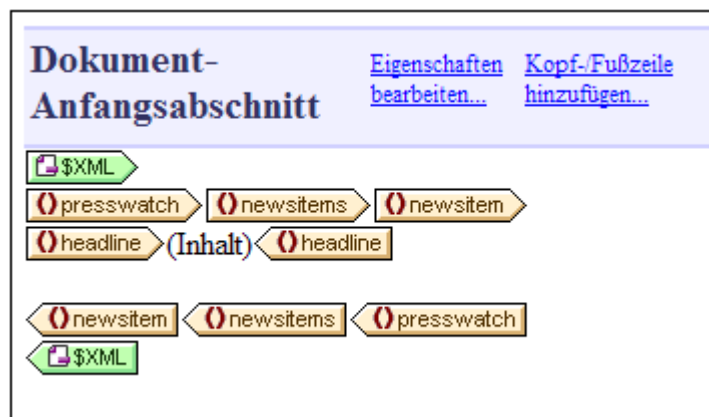
Sie sollten auch die Vorschau der [Authentic-Ansicht](#) und der RTF-Ausgabe.

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

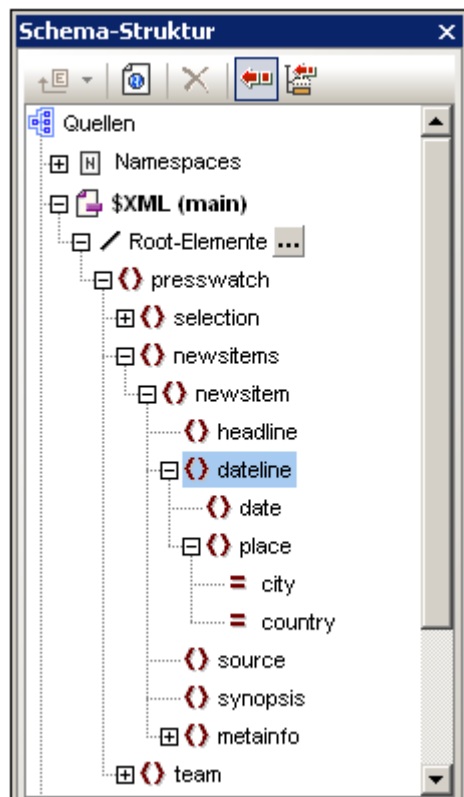
Einfügen von Attributinhalt

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 (Abbildung unten).



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

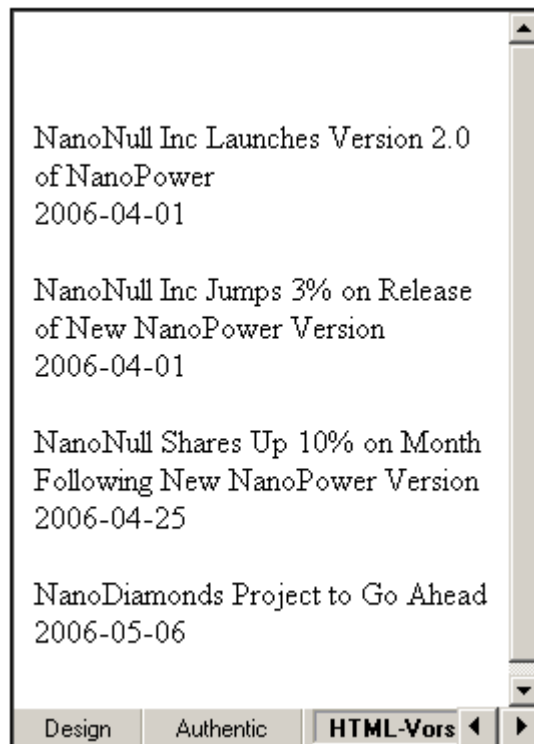


Beachten Sie, dass das Element `dateline` zwei Child-Elemente hat, nämlich `date` und `place` und dass das Element `place` zwei Attribute hat: `city` und `country`.

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".



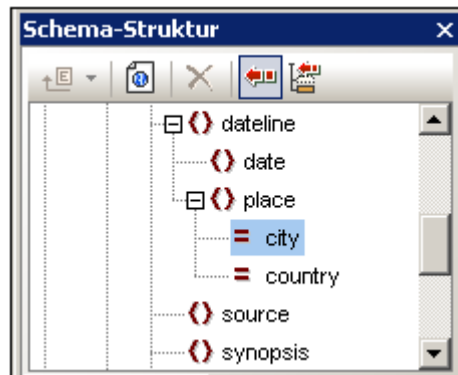
4. Wechseln Sie in die [HTML-Vorschau](#) und sehen Sie sich die Ausgabe von `dateline` genau an (Abbildung unten).



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 `city` und `country` enthalten sind und dass Attributinhalt nicht ausgegeben wird, wenn das übergeordnete Element des Attributs verarbeitet wird.

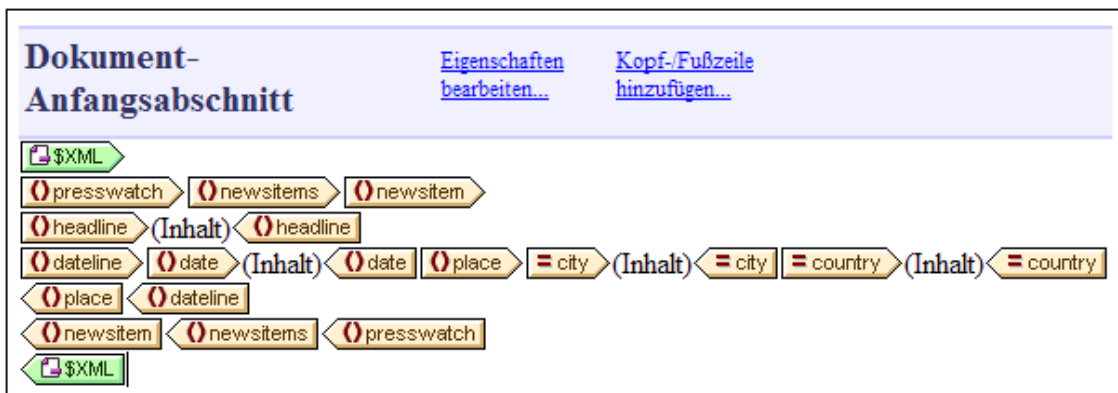
5. Gehen Sie in der Design-Ansicht zum Menübefehl [Authentic | Datumswahl automatisch hinzufügen](#) und deaktivieren Sie die Option, um das [automatische Hinzufügen der Datumswahl zu deaktivieren](#). (Das Symbol hat im deaktivierten Zustand keine Umrandung.) Dieser Schritt ist erforderlich, wenn die Datumswahl bei Einfügung eines Nodes vom Typ `xs:date` oder `xs:dateTime` in das Design (Beschreibung siehe nächster Schritt) nicht automatisch hinzugefügt werden soll.. Ziehen Sie das `date` Element aus der [Seitenleiste "Schema-Struktur"](#) zwischen den Start- und End-Tag des `dateline` Elements (und erstellen Sie es als Inhalt).
6. Wählen Sie in der [Seitenleiste "Schema-Struktur"](#) das `city` Attribut des `dateline/`

place Elements (Abbildung unten) aus.



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

Das SPS Design sollte anschließend etwa folgendermaßen aussehen:



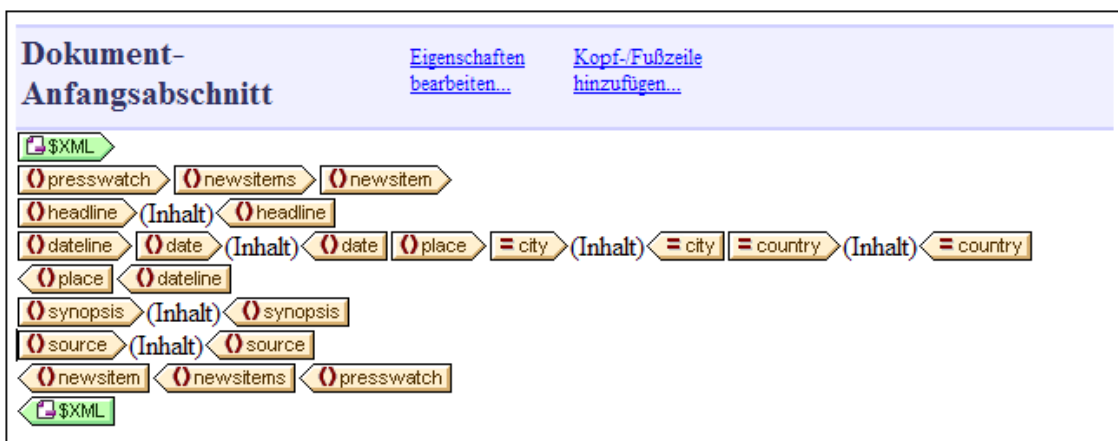
Die [HTML-Vorschau](#) sollte folgendermaßen aussehen:



Beachten Sie, dass 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](#), um die Ausgabe zu sehen. Dies ist ein wichtiger Punkt: Die Reihenfolge, in der Nodes in der [Hauptvorlage](#) angeordnet werden, gibt die [Struktur des Ausgabedokuments](#) vor.

Beachten Sie bitte auch, in welcher Form ein Node in das Design eingefügt wird. In der [HTML-Vorschau](#) 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](#) erfahren Sie, wie man einen Node als Auswahlliste erstellt und im Abschnitt [Verwendung von globalen Vorlagen und "Restlicher Inhalt"](#) 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.

6.3 Einfügen von statischem Inhalt

Statischer Inhalt ist Inhalt, den Sie direkt in das Design 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

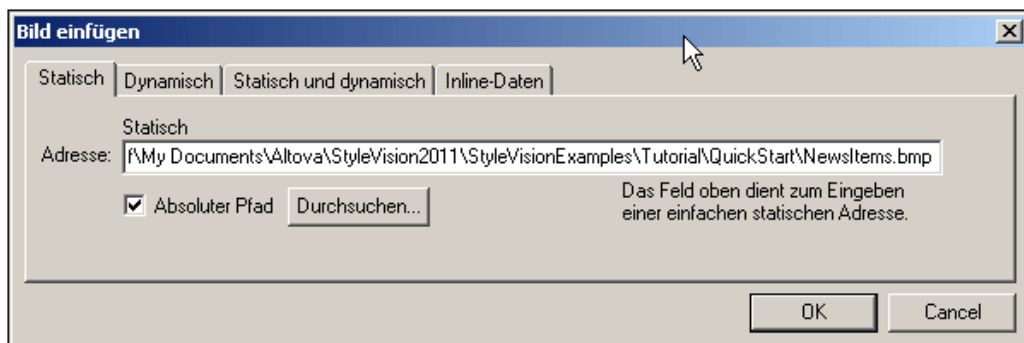
Das statische Bild, das wir einfügen werden, befindet sich im [Verzeichnis \(Eigene\) Dateien](#): c: / Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples\Tutorial\QuickStart\NewsItems.BMP. Es wird als Dokumentüberschrift verwendet. Um dieses Bild ganz oben im Dokument einzufügen, gehen Sie folgendermaßen vor:

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



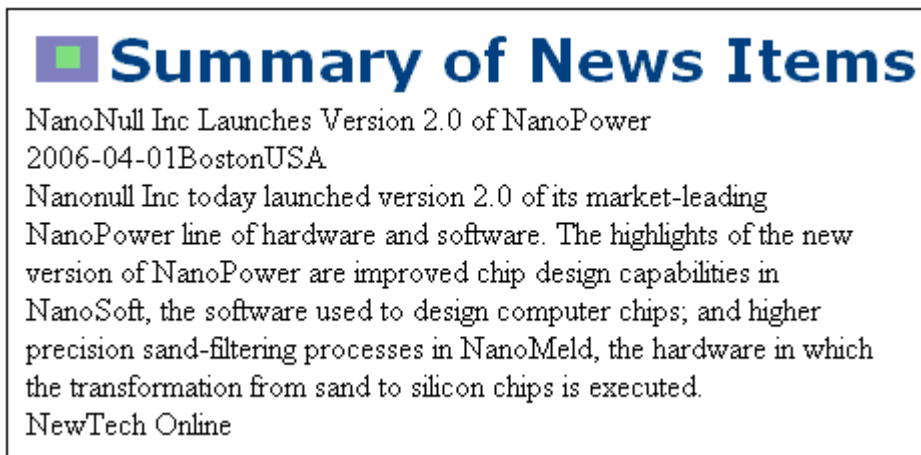
Beachten Sie, dass sich der Cursor innerhalb des Elements `newsitems` jedoch außerhalb des Elements `newsitem` befindet. Das Bild wird daher einmal in die Ausgabe eingefügt, und zwar am Beginn der Verarbeitung des `newsitems` Elements (da im Schema nur ein `newsitems` Element definiert ist).

2. Rechtsklicken Sie und wählen Sie den Befehl [Einfügen | Bild](#). Daraufhin wird das Dialogfeld "Bild einfügen" eingefügt (Screenshot unten).



3. Aktivieren Sie auf dem Register "Statisch" die Option "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](#). 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](#) ausgewählt ist, in der [Seitenleiste "Eigenschaften"](#) (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](#) ausgewählt ist, in der [Seitenleiste "Stil"](#) 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](#).

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.

Einfügen von statischem Text

Sie haben zu Ihrem Design bereits statischen Text hinzugefügt (im Abschnitt [Einfügen von dynamischem Inhalt \(aus der XML-Quelle\)](#)). 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:

Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower
 2006-04-01BostonUSA
 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.
 NewTech Online

Beachten Sie: In der Ausgabe des Elements `dateline` wird der Inhalt des Elements `date` und der Attribut `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)



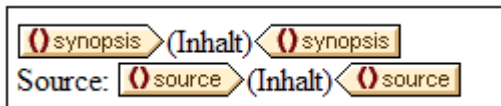
Dieser Teil der Ausgabe sieht nun folgendermaßen aus:

Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower
 2006-04-01: Boston, USA

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.

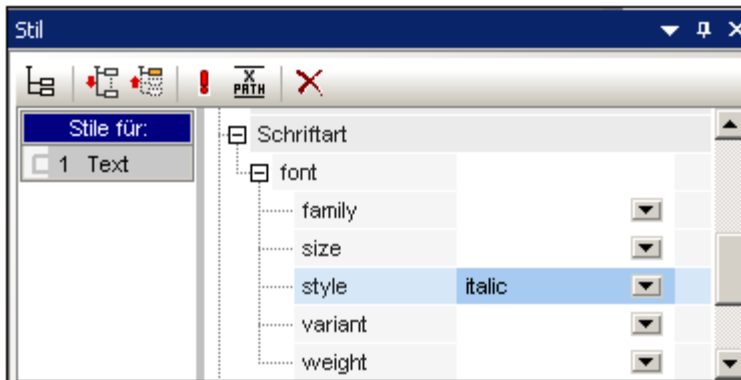
Nun wollen wir ein weiteres statisches Textelement hinzufügen. Geben Sie im Design genau vor dem Start-Tag des Elements `source` den String "Source: " ein (siehe Abbildung unten).



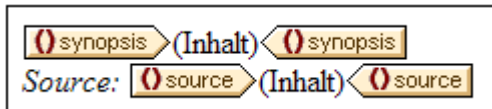
Formatieren von statischem Text

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"](#) (Abbildung unten) die Komponente **1 Text**

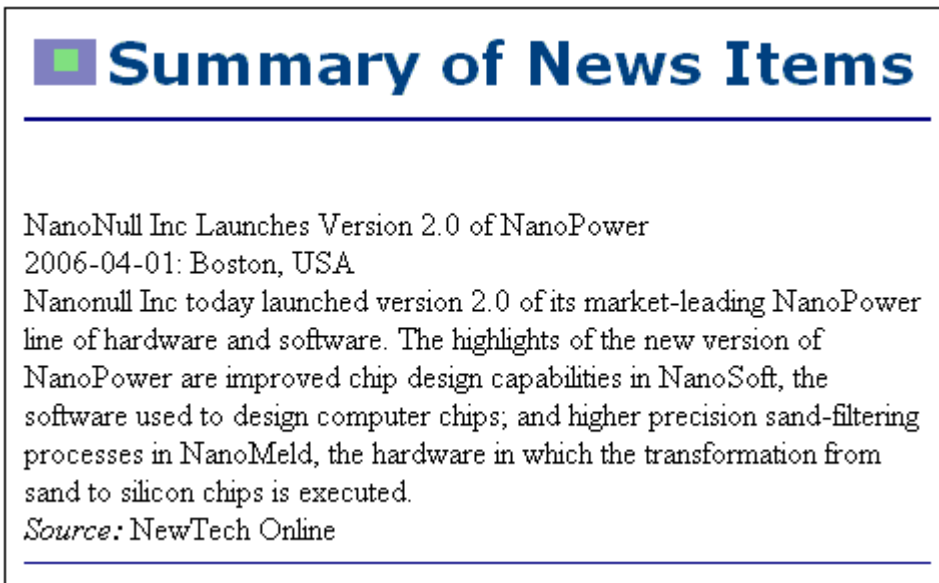
ausgewählt ist. Erweitern Sie nun die Eigenschaftsgruppe *Schriftart* wie in der Abbildung unten gezeigt und wählen Sie für die Eigenschaft `font-style` die Option *italic* aus dem Dropdown-Menü aus.



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:



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.

6.4 Formatieren des Inhalts

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

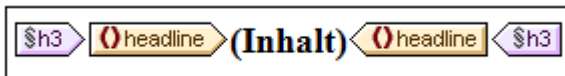
- CSS-Stilregeln können sowohl für Blockkomponenten als auch für Komponenten innerhalb von Zeilen definiert werden.
- [Vordefinierte Formate](#) sind Blockkomponenten, für die bereits Stile definiert sind und die als "Hülle" für eine Gruppe benachbarter Komponenten verwendet werden können, die als Block behandelt werden sollen. Die Stile dieser vordefinierten Formate können durch benutzerdefinierte Stile außer Kraft gesetzt werden.
- Klassenattribute können für Komponenten im Design deklariert werden und die Klasse kann als Selektor für [externe](#) oder [globale](#) Stilregeln verwendet werden.
- Sie können auf drei Ebenen Stile definieren und zwar, nach Priorität gereiht: (i) Stilregeln in [externen Stylesheets](#), (ii) [globale Stilregeln](#) und (iii) [lokale Stilregeln](#). Beachten Sie allerdings, dass bestimmte Arten von Selektoren in externen und globalen Stilregeln, wie z.B. namensbasierte Selektoren (`h1`, `a`, `img`, usw.) nur auf die Authentic-Ansicht und die HTML-Ausgabe angewendet werden können, nicht aber auf die RTF-Ausgabe. Regeln, die Klassenattribute haben, sind auf HTML-, RTF- und PDF-Formate anwendbar.

In diesem Abschnitt lernen Sie Folgendes:

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

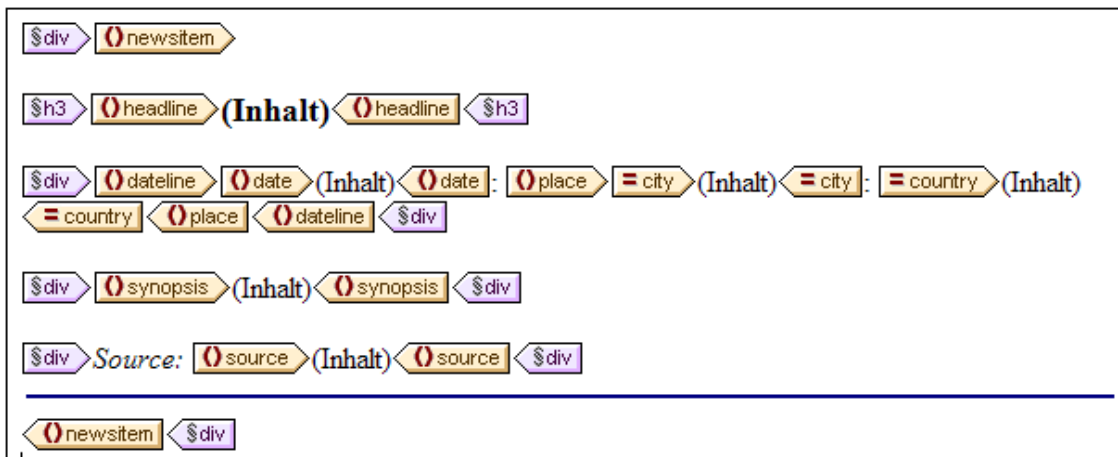
Zuweisen vordefinierter Formate

Ein Grund für die Zuweisung eines [vordefinierten Formats](#) ist, einer Komponente die Stileigenschaften dieses [vordefinierten Formats](#) zu geben. Wählen Sie im Design das Element `headline` aus und wählen Sie anschließend den Befehl **Einfügen | 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.

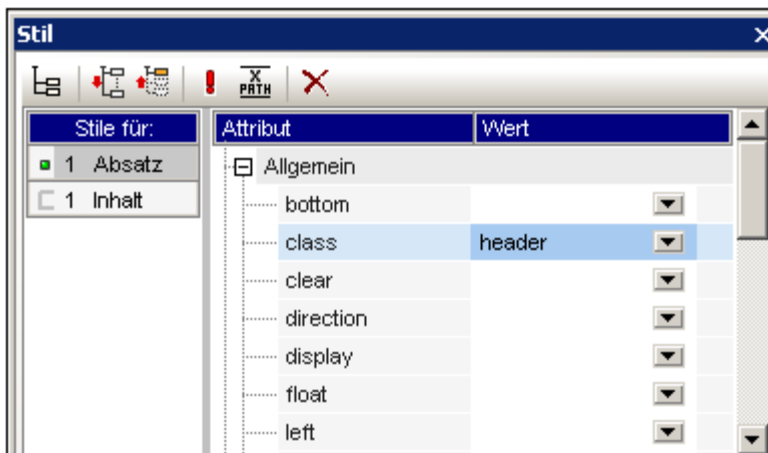


Sie haben Komponenten nun in verschiedenen `div` Blöcken gruppiert. [Weiter hinten in diesem Abschnitt](#) erfahren Sie, wie man solchen Blöcken gruppierter Komponenten Stile zuweist.

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 einzigen 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 Globale 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 "Globale Stile"](#) (in der Design-Ansicht) folgendermaßen vor:

1. Wählen Sie den Eintrag `Extern`.
2. Klicken Sie in der Symbolleiste auf die Schaltfläche **Hinzufügen**. Daraufhin wird das Dialogfeld "Öffnen" geöffnet..
3. Suchen Sie nach der Datei `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial\QuickStart\QuickStart.css`, die sich im [Verzeichnis \(Eigene\) Dateien](#) 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

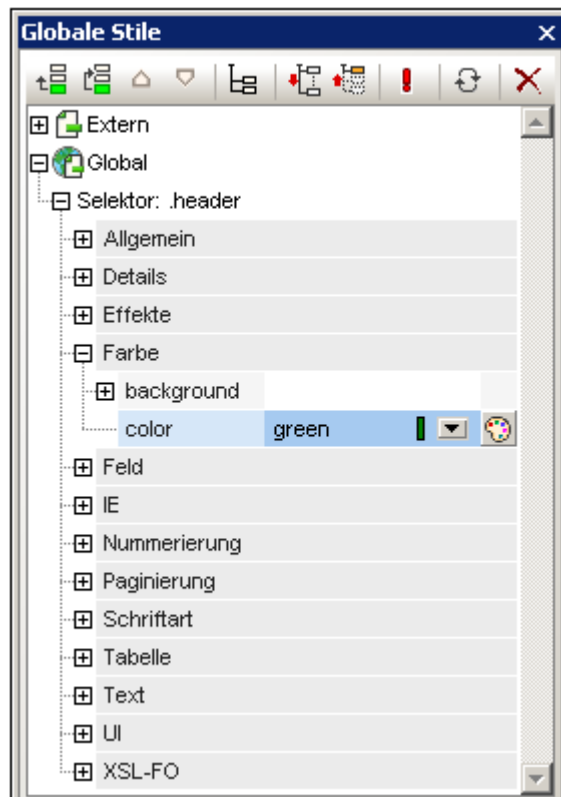
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld,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

Source: NewTech Online

Definieren globaler Stilregeln

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

1. Wählen Sie bei aktiver [Design-Ansicht](#) in der [Seitenleiste "Globale Stile"](#) den Eintrag "Global".
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 Stylesheet 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`).

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 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"](#) 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"](#) 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"](#) 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*). 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"](#), während in der Spalte "Stile für" der Eintrag 1 Absatz ausgewählt ist, die Eigenschaft `color` in der Eigenschaftsgruppe *color* auf `gray` (*Abbildung unten*). Setzen Sie in der Eigenschaftsgruppe *Schriftart* 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*) 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*).

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](#) den Eintrag Absatz (p). Daraufhin werden dem Block die vordefinierten Stile des HTML-Elements `p` zugewiesen. Die HTML-Vorschau sollte nun etwa folgendermaßen aussehen:

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

Wenn Sie damit fertig sind, speichern Sie die Datei.

6.5 Verwenden von automatischen Berechnungen

[Automatische Berechnungen](#) 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.

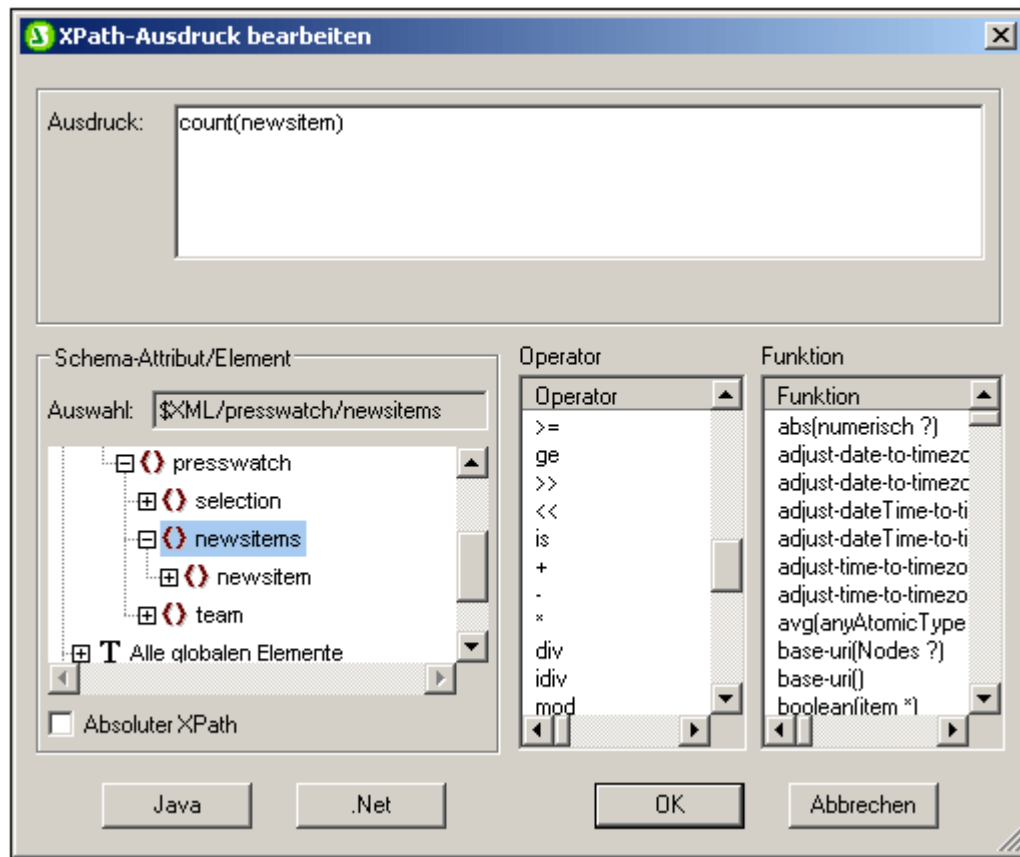
Zählen der "newsitem" Nodes

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).



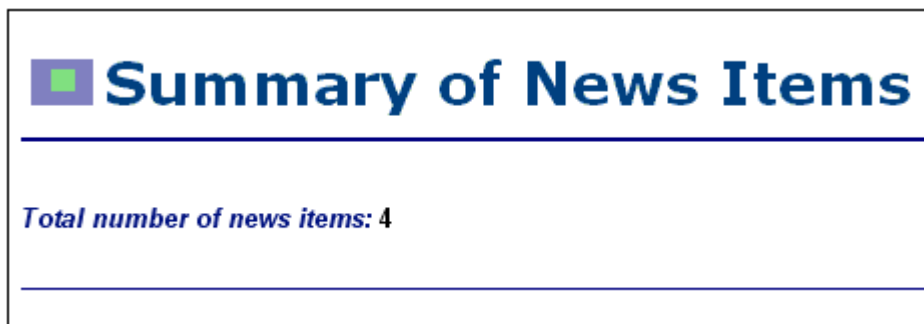
2. Geben Sie, wie in der Abbildung oben gezeigt, den statischen Text "Total number of news items: " ein.
3. Wenden Sie auf den statischen Text lokale Stileigenschaften Ihrer Wahl an. Ein Anleitung dazu finden Sie im Abschnitt [Formatieren des Inhalts](#).
4. Platzieren Sie den Cursor hinter den Doppelpunkt 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.)



5. Beachten Sie, dass der Kontext-Node in der Schemastruktur der Node `newsittem` ist. Dieser Node erscheint markiert. Geben Sie nun im Textfeld "Ausdruck" entweder den Ausdruck `count(newsittem)` ein oder erstellen Sie den Ausdruck mit Hilfe der Seitenleisten. (Doppelklicken Sie auf die Funktion `count`, um Sie zu übernehmen, setzen Sie anschließend den Cursor zwischen die Klammern der Funktion und doppelklicken Sie in der Schemastruktur auf den `newsittem` Node.
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](#).



Ihre HTML-Ausgabe sieht folgendermaßen aus:

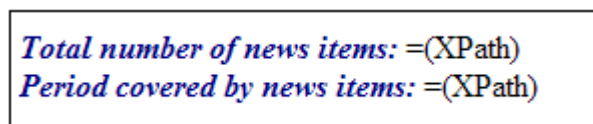


Anzeigen des Zeitraums der Nachrichtenmeldungen

Der Zeitraum, in dem die Nachrichtenmeldungen 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:

Total number of news items: 4
Period covered by news items: 4/2006 to 5/2006

Wenn Sie damit fertig sind, speichern Sie die Datei.

6.6 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. Sie können im SPS Design eine Auswahlliste erstellen, die mit eindeutigen `relevance` Elementwerten gefüllt ist. Wenn der Benutzer der Authentic-Ansicht ein Element aus der Dropdown-Liste im Auswahllistenfeld auswählt, so kann dieser Eintrag an einen Node im XML-Dokument übergeben werden. Mit Hilfe einer Bedingung kann überprüft werden, was der Benutzer ausgewählt hat (indem dieser Node überprüft wird) und es wird daraufhin die entsprechende Verarbeitung (Anzeige) für die Auswahl des Benutzers gewählt. In diesem Abschnitt werden Sie nun eine Bedingungs-vorlage erstellen, in der jene Nachrichtenmeldungen dargestellt werden, die ein `relevance` Element haben, das der Selektion des Benutzers entspricht.

Dabei gehen wir folgendermaßen vor:

1. Erstellung einer Auswahlliste, in der der Benutzer der Authentic-Ansicht den `byrelevance`-Wert auswählen kann. 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 Bedingungs-vorlage bezeichnet.
3. Innerhalb der Bedingungs-vorlage 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 Bedingungs-vorlage.
5. In die Bedingung für das `newsitem` Element wird eine Verzweigung eingefügt, die alle Nachrichtenmeldungen auswählt.

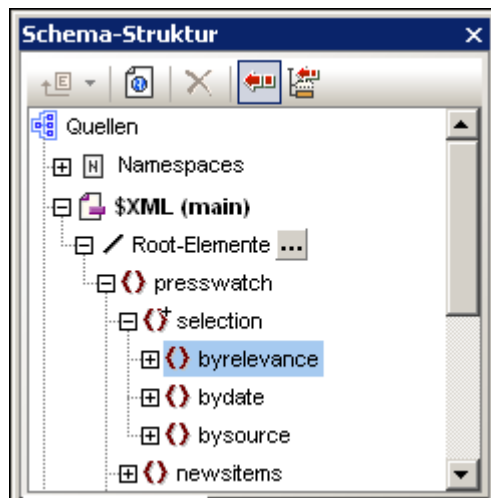
Erstellen der Auswahlliste zur Auswahl eindeutiger Node-Werte

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

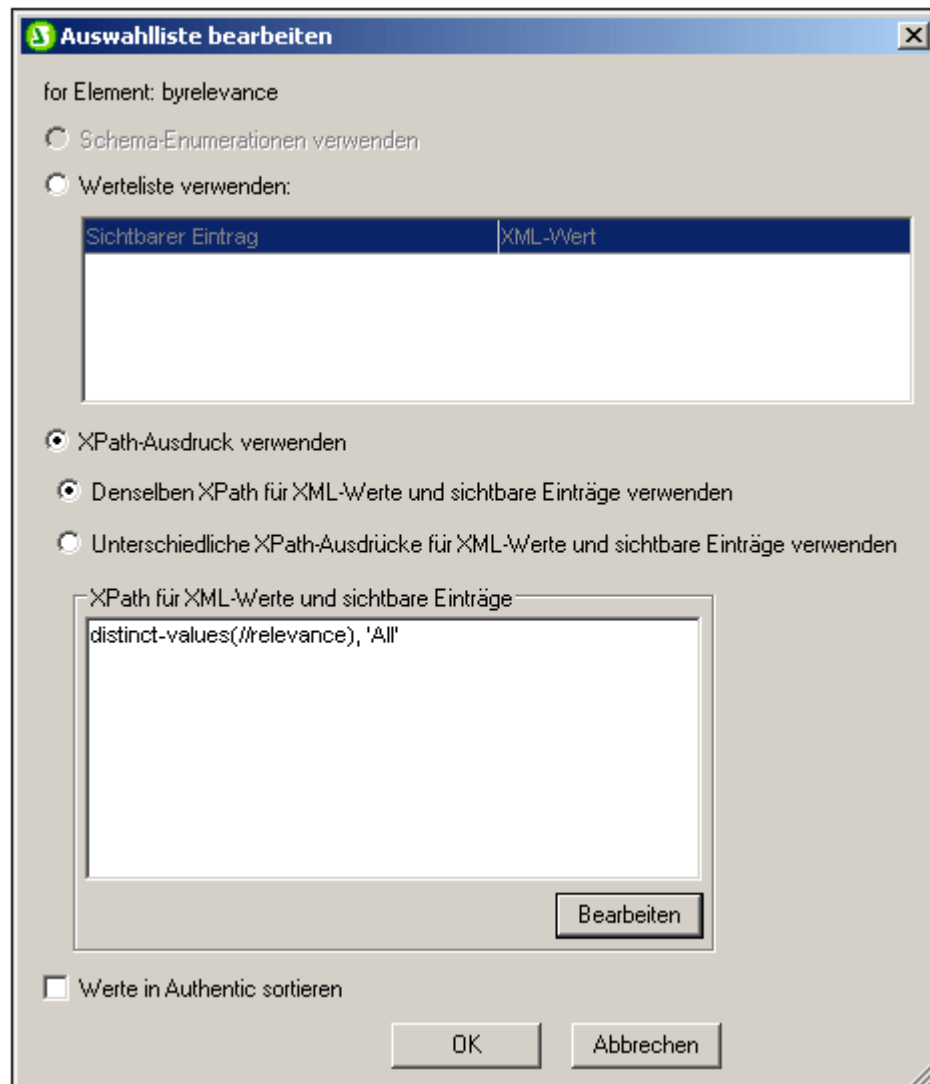
1. Fügen Sie den statischen Text "Select by relevance: " oben im Dokument knapp unterhalb der [zweiten automatischen Berechnung](#) ein (*Abbildung unten*).

Total number of news items: =(XPath)
Period covered by news items: =(XPath)
Select by relevance:

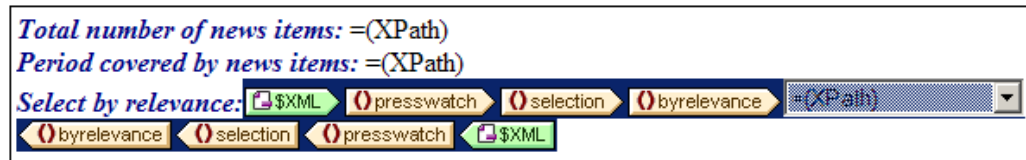
2. Ziehen Sie den `byrelevance`-Node aus der [Seitenleiste "Schema-Struktur"](#) (*Abbildung unten*) hinter den neu eingegebenen statischen Text.



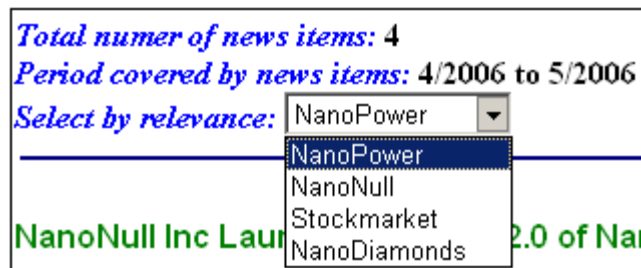
3. Wählen Sie im Kontextmenü, das daraufhin angezeigt wird, den Befehl "Auswahlliste erstellen". Das folgende Dialogfeld wird angezeigt.



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.
- Klicken Sie zum Abschluss auf **OK**. Die Auswahlliste wird eingefügt und das Design sieht daraufhin etwa folgendermaßen aus:



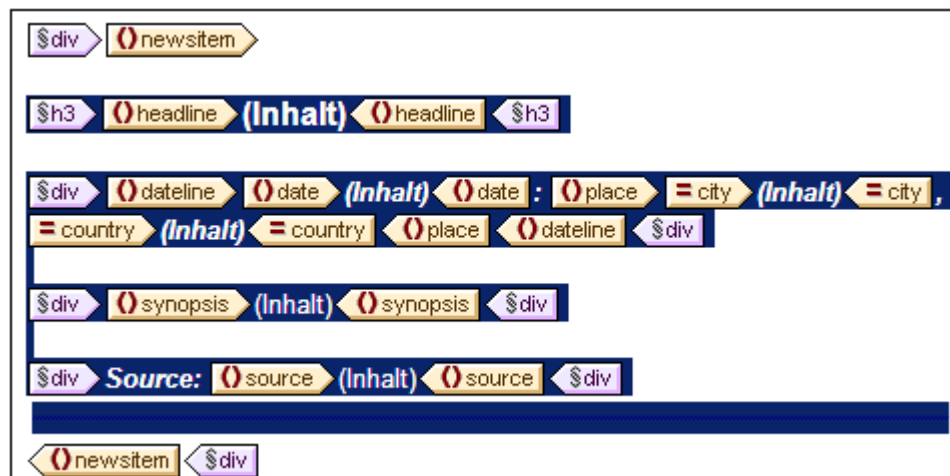
6. Wechseln Sie in die [Authentic-Ansicht](#). Wenn Sie im Auswahllistenfeld auf den Abwärtspfeil klicken, beachten Sie, dass die Liste die eindeutigen Werte aller `relevance` Nodes enthält (*Abbildung unten*). Vergleichen Sie dies mit dem XML-Dokument. Dies ist eine dynamische Liste, die jedes Mal,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



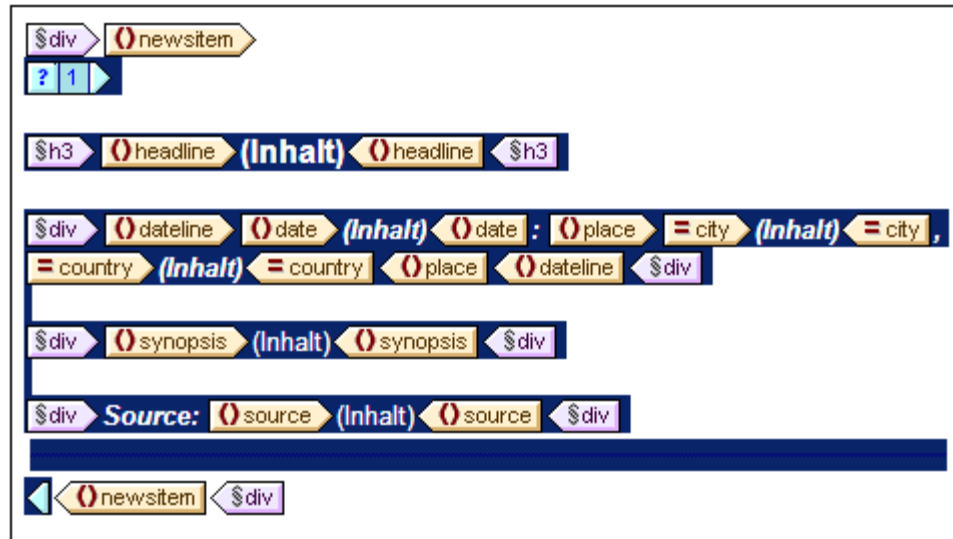
Einfügen einer Bedingung, um Nachrichtenmeldungen anzuzeigen, die die ausgewählte `relevance` haben

Die Bedingung wählt `newsitem`-Elemente aus, die ein `metainfo/relevance`-Element mit einem Wert haben, der mit dem vom Benutzer ausgewählten (und an das `/presswatch/selection/byrelevance`-Element übergebenen) Wert 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).



- Wählen Sie den Menübefehl (oder den Kontextmenübefehl) [Einschließen in Bedingung](#). Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" geöffnet.
- 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.
- Klicken Sie auf **OK**. Die Bedingung wird rund um den Inhalt des `newsitem`-Elements erstellt (Abbildung unten).

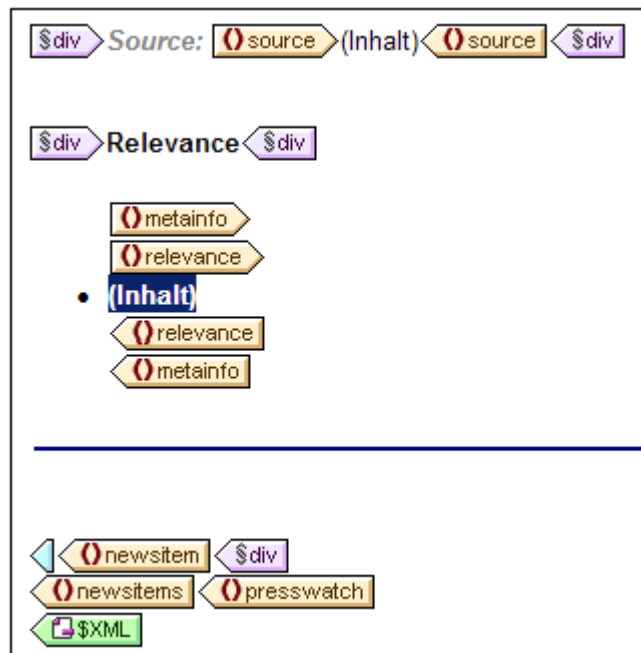


Beachten Sie, dass es in dieser Bedingung nur einen Branch gibt. Nachrichtenmeldungen, für die die Auswertung der Bedingung "true" ergibt, werden angezeigt, jene, für die die Auswertung nicht "true" ergibt, werden nicht angezeigt. Die Bedingung funktioniert also in diesem Fall als Filter. Weiter unten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

- Erstellen Sie unterhalb der `div`-Komponente für das `source`-Element und innerhalb des End-Tags der bedingten Vorlage einen vertikalen Abstand.
- 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).
- Ziehen Sie das `relevance`-Element aus der Root-Elemente-Struktur in der [Seitenleiste "Schema-Struktur"](#) in das Design unterhalb des statischen Texts `Relevance: .`
- 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.)
- Wenden Sie auf den Inhalt der Liste eine Textformatierung an. Wenn Sie fertig sind, sollte das Design ungefähr folgendermaßen aussehen:

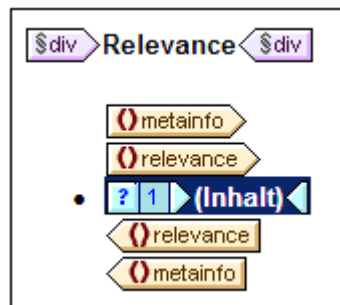


Überprüfen Sie nun in der Authentic-Ansicht, die Ergebnisse der unterschiedlichen Auswahlen von `relevance`; über die Auswahlliste können Sie die Auswahl ändern.

Fettformatierung des ausgewählten `relevance`-Elements

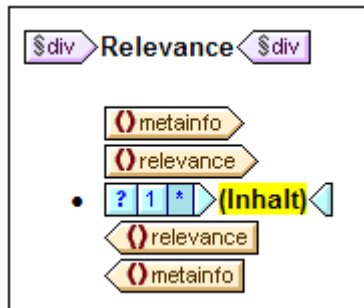
Einige Nachrichtenmeldungen haben mehr als ein `relevance`-Element. In solchen Fällen wäre es übersichtlicher, wenn das `relevance`-Element, das mit der Auswahl des Benutzers übereinstimmt, markiert erscheint und die anderen nicht. Dies lässt sich folgendermaßen bewerkstelligen:

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 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" das Kontrollkästchen "Andernfalls" (unterhalb des Textfelds für den Ausdruck).

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.



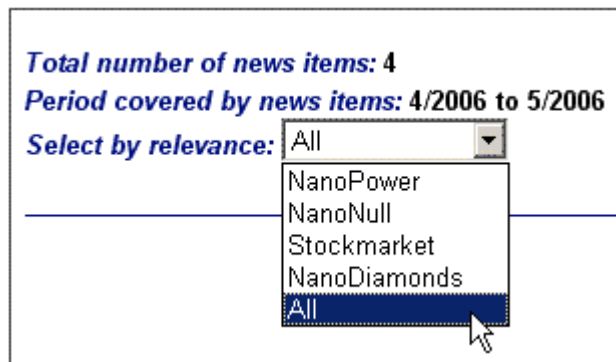
7. Beachten Sie, dass der Inhalt des *Otherwise*-Branch eine Kopie des ersten Branch ist; der Platzhalter `Inhalt` erscheint fett und mit gelbem Hintergrund. Entfernen Sie diese Formatierung (fett und Hintergrundfarbe Schwarz) aus dem Platzhalter `Inhalt`.

Sie haben eine Bedingung mit zwei Verzweigungen (jede mit ihrer Bedingungsvorlage) gesetzt, die an jedem `relevance`-Element die folgende Überprüfung durchführt: (i) 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 Authentic-Ansicht

Ändern der Auswahlliste und Einfügen einer zweiten Bedingungsverzweigung

In der Auswahlliste, in der der Benutzer der Authentic-Ansicht einen `byrelevance` Wert auswählt, 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 *Authentic*) auf die Schaltfläche **Bearbeiten** der Eigenschaft *Ursprung des Inhalts*.
3. Ändern Sie im nun angezeigten Dialogfeld "XPath-Ausdruck bearbeiten" 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 Authentic-Ansicht (*Abbildung unten*).



Wenn der Benutzer nun `All` auswählt, wird dieser Wert (`All`) an den Node übergeben / `presswatch/selection/byrelevance`. 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", 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 erste Verzweigung. Sie gibt die Vorlage für die Nachrichtenmeldungen aus, wenn der Benutzer die Option `All` auswählt.

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

6.7 Verwendung von globalen Vorlagen und "Restlicher Inhalt"

[Globale Vorlagen](#) 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 globalen Vorlage können in die lokale Vorlage kopiert werden.
- Alternativ dazu kann eine lokale Vorlage (in der Hauptvorlage) 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 aufgerufen oder verwendet wird.

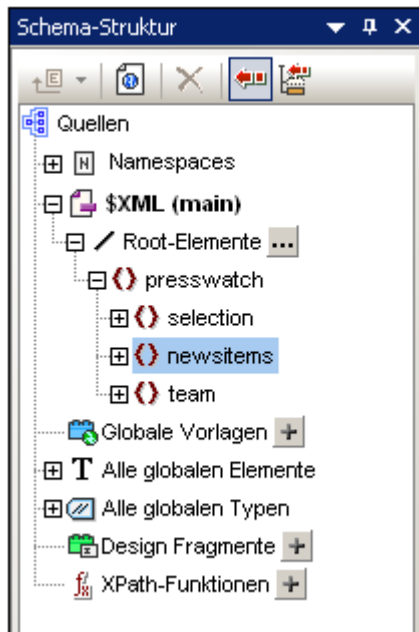
Es gibt zwei Methoden, um eine globale Vorlage von der Hauptvorlage aus aufzurufen.

- 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.)

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

Einfügen der Anweisung `restlicher Inhalt`

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



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** auswählen. Der Platzhalter `restlicher Inhalt` wird eingefügt (Abbildung unten).



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

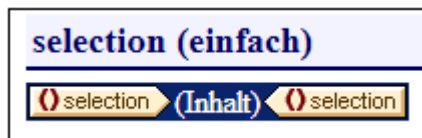
```
AllAndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.edwar
```

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"](#) 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).

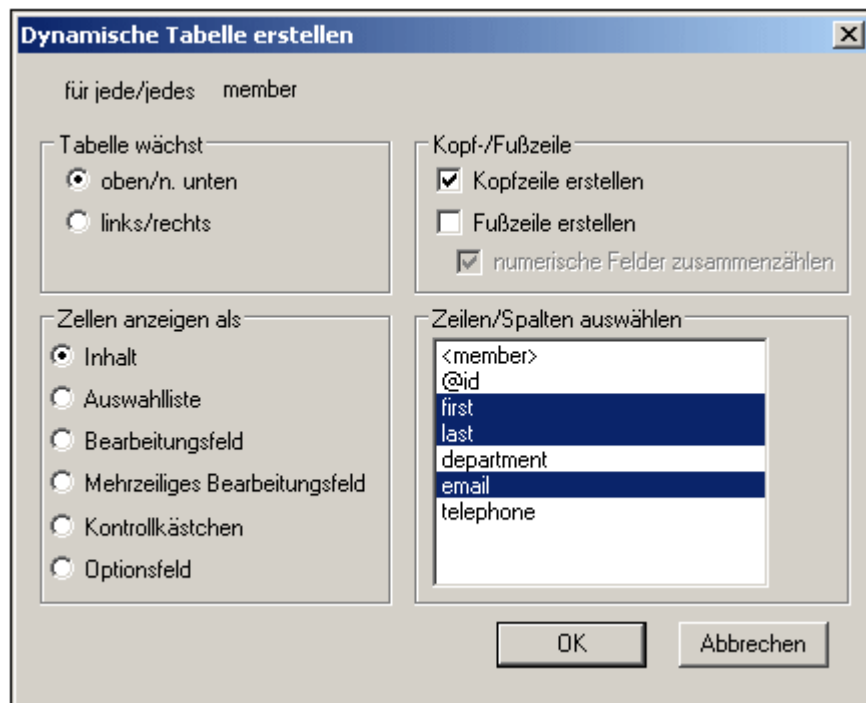
```
AndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.e
```

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

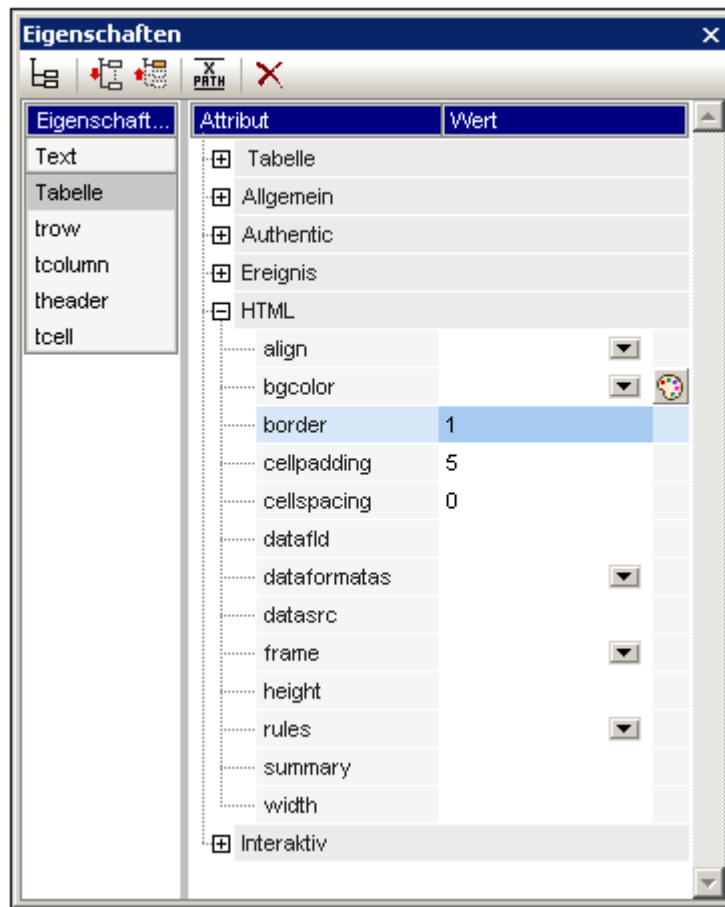
Erstellen einer globalen Vorlage für `team/member`

Ziel ist es, eine Tabelle zu erstellen, in der die Informationen über die Mitarbeiter des Pressteams 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"](#), während `Tabelle` in der Spalte "Eigenschaften für" ausgewählt ist, die Tabelleneigenschaften, wie in der Abbildung unten gezeigt, an.



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

6.8 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](#) ([Hauptvorlage](#) und [globale Vorlagen](#)).
- Wie man unter Verwendung der verschiedensten dynamischen und statischen SPS-Komponenten [dynamischen](#) und [statischen](#) Inhalt in das Design einfügt.
- Wie man [CSS-Stile](#) in [externen Stylesheets](#), [globalen Stilregeln](#) und in [lokalen Stilregeln](#) verwendet.
- Wie man mittels einer [automatischen Berechnung](#) zusätzliche Informationen aus den verfügbaren XML-Daten extrahiert.
- Wie man die XML-Daten mittels [Bedingungen](#) filtert und wie man je nach den Werten in den XML-Daten unterschiedliche Ausgabeergebnisse erhält.
- Wie man [globale Vorlagen](#) und ["restlicher Inhalt"](#) verwendet.

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

- [SPS-Datei: Inhalt](#)
- [SPS-Datei: Struktur](#)
- [SPS-Datei: Komplexe Funktionen](#)
- [SPS-Datei: Darstellung](#)
- [SPS-Datei: Zusätzliche Funktionen](#)

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

Verwendung des SPS

Nachdem Sie das SPS fertiggestellt haben, sollten Sie auch die wichtigsten Verwendungsweisen eines SPS ausprobieren:

- Bearbeitung von XML-Dokumenten in der Authentic-Ansicht von XMLSpy oder Authentic Desktop. (Die Enterprise Edition und die Professional Edition enthalten ein Register für eine Vorschau auf die Authentic-Ansicht, das einige Funktionen wie z.B. Seitenleisten und Textstatussymbole nicht hat.) Diese beiden Produkte enthalten eine Authentic-Ansicht mit allen Funktionen. Sie können darin die Seitenleisten und das Kontextmenü ausprobieren. Um die Datei `QuickStart.xml` in der Authentic-Ansicht in XMLSpy oder Authentic Desktop zu bearbeiten, verknüpfen Sie die XML-Datei mit `MyQuickStart.sps` und wechseln Sie in die Authentic-Ansicht.
- Generierung von XSLT Stylesheets zur Transformation der XML-Datei in eine HTML-Ausgabe. Die XSLT Stylesheets können mit Hilfe des Befehls [Datei | Generierte Dateien speichern](#) oder über die [Befehlszeile](#) generiert werden. Versuchen Sie XSLT Stylesheets anhand von `MyQuickStart.sps` zu generieren und die Datei `QuickStart.xml` dann mit Hilfe dieser Stylesheets zu transformieren.

Kapitel 7

Übersicht über die Verwendung

7 Übersicht über die Verwendung

Ziele

SPS-Dokumente, die Sie in StyleVision erstellen, können grundsätzlich für zwei Dinge verwendet werden:

- Um festzulegen, wie das XML-Quelldokument in der Authentic-Ansicht angezeigt werden soll, und um die Eingabe von Daten in XML-Dokumente oder Datenbanken über die Benutzeroberfläche der Authentic-Ansicht zu ermöglichen.
- Um XSLT Stylesheets für die HTML- und RTF-Ausgabe zu erstellen.

Auf diese Art kann das SPS dazu verwendet werden, um die Bearbeitung von XML-Dokumenten zu ermöglichen und um anhand des editierten XML-Dokuments HTML- und RTF-Ausgabedokumente zu generieren. Zusätzlich dazu können Sie mit Hilfe der generierten XSLT Stylesheets auch andere XML-Dokumente transformieren, die auf demselben Schema basieren wie das SPS.

Schritte beim Erstellen eines SPS

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

1. [Zuweisen eines Schemas](#) 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"](#). Alternativ dazu kann mit Hilfe des Befehls **Datei | Neu** ein neues SPS direkt mit einem Schema erstellt werden.
2. [Zuweisen einer XML-Arbeitsdatei](#) zum SPS. Die [XML-Arbeitsdatei](#) stellt die XML-Daten zur Verfügung, die vom SPS beim Generieren der Authentic-Ansicht bzw der Ausgabeansichten verarbeitet werden. Die [XML-Arbeitsdatei](#) wird in der [Seitenleiste "Design-Übersicht"](#) 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](#).
4. Auswahl der [Internet Explorer-Kompatibilität](#). Diese muss mit der Version des installierten Internet Explorers übereinstimmen.
5. Das SPS Dokument wird in der [Design-Ansicht](#) unter Verwendung der verschiedenen verfügbaren Design-Komponenten erstellt. Der [Design-Prozess](#) besteht darin, dass eine Dokumentstruktur erstellt wird und [Darstellungseigenschaften](#) definiert werden.
6. Die Authentic-Ansicht und 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](#) benötigt werden, werden diese [generiert](#).
8. Falls erforderlich, weisen Sie eine XML-Vorlagendatei zu. Die [XML-Vorlagendatei](#) liefert die Ausgangsdaten für ein neues XML-Dokument, das mit Hilfe des SPS in der Authentic-Ansicht bearbeitet werden kann.
9. Das SPS wird mehreren Benutzern der Authentic-Ansicht zur [Verwendung zur Verfügung gestellt](#).

7.1 SPS und Quellen

Erstellen einer neuen SPS-Datei

Um ein neues SPS Dokument zu erstellen, wählen Sie eine Option unter [Datei | Neu \(Strg+N\)](#) oder klicken Sie in der [Standardsymbolleiste](#) 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 drei Arten von Quelldateien, die einem SPS zugewiesen werden können:

- [Schemaquellen](#)
- [XML-Arbeitsdatei](#)
- [XML-Vorlagendatei](#)

Diese Zuweisungen erfolgen in der [Seitenleiste "Design-Übersicht"](#). Eine Beschreibung dazu finden Sie im Abschnitt [Design-Übersicht](#). 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"](#) zugewiesen. Beim Schema kann es sich um eine XML-Schemadatei (`.xsd` Datei), ein von einer XML-Datei generiertes XML-Schema, ein anhand von einer Datenbankdatei generiertes XML-Schema, eine DTD oder ein benutzerdefiniertes Schema handeln. Für jedes Schema können eine optionale [XML-Arbeitsdatei](#) und eine optionale [XML-Vorlagendatei](#) 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

Ein SPS kann optional eine damit verknüpfte [XML-Arbeitsdatei](#) haben. Die Funktion der [XML-Arbeitsdatei](#) 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](#) wird in der [Seitenleiste "Design-Übersicht"](#) zugewiesen.

XML-Vorlagendatei

Ein SPS kann optional mit einer [XML-Vorlagendatei](#) verknüpft werden. Die Funktion der [XML-Vorlagendatei](#) ist es, die Ausgangsdaten für das neue XML-Dokument bereitzustellen, das jedes Mal erstellt wird, wenn das SPS in der [Authentic-Ansicht](#) eines anderen Produkts als StyleVision geöffnet wird. Die [XML-Vorlagendatei](#) muss entsprechend dem damit verknüpften Schema gültig sein. Sie wird in der [Seitenleiste "Design-Übersicht"](#) zugewiesen.

7.2 Erstellen des Designs

Im SPS Design werden folgende Punkte definiert:

1. [Welcher Inhalt](#) (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](#); und
3. [Welche Darstellungseigenschaften \(Formatierung\)](#) 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 oder der Datenbank, auf das/die das SPS angewendet wird. Inhalt aus dem [XML-Dokument](#) wird durch Ziehen des gewünschten XML-Daten-Node aus der jeweiligen Schemastruktur in der [Seitenleiste "Schema-Struktur"](#) an die gewünschte Stelle ins SPS eingefügt.
2. einem externen XML-Dokument, auf das die Applikation (also StyleVision oder ein Produkt mit der [Authentic-Ansicht](#)) 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](#) 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 1.0 und XPath 2.0 Ausdrücken vorgenommen werden. Bearbeitungen werden normalerweise durch [automatische Berechnungen](#) vorgenommen.
5. Bei der HTML-Ausgabe kann Inhalt mittels [JavaScript-Funktionen](#) generiert werden.

Struktur der Ausgabe

Im SPS Design kann die [Struktur des Ausgabedokuments](#) auf folgende Arten gesteuert werden: (i) entweder durch einen prozeduralen Ansatz. Dabei wird die Struktur des Ausgabedokuments in einer [Vorlage](#) (der StyleVision [Hauptvorlage](#)) 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](#) ([globale Vorlagen](#) 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](#) und [globalen Vorlagen](#) erstellen. [Modulare SPSs](#) und [Design-Fragmente](#) 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](#) auf Komponenten angewendet werden. Die Definition von Darstellungseigenschaften wird im Abschnitt [Darstellungseigenschaften](#) näher beschrieben.

7.3 XSLT- und XPath-Versionen

Ein SPS ist im Grunde ein XSLT Stylesheet. Für jedes SPS muss die XSLT-Version angegeben werden: 1.0 oder 2.0. Dies geschieht, indem Sie in der Symbolleiste auf die entsprechende

Schaltfläche klicken:  oder . Die Auswahl, die Sie vornehmen, legt zweierlei fest:

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

XSLT-Transformationen

XSLT-Transformationen in StyleVision dienen dazu um: (i) auf der Benutzeroberfläche eine [Ausgabevorschau](#) zu erstellen; und (ii) um [über die Benutzeroberfläche](#) und über die [Befehlszeile Ausgabedateien zu generieren und zu speichern](#) (HTML und RTF). Der für die Transformationen verwendete XSLT-Prozessor (Altova XSLT 1.0 Prozessor oder Altova XSLT 2.0 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 und XPath 2.0 für XSLT 2.0) in Zusammenhang. XPath-Ausdrücke werden in StyleVision häufig verwendet - v.a. in Funktionen wie [automatischen Berechnungen](#) und [Bedingungsvorlagen](#). 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.

7.4 Kompatibilität mit Internet Explorer

Damit das SPS-Design (in der Design-Ansicht) und die Ausgabevorschau (in der Authentic-Ansicht und 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](#).
- Unterstützung für die unten aufgelisteten neuen CSS-Stile (u.a. auch CSS 3-Stile). Die Anwendung dieser Stile ist auf die Authentic-Ansicht und 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](#) 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](#)" 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 IE 9 bei Auswahl von IE 9 installiert sein muss, damit die unterstützten CSS-Stile auf der Design-Oberfläche zur Verfügung stehen.) Wenn z.B. IE 9 installiert und als kompatible Version ausgewählt ist, so stehen auf der Design-Oberfläche die in

- IE 9 unterstützten CSS 3-Stile zur Verfügung.
- Die HTML-Elemente der entsprechenden ausgewählten IE-Version können als [vordefinierte Formate](#) oder als [benutzerdefinierte Elemente](#) eingegeben werden. Das HTML-Element wird in der Authentic-Ansicht und der HTML-Vorschau so dargestellt, wie die installierte IE-Version das Element unterstützt. Wenn z.B. IE 9 installiert und als kompatible Version ausgewählt ist, so werden die unterstützten HTML5-Elemente in der Authentic-Ansicht und 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](#).

Kompatibilität älterer SPS-Designs mit IE 9

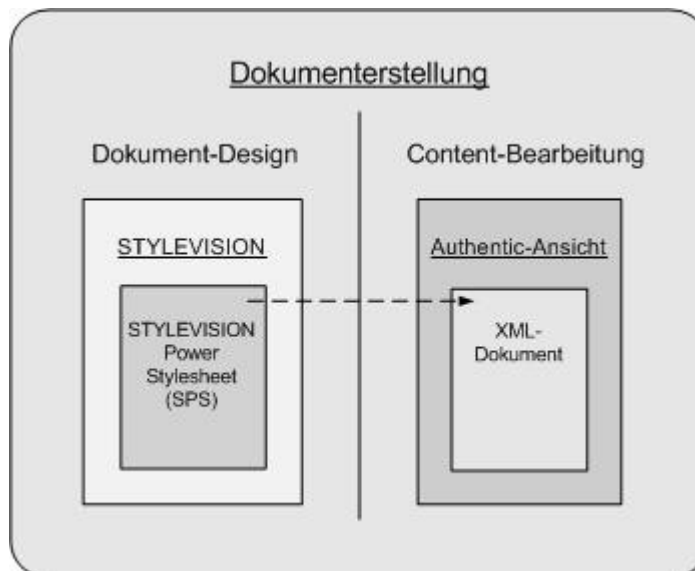
Wenn Sie ein SPS-Design öffnen, das für eine ältere IE-Version erstellt wurde und wenn die neuere IE Version 9 oder eine höhere Version auf Ihrem StyleVision-Rechner installiert ist, so erkennt StyleVision die neuere Version und fragt Sie in einem Dialogfeld, ob Sie die Kompatibilität in IE 9-Kompatibilität ändern möchten. Wenn Sie zur neuen Kompatibilität wechseln, stehen die oben aufgelisteten Optionen zur Verfügung. Die Darstellung des Dokuments in der Design-Ansicht, der Authentic-Ansicht und der HTML-Ausgabe bleibt mit Ausnahme von **Tabellenspalten**, die in IE 9 anders behandelt werden, unverändert. Wenn Sie die Kompatibilität in IE 9-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](#) wieder in die zuvor ausgewählte Version ändern.

7.5 SPS und Authentic-Ansicht

Einer der wichtigsten Verwendungszwecke des mit StyleVision erstellten SPS ist zu definieren, welche Daten im Ausgabedokument vorkommen sollen wie das XML-Dokument in der [Authentic-Ansicht](#), einer in Altova-Produkten verfügbaren Dokumentansicht, aussehen soll. In der Authentic-Ansicht können Benutzer, die mit XML nicht vertraut sind, problemlos Inhalt in ein XML-Dokument eingeben und darin editieren.

Die Erstellung und Bearbeitung von Dokumenten in der Authentic-Ansicht erfolgt in zwei separaten Schritten:

- **Dokumentdesign.** Die Authentic-Ansicht des XML-Dokuments ist eine grafische Ansicht und wird in StyleVision erstellt. Das Design-Dokument ist ein SPS. Das SPS steuert nicht nur die Verarbeitung des XML-Dokuments zur Anzeige in der Authentic-Ansicht und zur endgültigen Ausgabe sondern bietet in der Authentic-Ansicht auch Funktionen zur **Eingabe von Daten in die XML-Datei oder eine Datenbank.**



- **Bearbeitung von Inhalt.** Dieses SPS, das in der Design-Phase erstellt wurde, ist mit dem XML-Dokument verknüpft, das bearbeitet werden soll. (Das XML-Dokument muss entsprechend dem Schema, auf dem das SPS basiert, gültig sein.) Ein XML-Dokument, welches mit einem SPS verknüpft ist, wird in der [Authentic-Ansicht](#) eines Altova-Produkts als Authentic-Ansicht dieses XML-Dokuments grafisch dargestellt. Bei der Erstellung eines neuen Authentic-XML-Dokuments können Sie ihm ein SPS zuweisen und es anschließend unter Verwendung der im SPS definierten Dokumentvorlage ([XML-Vorlagendatei](#)) und der im SPS definierten Steuerelemente in der Authentic-Ansicht bearbeiten. Wenn ein bestehendes XML-Dokument geöffnet wird und ihm ein SPS zugewiesen wird, können die vorhandenen Daten entsprechend dem SPS-Design in der [Authentic-Ansicht](#) angezeigt werden und das Dokument kann in der [Authentic-Ansicht](#) bearbeitet werden.

Der Benutzer der Authentic-Ansicht benötigt keine XML-Kenntnisse oder Kenntnisse über das für das Dokument verwendete Schema. Die Anzeige des Dokuments in der Authentic-Ansicht sollte eine möglichst einfache Bearbeitung des Inhalts ermöglichen. Aufgabe desjenigen, der das SPS erstellt, ist es daher, eine benutzerfreundliche Anzeige in der Authentic-Ansicht zu entwerfen. Nähere Informationen zur Verwendung der Authentic-Ansicht finden Sie in der [Dokumentation zur Authentic-Ansicht](#) im Benutzerhandbuch von XMLSpy oder Authentic Desktop.

SPSs für Standard-Industrieschemas

Das [Authentic-Ansicht](#) Paket von Altova enthält SPSs für eine ganze Reihe von Standard-Industrieschemas. Der Benutzer kann daher sofort ein XML-Dokument in der Authentic-Ansicht erstellen, das auf einem Standardschema basiert. In der Abbildung unten sehen Sie den Teil einer Authentic-Ansicht des NCA Invoice-Standards.

NCA XML Invoice	
add documentID Invoice Type: (Proforma, Final, Debit Note, Credit Note) Status: (Draft, Final, Ammended)	
This is a template for the NCA Invoice used to define a detailed list of goods shipped or services rendered, with an account of costs. Information created could consist of parties assoicated with the transaction, type of goods shipped or method of shipment.	
Required	Optional
General Information	
References and other general information pertaining to the contract and this document.	
Date of Issue: (yyy-mm-dd) Document Creator Identifier: Docuement Number: Invoice Number:	Contract Id add contractExtension add documentVersion add contractType TransactionNumber add buyerContractIdentifier add sellerContractIdentifier add brokerContract

Die mitgelieferten Standard SPSs, die im Ordner **Examples/IndustryStandards** Ihrer Applikation abgelegt sind, können ganz einfach angepasst werden.

7.6 Synchronisieren von StyleVision und Authentic

Jede neue Version von StyleVision enthält Features, die die umfangreichen Funktionen der [Authentic-Ansicht](#) erweitern. Beachten Sie dabei jedoch bitte Folgendes:

- Es wird zu Kompatibilitätsproblemen kommen, wenn Sie ein mit einer neuen Version von StyleVision erstelltes SPS und eine ältere Version von Authentic verwenden.
- Neue mit einer neuen Version von StyleVision erstellte Funktionalitäten können nur von einer entsprechenden (oder höheren) Version von Authentic interpretiert werden.

Darum müssen alle online bereitgestellten [Authentic](#)-Produkte, die in Projekten eingesetzt werden, in dem ein mit einer neuen Version von StyleVision erstelltes SPS verwendet wird, synchronisiert werden. D.h. wenn z.B. zum Erstellen einer SPS-Datei **StyleVision 2008 release 2 Authentic Desktop 2008 release 2** (oder ein anderes Authentic View-Product aus dieser Release) verwendet werden muss, um diese SPS-Datei ordnungsgemäß bearbeiten zu können.

Natürlich kann auch jede neuere Version eines [Authentic-Produkts](#) SPSs interpretieren, die mit früheren Versionen von StyleVision erstellt wurden.

Synchronisierungsschritte

Wenn ein SPS bereits für mehrere Benutzer von Authentic bereitgestellt wurde und wenn später neue Authentic-Funktionalitäten zum SPS hinzugefügt werden, die mit einer neueren Version von StyleVision erstellt wurden, dann sollte der Entwickler bei der Synchronisierung folgendermaßen vorgehen:

1. Der Entwickler holt sich einen Lizenzschlüssel für seine neue Version von Authentic.
2. Der Entwickler testet, ob die Änderungen am SPS mit den neuen Versionen von StyleVision und Authentic funktionieren.
3. Die neue Version des [Authentic Produkts](#) wird an alle Benutzer von Authentic verteilt.
4. Erst wenn die obigen drei Schritte erfolgreich durchgeführt wurden, sollte das geänderte SPS den Authentic Benutzern zur Verfügung gestellt werden.

7.7 Generierte Dateien

In StyleVision können XSLT Stylesheets und Ausgabedateien über den Befehl [Datei | Generierte Dateien speichern](#) oder über das [StyleVisionBatch.exe](#) generiert werden. Wenn Sie XML nur validieren oder mittels XSLT transformieren wollen, können Sie dies direkt über die [Altova-Prozessoren](#) tun, ohne StyleVision aufrufen zu müssen. Die Altova-Prozessoren stehen auf der [Altova Website](#) in Form des kostenlosen Altova-Produkts AltovaXML zur Verfügung.

In StyleVision können die folgenden Dateien generiert werden:

- XSLT Stylesheets auf Basis des SPS Designs. Für die HTML- und RTF-Ausgabe werden separate XSLT Stylesheets generiert.
- Ausgabedateien, die durch Verarbeitung der im SPS zugewiesenen [XML-Arbeitsdatei](#) generiert werden, wobei anhand des SPS das XSLT Stylesheet generiert wird. Das [Befehlszeilentool](#) bietet die Option, als XML-Eingabedatei andere XML-Dateien als die [XML-Arbeitsdatei](#) zu definieren.

Die Markup-Tags für die Ausgabe sind im SPS enthalten. Die Daten für die Ausgabe stammen aus dem XML-Dokument oder der Datenbank. 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](#) 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.

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 und sind nahezu identisch mit der Authentic-Ansicht des XML-Dokuments. (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.
- **RTF-Ausgabe und Stylesheets:** (1) Im RTF-Design müssen Angaben für die Paginierung gemacht werden. Dies (z.B. Design der Umschlagseite, Links/Rechtspaginierung, usw.) erfolgt in der [Seitenleiste "Eigenschaften"](#) und in der [Seitenleiste "Design-Struktur"](#). (2) Wenn im SPS Dateneingabeelemente verwendet wurden, so werden diese, wenn möglich, in Form von Grafiken auf der RTF-Seite dargestellt. Wenn ein Dateneingabeelement nicht so leicht grafisch simuliert werden kann (z.B. Kontrollkästchen), wird eine Ersatzgrafik verwendet.

RTF-Ausgabe

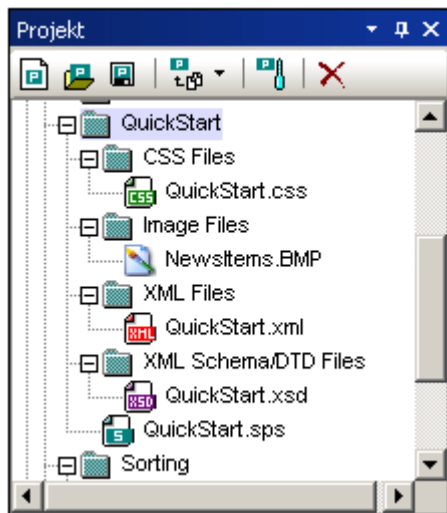
Die RTF-Ausgabe wird in einem einzigen Schritt anhand Ihrer XML-Datei generiert. Dabei wird das XML-Dokument mittels der für das SPS generierten XSLT-für-RTF-Datei verarbeitet. Die Eigenschaften der RTF-Ausgabe sind im SPS definiert. Die Ausgabe kann in der RTF-Vorschau überprüft werden. Um eine RTF-Datei zu erhalten, müssen Sie diese (über den Befehl [Datei | Generierte Dateien speichern](#) oder die [Befehlszeile](#)) generieren.

Anmerkung: Falls es ein Problem mit der eingebetteten Vorschau gibt, versucht StyleVision

das Vorschaudokument in einer externen Applikation (normalerweise in MS Word oder Adobe Reader) anzuzeigen. In StyleVision wird eine Fehlermeldung über die eingebettete Vorschau angezeigt. Wenn das Vorschaudokument in einer externen Applikation angezeigt wird, müssen Sie die externe Applikation schließen, bevor Sie das temporäre Ausgabedokument neu generieren, da sonst eine Fehlermeldung ausgegeben wird, die besagt, dass die Datei von einem anderem Programm verwendet wird. Sie sollten die externe Applikation außerdem vor dem Schließen des SPS-Designs schließen, da StyleVision die temporäre Ausgabedatei sonst nicht schließen kann, da sie von der externen Applikation gesperrt wurde.

7.8 Projekte in StyleVision

Miteinander in Zusammenhang stehende Dateien können in der Projekt-Seitenleiste (*Abbildung unten*) in einem Projekt zusammengefasst werden. Auf diese Art können Sie die Dateien in einem Projekt beim Erstellen eines SPS jederzeit aufrufen. So können Sie z.B. eine SPS-Datei aus der Projekt-Seitenleiste in die Design-Struktur-Seitenleiste ziehen und dort als Modul erstellen; oder Sie ziehen eine Bilddatei als statisches Bild in das Design; oder Sie ziehen ein CSS Stylesheet als externes Stylesheet in die Seitenleiste "Globale Stile".



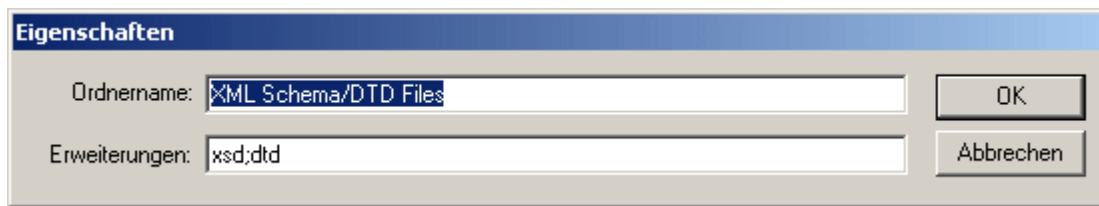
Erstellen und Speichern eines Projekts

Mit dem Befehl **Projekt | Projekt erstellen** können Sie ein neues Projekt erstellen. Ein Projekt enthält bei seiner Erstellung verschiedene Ordner für unterschiedliche Typen (*siehe Abbildung oben*). Dateitypen werden einem Ordner über das Dialogfeld "Eigenschaften" eines Ordners zugewiesen. Ein Projekt erhält einen Namen, wenn es zum ersten Mal (mit der Erweiterung `.svp`) gespeichert wird. Um den Namen eines Projekts später zu ändern, können Sie den Namen der Projektdatei in einer Anwendung wie z.B. dem Windows Datei-Explorer umbenennen.

Projektordner

Sie können sowohl zum Hauptprojektordner als auch zu Ordnern im Hauptprojektordner und Unterordnern auf allen Ebenen Ordner hinzufügen. Es können drei Arten von Ordnern hinzugefügt werden: (i) ein Projektordner; (ii) ein externer Ordner (welcher durch Suchen und Auswahl des Ordners hinzugefügt wird); (iii) ein externer Web-Ordner (welcher über eine URL hinzugefügt wird). Jede dieser drei Ordnerarten wird mit Hilfe eines der folgenden Projektbefehle hinzugefügt: (i) **Projektordner zu Projekt hinzufügen**; (ii) **Externes Verzeichnis zum Projekt hinzufügen**; (iii) **Externes Webverzeichnis zum Projekt hinzufügen**. Um eine dieser Ordnerarten zu einem Ordner oder Unterordner im Hauptprojekt hinzuzufügen, wählen Sie den entsprechenden Befehl im Kontextmenü für diesen Ordner bzw. Unterordner aus.

Jedem Ordner können im Dialogfeld "Eigenschaften" (*Abbildung unten*) eine oder mehrere Dateiarnten zugewiesen werden. Um das Dialogfeld aufzurufen, klicken Sie mit der rechten Maustaste auf den Ordner und wählen Sie im Kontextmenü den Befehl **Eigenschaften**.

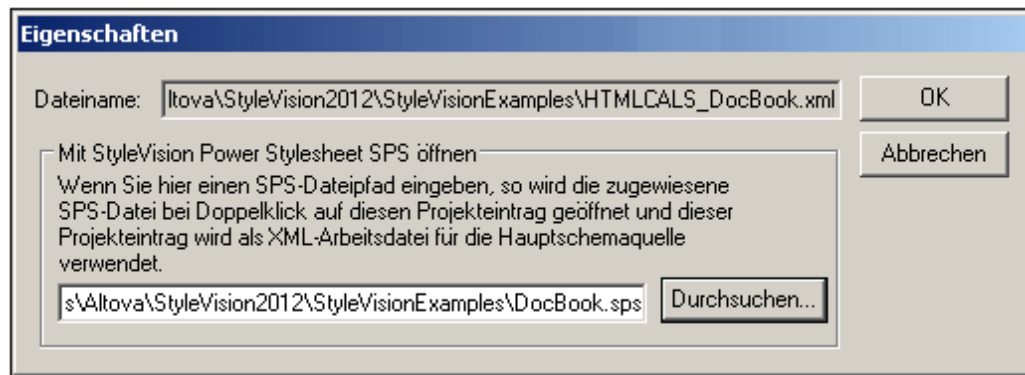


Im Menü "Eigenschaften" können Sie den Ordnernamen und die Dateierweiterungen für diesen Ordner bearbeiten (jede Dateierweiterung muss durch einen Strichpunkt getrennt sein). Auch die Namen von Projektordnern können bearbeitet werden. Wählen Sie dazu den Ordner in der Projekt-Seitenleiste aus, drücken Sie **F2** und bearbeiten Sie den Namen. Wenn für einen Ordner Dateierweiterungen definiert wurden, werden Dateien mit dieser Erweiterung zu diesem Ordner hinzugefügt, wenn Sie mit dem Befehl **Dateien zu Projekt hinzufügen** hinzugefügt werden. Wenn für mehrere Ordner dieselbe Dateierweiterung definiert wurde, wird die Datei zum ersten Ordner in der Projekt-Seitenleiste hinzugefügt, der diese Erweiterung hat. Die Ordner können in der Projekt-Seitenleiste mit Drag-and-Drop neu angeordnet werden. Auf der ersten Ebene (d.h. der Ebene unmittelbar unterhalb des Hauptprojektordners) werden die Ordner folgendermaßen angeordnet: (i) Projektordner; (ii) externe Ordner; (iii) externe Webverzeichnisse.

Projektdateien

Dateien können sowohl zum Hauptprojektordner als auch zu Ordnern und Unterordnern im Hauptprojektordner hinzugefügt werden. Dafür stehen im Projekt die folgenden Befehle zur Verfügung:

- **Dateien zu Projekt hinzufügen:** Im Browse-Fenster werden eine oder mehrere Dateien zum Hinzufügen ausgewählt. Die einzelnen hinzugefügten Dateien werden zum ersten Ordner hinzugefügt, dem diese Dateierweiterung zugewiesen wurde.
- **Globale Ressource zu Projekt hinzufügen:** Eine Datei wird über eine globale Ressource hinzugefügt.
- **URL zu Projekt hinzufügen:** Eine Datei wird über ihre URL (die im Dialogfeld "URL zu Projekt hinzufügen" definiert wird) hinzugefügt.
- **Aktive Datei zu Projekt hinzufügen:** Die aktive (SPS)-Datei wird zum ersten Ordner in der Projekt-Seitenleiste, für den als Dateierweiterung .sps definiert ist, hinzugefügt.
- **Aktive und verwandte Dateien zu Projekt hinzufügen:** Die aktive (SPS)-Datei sowie damit in Zusammenhang stehende Dateien wie z.B. Schema/s, XML-Arbeitsdateien, CSS-Dateien, statische Bilddateien usw., werden nach Dateierweiterung zu den entsprechenden Ordnern des Projekts hinzugefügt. Dies ist ein sehr nützlicher Befehl, um schnell alle für ein bestimmtes Design relevanten Dateien in einem Projekt zusammenzufassen.
- **Eigenschaften:** Wenn eine Projektdatei ausgewählt ist, wird bei Betätigen der Schaltfläche **Eigenschaften** in der Symbolleiste des Projektfensters bzw. bei Auswahl des Befehls "Eigenschaften" aus dem Kontextmenü ein Fenster angezeigt, in dem Sie den Pfad der Datei sehen. Bei XML-Dateien können Sie zusätzlich dazu eine SPS-Datei auswählen, mit der die Datei verknüpft werden kann (*siehe Abbildung unten*).



Wenn eine SPS-Datei mit einer XML-Datei, die sich in einem Projekt befindet, verknüpft ist, wird bei Doppelklick auf die XML-Datei im Projektfenster die SPS-Datei geöffnet, der die XML-Datei als XML-Arbeitsdatei des SPS zugewiesen wurde. Dies ist nützlich, wenn eine einzige SPS-Datei für mehrere XML-Dateien verwendet wird. Mit Hilfe dieser Funktion können Sie effizienter arbeiten, da Sie nicht mehr nach der SPS-Datei und/oder XML-Arbeitsdatei suchen müssen.

Mit den oben aufgelisteten Befehlen werden gemäß den Dateierweiterungen der Ordner im Hauptprojektordner Dateien zum Hauptprojekt hinzugefügt. Um Dateien zu spezifischen Ordnern oder Unterordnern hinzuzufügen, klicken Sie mit der rechten Maustaste auf den gewünschten Ordner und wählen Sie im Kontextmenü den entsprechenden Befehl. In einem Ordner sind die Dateien in alphabetischer Reihenfolge aufgelistet. Beachten Sie, dass Dateien auch in einen andere Ordner gezogen werden können. Um den Speicherpfad einer Datei zu sehen, klicken Sie im Kontextmenü der Datei auf **Eigenschaften**.

Globale Ressourcen

Eine [globale Ressource](#) vom Typ Datei oder Ordner kann zu einem Ordner hinzugefügt werden. Eine globale Ressource vom Typ Datei ist ein Alias für eine Dateiresource. Ein Alias kann mehrere Konfigurationen haben, wobei jede Konfiguration auf eine Dateiresource verweist. Wenn daher eine globale Ressource in einem Projekt verwendet wird, kann sie mit jeder beliebigen Zielressource verknüpft sein, je nachdem, welche Konfiguration in StyleVision gerade aktiv ist. Ähnlich dazu ist eine globale Ressource vom Typ Ordner ein Alias, dessen Ziel jeder beliebige von mehreren Ordnern sein kann, je nachdem, welche Konfiguration gerade aktiv ist. Wenn zum Definieren einer Datei (z.B. einer XML-Arbeitsdatei oder einer CSS-Datei) eine globale Ressource vom Typ Ordner verwendet wird, definiert die globale Ressource vom Typ Ordner nur einen Ordner; der Pfad von diesem Ordner zur benötigten Datei muss zusätzlich definiert werden. Nähere Informationen zur Verwendung von globalen Ressourcen finden Sie unter [Verwendung von globalen Ressourcen](#).

Drag-and-Drop

Sie können einen Ordner in der Projekt-Seitenleiste in einen anderen Ordner oder an eine andere Stelle innerhalb desselben Ordners ziehen. Eine Datei kann in einen anderen Ordner, nicht aber innerhalb desselben Ordners (in dem die Dateien alphabetisch geordnet sind) verschoben werden. Dateien und Ordner können auch aus dem Windows Datei-Explorer in die Projekt-Seitenleiste gezogen werden.

Verwendung von Projekten

Dateien in einem Projekt können je nach Dateityp auf verschiedene Arten verwendet werden. Für jede Datei in der Projekt-Seitenleiste werden die dafür zur Verfügung stehenden Aktionen in ihrem Kontextmenü (Aufruf durch Rechtsklick) aufgelistet. Wenn Sie eine Datei an eine Stelle ziehen, an der eine Aktion ausgeführt werden kann, wird ein Menü angezeigt, das den/die entsprechenden Befehl(e) enthält; bei einige Befehlen wird der Befehl sofort nach Ziehen der

Datei an die entsprechende Stelle ausgeführt. Im Folgenden finden Sie eine Liste der für die verschiedenen Dateitypen verfügbaren Befehle.

SPS-Dateien

- *Design öffnen* öffnet das SPS in einem neuen Design-Fenster. (Um diesen Befehl zu verwenden, können Sie die Datei auch in das Design-Fenster ziehen)
- *Als Modul importieren* importiert das SPS als Modul in das derzeit aktive SPS; die importierte Datei wird in der Design-Struktur unter der Überschrift "Module" aufgelistet. (Sie können die Datei auch durch Ziehen auf die Überschrift "Module" in der Seitenleiste "Design-Übersicht" importieren.)

XML-Dateien

- *Datei in XMLSpy bearbeiten* öffnet die XML-Datei in XMLSpy.
- *Neues Design erstellen* erstellt ein neues SPS. Das Schema für das SPS ist ein anhand des XML-Dokuments generiertes XML-Schema. Die XML-Arbeitsdatei des SPS ist die XML-Datei. (Um diesen Befehl zu verwenden, können Sie die Datei auch in die Hauptvorlagenleiste der Design-Ansicht ziehen.)
- *Als XML-Arbeitsdatei zuweisen* weist die XML-Datei als XML-Arbeitsdatei des SPS zu. (Um die Datei als XML-Arbeitsdatei hinzuzufügen, können Sie sie auch auf den Eintrag *Working XML* der Seitenleiste "Design-Übersicht" ziehen.)
- *Als XML-Vorlagendatei zuweisen* weist die XML-Datei als XML-Vorlagendatei des SPS zu. (Um die Datei als XML-Vorlagendatei hinzuzufügen, können Sie die Datei auch auf den Eintrag *Template XML* der Seitenleiste "Design-Übersicht" ziehen.)
- **Anmerkung:** Wenn Sie auf eine XML-Datei doppelklicken, wird, je nachdem, was auf dem Register "Projekt" des Dialogfelds [Optionen](#) festgelegt ist, eine von drei Aktionen durchgeführt: (i) Bearbeiten der Datei in XMLSpy; (ii) Erstellen eines neuen Designs auf Basis der XML-Datei; (iii) Der Benutzer wird gefragt, welche Aktion durchgeführt werden soll.

XML-Schema / DTD-Dateien

- *Datei in XMLSpy bearbeiten* öffnet die Schema-Datei in XMLSpy.
- *Neues Design erstellen* erstellt anhand des ausgewählten Schemas ein neues SPS. (Um ein neues SPS mit dem ausgewählten Schema als Schemaquelle zu erstellen, können Sie die Datei auch in die Hauptvorlagenleiste der Design-Ansicht ziehen).
- *Als Schema-Datei zuweisen* weist das ausgewählte Schema als Schemaquelle des aktiven SPS zu, wodurch die aktuelle Schemaquelle ersetzt wird. Dieser Befehl eignet sich besonders dazu, Schemas schnell zu ändern, z.B. wenn sich der Schemapfad geändert hat oder um eine falsche Zuweisung zu korrigieren. (Um diesen Befehl zu verwenden, können Sie die Datei auch auf den Eintrag *Schema* der Seitenleiste "Design-Übersicht" ziehen).
- **Anmerkung:** Wenn Sie auf eine XML-Datei doppelklicken, wird, je nachdem, was auf dem Register "Projekt" des Dialogfelds [Optionen](#) festgelegt ist, eine von drei Aktionen durchgeführt: (i) Bearbeiten der Datei in XMLSpy; (ii) Erstellen eines neuen Designs auf Basis der Schema-Datei; (iii) Der Benutzer wird gefragt, welche Aktion durchgeführt werden soll.

CSS-Dateien

- *Datei in XMLSpy bearbeiten* öffnet die CSS-Datei in XMLSpy.
- *In globale Stile importieren* fügt die CSS-Datei zu den externen CSS-Dateien der globalen Stile hinzu (Überschrift "Extern" in der Seitenleiste "Globale Stile"). (Um die Datei in die globalen Stile zu importieren, können Sie die Datei auch mit der Maus in die Seitenleiste "Globale Stile" auf die Überschrift "Extern" ziehen.)

HTML-Dateien

- *Datei in XMLSpy bearbeiten* öffnet die HTML-Datei in XMLSpy.
- *Öffnen* öffnet die HTML-Datei im Standard-Browser.
- *Neues Design erstellen* erstellt ein neues SPS, in dem Sie das Schema auf Basis des HTML-Dokuments erstellen können. (Um diesen Befehl zu verwenden, können Sie die Datei auch mit der Maus in der Design-Ansicht ziehen.)

Bilddateien

- *Öffnen* öffnet die Bilddatei im Standard-Bildanzeigeprogramm / in der Standard-Bildbearbeitungsapplikation.
- *Bild in Design einfügen* fügt das Bild als statisches Bild in das SPS ein. (Sie können das Bild auch durch Ziehen mit der Maus an einer bestimmten Stelle einfügen.)

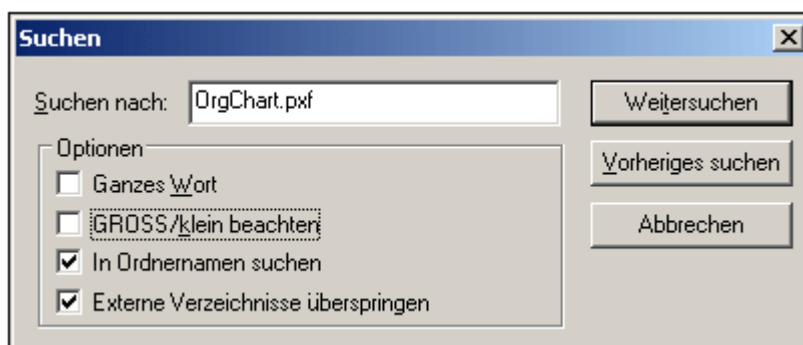
Alle Dateitypen

- *Übergeordneten Ordner anzeigen* öffnet ein Windows-Datei-Explorer-Fenster, in dem Sie den Inhalt des Ordners, in dem sich die ausgewählte Datei befindet, sehen.
- Die Befehle *Ausschneiden*, *Kopieren*, *Einfügen*, *Löschen* funktionieren wie die Standard-Windows-Befehle und dienen zum Ausschneiden und Kopieren der ausgewählten Datei in die Zwischenablage; zum Einfügen der Dateien aus der Zwischenablage; und zum Löschen. Diese Befehle können auch auf eine Auswahl mehrerer Dateien angewendet werden..
- *Alles markieren* wählt alle Dateien im Projekt aus.
- *Eigenschaften* ruft das Dialogfeld "Eigenschaften" auf, in dem der Speicherpfad der Datei angegeben ist.

In Projekt suchen

Sie können Projektdaten und -ordner nach ihrem Namen bzw. einem Teil ihres Namens suchen. Wurden Treffer gefunden, so werden die gefundenen Dateien bzw. Ordner der Reihe nach markiert.

Um eine Suche zu starten, wählen Sie den zu durchsuchenden Projektordner in der Seitenleiste "Projekt" aus und klicken Sie auf den Befehl **Bearbeiten | Suchen** (oder das Tastenkürzel **Strg+F**). Geben Sie nun im Dialogfeld "Suchen" (*Abbildung unten*) den gewünschten Such-String ein und aktivieren bzw. deaktivieren Sie die Suchoptionen nach Bedarf (*Erklärung weiter unten*)



Es stehen die folgenden Suchoptionen zur Verfügung:

- **Ganzes Wort:** Hierbei ist die Suche stärker eingeschränkt, da der Suchstring mit einem ganzen Wort im Datei- bzw. Ordnernamen übereinstimmen muss. Die Teile vor und nach dem Punkt in den Dateinamen werden (ohne Berücksichtigung des Punkts) jeweils als Wort behandelt.

- Sie können festlegen, dass die Groß- und Kleinschreibung im Suchstring genau mit dem Textstring im Datei- oder Ordnernamen übereinstimmen muss.
- Ordnernamen können in die Suche einbezogen werden. Andernfalls werden nur Dateinamen durchsucht.
- [Externe Ordner](#) können in die Suche einbezogen oder davon ausgeschlossen werden. Bei externen Ordnern handelt es sich um tatsächliche Ordner auf Ihrem Rechner bzw. Netzwerk und nicht um Projektordner, die innerhalb des Projekts und nicht auf dem Rechner angelegt wurden.

Wurden Treffer gefunden, wird der erste Treffer in der Seitenleiste "Projekt" markiert. Sie können anschließend im Dialogfeld **Suchen** mit Hilfe von **Weitersuchen** und **Vorh. suchen** der Reihe nach zu allen gefundenen Suchtreffern springen.

7.9 Kataloge in StyleVision

StyleVision unterstützt einen Teil der OASIS XML-Kataloge. Mit Hilfe von Katalogen kann StyleVision gebräuchliche Schemas (sowie Stylesheets und andere Dateien) aus lokalen Benutzerordnern abrufen, wodurch die Verarbeitungsgeschwindigkeit erhöht wird, Benutzer auch offline arbeiten können (d.h. ohne mit einem Netzwerk verbunden zu sein) und Dokumente leichter übertragbar werden (da die URIs nur in den Katalogdateien geändert werden müssen).

Der Mechanismus in StyleVision funktioniert, wie unten beschrieben.

RootCatalog.xml

StyleVision lädt eine Datei namens `RootCatalog.xml` (*Struktur siehe unten*), die eine Liste von Katalogdateien enthält, die durchsucht werden. Sie können jeweils im Element `nextCatalog` in `RootCatalog.xml` beliebig viele Katalogdateien definieren, die durchsucht werden sollen.

```
<?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"/>
  <nextCatalog catalog="CoreCatalog.xml"/>
  <!-- Include all catalogs under common schemas folder on the first directory
level -->
  <nextCatalog spy:recurseFrom="%AltovaCommonFolder%/Schemas" catalog="
catalog.xml" spy:depth="1"/>
  <!-- Include all catalogs under common XBRL folder on the first directory
level -->
  <nextCatalog spy:recurseFrom="%AltovaCommonFolder%/XBRL" catalog="
catalog.xml" spy:depth="1"/>
</catalog>
```

Die in `RootCatalog.xml` inkludierten Katalogdateien werden durchsucht und die URIs werden entsprechend den Zuordnungen in den Katalogdateien aufgelöst. Beachten Sie, dass die Katalogdateien `catalog.xml` in den Ordnern `Schemas` und `XBRL` des durch die Variable `%AltovaCommonFolder%` definierten Ordners durchsucht werden. (Der Wert der `%AltovaCommonFolder%` Variablen wird in der Tabelle unten aufgelistet.) Die Katalogdateien im `Altova-Ordner "Common"` mappen die vordefinierten Public und System Identifiers häufig verwendeter Schemas (wie z.B. SVG und WSDL) und XBRL-Taxonomien auf URIs, die auf lokal gespeicherte Kopien der entsprechenden Schemas verweisen. Diese Schemas werden bei der Installation von StyleVision im `Altova Ordner "Common"` installiert. Achten Sie darauf, keine Zuordnungen doppelt zu vergeben, da dies Fehler verursachen könnte.

CoreCatalog.xml, CustomCatalog.xml und Catalog.xml

Beachten Sie in der `RootCatalog.xml` Liste oben, dass `CoreCatalog.xml` und `CustomCatalog.xml` als `Lookup` aufgelistet sind:

- `CoreCatalog.xml` enthält bestimmte `Altova-spezifische` Zuweisungen zum Auffinden von Schemas im `Altova Ordner "Common"`.
- `CustomCatalog.xml` ist eine `Skelettdatei`, in der Sie ihre eigenen Zuweisungen erstellen können. Sie können für jedes gewünschte Schema Zuweisungen zu `CustomCatalog.xml` hinzufügen, doch dieses wird nicht von den Katalogdateien im `Altova Common-Ordner` angegeben. Sie können dies mit Hilfe der unterstützten Elemente des OASIS-Katalogs (*siehe unten*) tun (*siehe unten*).
- Der `Altova-Ordner "Common"` enthält eine Reihe von `Catalog.xml` Dateien. Jede

befindet sich in einem Ordner eines bestimmten Schemas oder einer XBRL-Taxonomie im Altova-Ordner "common" und jede weist URIs Public und/oder System Identifier zu, die auf lokal gespeicherte Kopien der entsprechenden Schemas verweisen.

Speicherpfad von Katalogdateien und Schemas

Die Dateien `RootCatalog.xml` und `CoreCatalog.xml` sind im StyleVision Applikationsordner installiert. Die Datei `CustomCatalog.xml` befindet sich im Ordner `MyDocuments/Altova/StyleVision`. Die `Catalog.xml` Dateien befinden sich in den jeweiligen Schemaordnern. Diese Schemaordner befinden sich im Ordner: `%AltovaCommonFolder%\Schemas` and `%AltovaCommonFolder%\XBRL`.

Shell-Umgebungsvariablen und Altova-Variablen

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:

<code>%AltovaCommonFolder%</code>	C:\Program Files\Altova\Common2012
<code>%DesktopFolder%</code>	Vollständiger Pfad zum Desktop-Ordner für den aktuellen Benutzer.
<code>%ProgramMenuFolder%</code>	Vollständiger Pfad zum Programmenü-Ordner für den aktuellen Benutzer.
<code>%StartMenuFolder%</code>	Vollständiger Pfad zum Startmenü-Ordner für den aktuellen Benutzer.
<code>%StartUpFolder%</code>	Vollständiger Pfad zum Startordner für den aktuellen Benutzer.
<code>%TemplateFolder%</code>	Vollständiger Pfad zum Vorlagen-Ordner für den aktuellen Benutzer.
<code>%AdminToolsFolder%</code>	Vollständiger Pfad zum Dateisystemverzeichnis, in dem die Verwaltungstools für den aktuellen Benutzer gespeichert sind.
<code>%AppDataFolder%</code>	Vollständiger Pfad zum Applikationsdatenordner für den aktuellen Benutzer.
<code>%CommonAppDataFolder%</code>	Vollständiger Pfad zum Dateiverzeichnis mit den Applikationsdaten für alle Benutzer.
<code>%FavoritesFolder%</code>	Vollständiger Pfad zum Ordner "Favoriten" für den aktuellen Benutzer.
<code>%PersonalFolder%</code>	Vollständiger Pfad zum persönlichen Ordner für den aktuellen Benutzer.
<code>%SendToFolder%</code>	Vollständiger Pfad zum Ordner "SendTo" für den aktuellen Benutzer.
<code>%FontsFolder%</code>	Vollständiger Pfad zum Systemschriftarten-Ordner.
<code>%ProgramFilesFolder%</code>	Vollständiger Pfad zum Programmdateiordner für den aktuellen Benutzer.
<code>%CommonFilesFolder%</code>	Full path to the Common Files folder for the current user. Vollständiger Pfad zum Ordner "Gemeinsame Dateien" für den aktuellen Benutzer.
<code>%WindowsFolder%</code>	Vollständiger Pfad zum Windows-Ordner für den aktuellen Benutzer.
<code>%SystemFolder%</code>	Vollständiger Pfad zum Systemordner für den aktuellen Benutzer.
<code>%CommonAppDataFolder%</code>	Vollständiger Pfad zum Dateiverzeichnis mit den Applikationsdaten für alle Benutzer.

`%LocalAppDataF` Vollständiger Pfad zum Dateisystemverzeichnis, das als Datenspeicher für
`older%` lokale (nicht-Roaming) Applikationen dient.

`%MyPicturesFol`
`der%` Vollständiger Pfad zum Ordner "Meine Bilder".

Funktionsweise von Katalogen

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

Für gebräuchliche Schemas ist der PUBLIC Identifier normalerweise vordefiniert, sodass nur die URI in der Katalogdatei in die Adresse der richtigen lokalen Kopie geändert werden muss. Wenn das XML-Dokument geparkt 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"/>
    <rect x="5" y="5" width="15" height="15"/>
  </g>
</svg>
```

Dieses Dokument wird gelesen und der Katalog wird nach dem PUBLIC Identifier 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 auf die URI `schemas/svg/svg11.dtd` umgeleitet wird; (dieser Pfad ist relativ zur Katalogdatei). Diese lokale Datei wird dann als DTD verwendet. Wenn im Katalog kein passender Treffer gefunden wird, wird die URL im XML-Dokument verwendet (im Beispiel oben: `http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd`).

Die von StyleVision unterstützte Katalogteilmenge

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](#). Bitte beachten Sie: Jedes Element kann das Attribut `xml:base` erhalten, welches dazu dient, die Basis-URI dieses Elements zu definieren.

- `<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"/>`

In Fällen, in denen es wie z.B. bei den meisten Stylesheets keinen Public Identifier gibt, kann der System Identifier einer URL direkt über das `system` Element zugewiesen werden. Außerdem kann eine URI auch über das `uri` Element einer anderen URI zugewiesen werden. Mit Hilfe der Elemente `rewriteURI` und `rewritsSystem` kann der Anfangsteil einer URI bzw. eines System Identifier neu geschrieben werden. Auf diese Art kann der Beginn eines Dateipfads ersetzt werden, wodurch ein anderes Verzeichnis als Ziel definiert werden kann. Nähere Informationen zu diesen Elementen finden Sie in der [XML Catalogs Specification](#).

Weitere Informationen

Nähere Informationen zu Katalogen finden Sie in der [XML-Katalogspezifikation](#).

Kapitel 8

SPS-Dateien: Inhalt

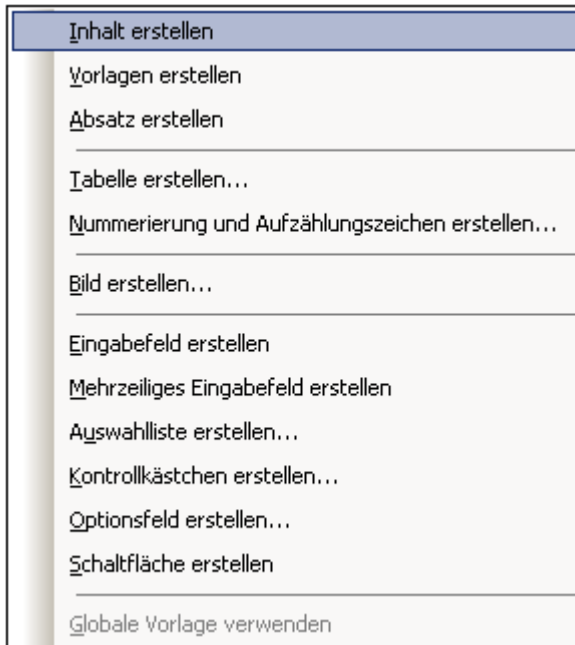
8 SPS-Dateien: Inhalt

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](#), [globale Vorlage](#) oder [Design-Fragment](#).

- [Einfügen von XML-Inhalt als Text](#). 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.
- [Arbeiten mit Tabellen](#). 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](#). 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](#): 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
- Diagramme
- [Verwendung von Dateneingabe- \(oder Formularsteuer\) -Elementen](#). 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](#)
- [Barcodes](#)
- [Layout-Module](#)
- [Die Funktion "Ändern in"](#). Mit Hilfe dieser Funktion können Sie einen anderen Node für eine Vorlage auswählen. Außerdem kann ein Node in einen anderen Content Type umgewandelt werden.

8.1 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*).



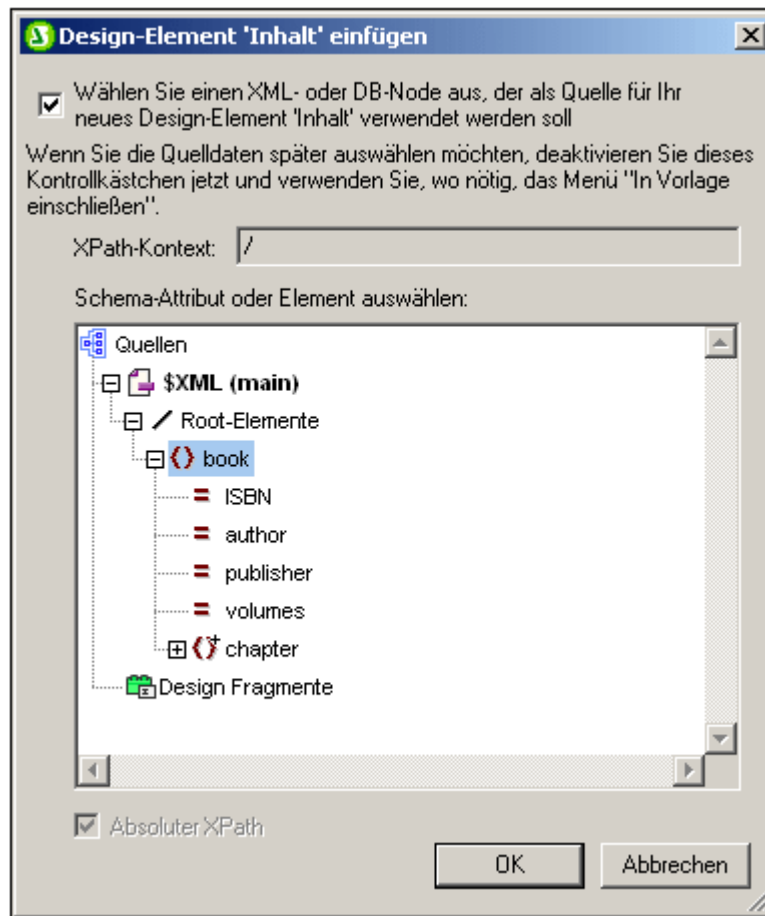
Typen von Schema-Nodes

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 Symbolleiste "Inhalt einfügen"

Auch über die Schaltfläche **Inhalt einfügen** der [Symbolleiste "Design-Elemente einfügen"](#) 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.:

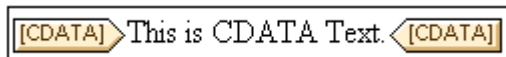


In der Abbildung oben wurde das Element *Desc* als Inhalt erstellt. In der Ausgabe wird der Textinhalt von *Desc* angezeigt. Wenn *Desc* untergeordnete Elemente hat wie z.B. *Bold* und *Italic*, dann wird der Textinhalt der untergeordneten Elemente ebenfalls als Teil des Inhalts von *Desc* ausgegeben. 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.

CDATA-Abschnitte

Wenn im XML-Dokument CDATA-Abschnitte vorhanden sind, werden sie ausgegeben und in der

Authentic-Ansicht werden sie durch Tags gekennzeichnet, wenn die Anzeige von Markup aktiviert ist (Menübefehl [Authentic | Markup](#)). CDATA-Abschnitte können auch beim Bearbeiten des Dokuments in der Authentic-Ansicht (über das Kontextmenü) in das XML-Dokument eingefügt werden.



Anmerkung: CDATA-Abschnitte können in der Authentic-Ansicht 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.

In den Unterabschnitten dieses Abschnitts beschreiben wir andere Aspekte des Einfügens von XML-Inhalt als Text:

- Wie der Textinhalt eines Node [direkt mit Markup-Tags für ein vordefiniertes Format versehen werden kann](#) wenn der Node eingefügt wird.
- Wie die Struktur des Quellschemas vorgibt, [wie sich die Verwendung der Authentic-Ansicht auswirkt](#).
- Wie untergeordnete Nodes, die nicht explizit in einem Node inkludiert sind, für die Verarbeitung inkludiert werden können. Siehe [Restliche Inhalt](#).

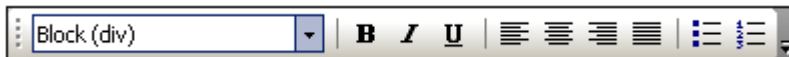
Anmerkung: Sie können eine **leere Vorlagenregel** erstellen, indem Sie den (`Inhalt`) Platzhalter eines Node löschen. Eine leere Vorlagenregel eignet sich dazu, zu definieren, dass ein bestimmter Node nicht verarbeitet werden soll, d.h. keine Ausgabe erzeugen soll.

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 (*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](#) 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.

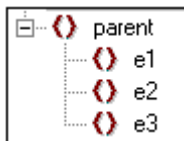
Hinzufügen von Elementen in der Authentic-Ansicht

Beim Erstellen von Elementen im Design bestimmt die Art, wie Sie Elemente erstellen, wie die Authentic-Ansicht reagiert, wenn der Benutzer z.B. die Tabulatortaste drückt oder auf die Aufforderung `Hinzufügen...` klickt. Grundsätzlich geht es darum, welche Elemente in der Authentic-Ansicht erstellt werden, wenn ein Element vom Benutzer hinzugefügt wird. Wenn der Benutzer z.B. ein Element hinzufügt (z.B. durch Klicken auf das Symbol **Element hinzufügen** in der Seitenleiste "Elemente"), welche Child-Elemente werden automatisch erstellt?

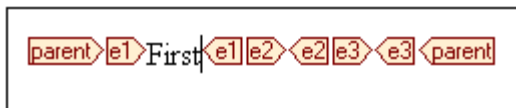
Zu beachten ist, dass die Authentic-Ansicht der Struktur des zugrundeliegenden Schemas folgt. Um sicherzustellen, dass die Authentic-Ansicht die Schemastruktur korrekt implementiert, sollten Sie sich an die unten angeführten Design-Regeln halten.

Eindeutiges Content Model

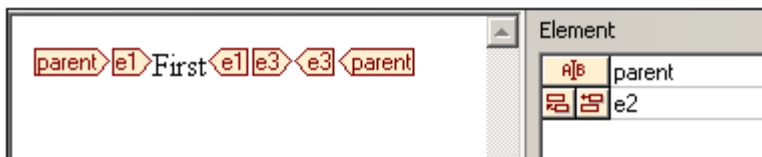
Ein Content Model wird dann als eindeutig betrachtet, wenn es aus einer einzigen Sequenz (mit `maxOccurs=1`) von Child-Elementen besteht (ohne Choices, Groups, Substitutions usw.). In diesem Fall ist die Reihenfolge der Child-Elemente beim Hinzufügen eines Elements eindeutig und sie werden automatisch hinzugefügt. Im Beispiel in der Abbildung unten sind alle drei Child-Elemente obligatorisch und dürfen nur einmal vorkommen.



Wenn das Element `parent` in der Authentic-Ansicht hinzugefügt wird, werden seine Child-Elemente automatisch eingefügt (*Abbildung unten*). Wenn Sie die Tabulatortaste drücken, gelangen Sie zum nächsten Element in der Sequenz.



Wäre das Element `e2` optional, würden bei Hinzufügen des Elements `parent` in der Authentic-Ansicht die Elemente `e1` und `e3` automatisch eingefügt und das Element `e2` würde in der Seitenleiste "Elemente" angezeigt werden, damit es bei Bedarf eingefügt werden kann (*Abbildung unten*). Durch Drücken der Tabulatortaste in `e1` gelangt der Benutzer zu `e3`.



Das obige Content Model-Szenario ist das einzige Szenario, das in der Authentic-Ansicht als eindeutig gilt. Alle anderen Fälle gelten als nicht eindeutig. Damit das Szenario in der Authentic-Ansicht eindeutig wird und die gewünschten Elemente angezeigt werden, müssen Sie sich im Design an die folgenden einfachen Regeln halten.

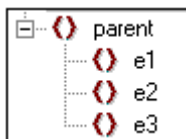
Nicht eindeutiges Content Model

Damit die Elemente während der Bearbeitung eines XML-Dokuments in der Authentic-Ansicht richtig und effizient angezeigt werden können, gelten für das SPS die folgenden Regeln.

- Child-Elemente werden in der Reihenfolge angezeigt, in der sie im Design angelegt wurden.
- Damit es in der Authentic-Ansicht keine Verwechslungen zwischen Child-Elementen gibt, sollten alle Child-Elemente im Design-Dokument in der erforderlichen Reihenfolge angelegt sein und sich **innerhalb eines einzigen Parent Node befinden**. Wenn in der Authentic-Ansicht eine gleichrangige Beziehung beibehalten werden soll, sollten die einzelnen Child-Elemente eines einzigen Parent-Elements nach Möglichkeit nicht in mehrere Instanzen des Parent-Node platziert werden.

Diese beiden Regeln werden im folgenden Beispiel veranschaulicht.

Angenommen wir haben das Content Model eines Elements `parent`, welches aus einer einzigen Sequenz obligatorischer Child-Elemente besteht. Dieses Content Model ist dem oben beschriebenen eindeutigen Content Model ähnlich. Es gibt jedoch einen Unterschied: Die einzige Sequenz ist optional. Dadurch ist das Content Model nicht mehr eindeutig, da nicht mehr sicher ist, ob die Sequenz vorhanden ist. Wenn Sie ein Design-Dokument wie in der Abbildung unten erstellen, gibt es in der Authentic-Ansicht ein Problem aufgrund mangelnder Eindeutigkeit.



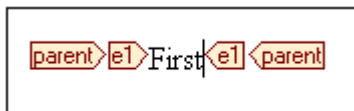
Die Authentic-Ansicht des `parent` Elements sieht folgendermaßen aus (da die Sequenz optional ist):



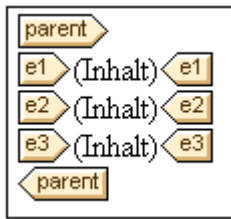
Wenn Sie auf `Hinzufügen...` klicken, erscheint ein Menü mit den drei Child-Elementen:



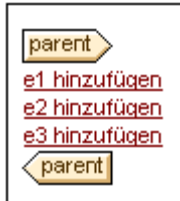
Wenn Sie eines dieser Elemente auswählen, wird es eingefügt (*Abbildung unten*). Da die Sequenz in der Authentic-Ansicht jedoch nicht eindeutig ist, wird keines der restlichen beiden Elemente eingefügt und Sie haben auch nicht die Möglichkeit sie einzufügen:



Richtig müssten die erforderlichen Nodes in diesem Content Model (entsprechend der obigen Regeln) in der gewünschten Reihenfolgen innerhalb des einzelnen Parent Node erstellt werden. Das Design-Dokument würde folgendermaßen aussehen:



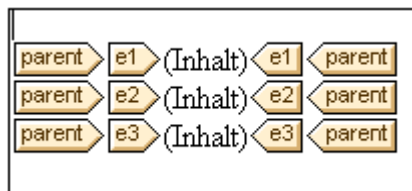
Beachten Sie, dass sich alle drei Child-Elemente **in einem einzigen Parent Node befinden**. Das oben gezeigte Design würde die folgende Authentic-Ansicht erzeugen:



Der Benutzer der Authentic-Ansicht klickt auf den entsprechenden Eintrag zum Hinzufügen eines Elements, um das Element und seinen Inhalt einzufügen.

Anmerkung:

- Wenn ein Element mehrmals vorkommen kann und wenn Sie sich an die oben angeführten Regeln halten, so wird das Element solange in der Seitenleiste angezeigt, bis es in der Authentic-Ansicht so oft vorkommt, wie dies durch die Maximalanzahl im Schema (`maxOccurs`) zugelassen ist.
- Wenn jedes Child-Element in einem separaten Parent-Node erstellt wird (*siehe Abbildung unten*), werden nicht nur separate Child-Element-Beziehungen für die einzelnen auf diese Art instantiierten Child-Elemente erstellt, sondern es wird auch die Verarbeitungsdauer erhöht, da für jedes Child-Element auch der Parent Node wieder erneut aufgerufen werden muss.



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](#) definiert sind, werden die globalen Vorlagen verwendet. Andernfalls werden die vordefinierten Standardvorlagen (für Element-, Attribut- und Text-Nodes) angewendet.

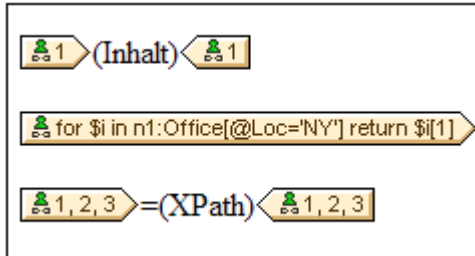
Wichtig: Beachten Sie bitte, welche Nodes für `restlicher Inhalt` ausgewählt werden.

- 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.

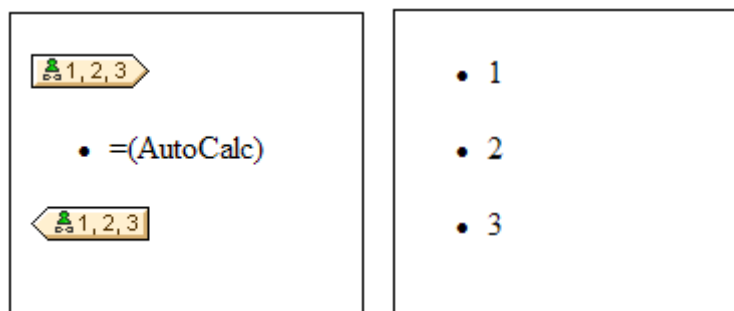
8.2 Benutzerdefinierte Vorlagen

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 auf der linken Seite der Tags. Benutzerdefinierte Vorlagen sind sehr nützlich, da sie beim Erstellen von Vorlagen besonders viel Flexibilität bieten. Beachten Sie, dass Inhalt, der durch benutzerdefinierte Vorlagen generiert wurde, in der **Authentic-Ansicht nicht bearbeitet werden kann**.



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, sodass diese in der Authentic-Ansicht bearbeitet werden kann. 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 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 `xml: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 eingeebene

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 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 weiter oben.

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	<pre><xsl:for-each select="Org"> <xsl:for-each select="Office"> <xsl:for-each select="Dept"> ... </xsl:for-each> </xsl:for-each> </xsl:for-each></pre>	Jedes <code>Office</code> Element hat seine eigene <code>Dept</code> Population. Dadurch kann die Gruppierung und Sortierung innerhalb der einzelnen <code>Office</code> Elemente durchgeführt werden.
Ja	<pre><xsl:for-each select="/Org/Office/Dept"> ... </xsl:for-each></pre>	Die <code>Dept</code> Population umfasst alle <code>Office</code> Elements und alle <code>Org</code> 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](#), 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

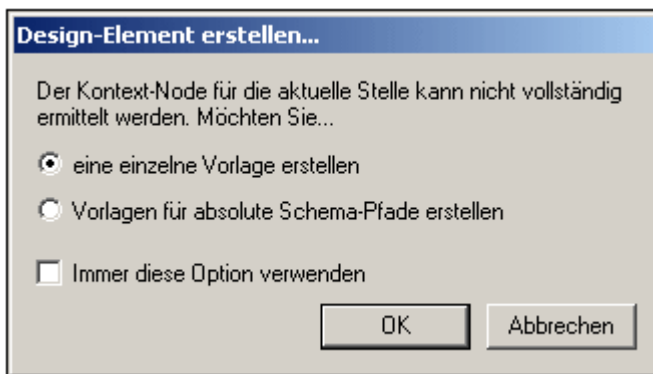
verbundenen und getrennten Nodes) dasselbe, es sei denn ein Gruppierungs- oder Sortierkriterium wurde definiert. Nähere Informationen darüber, wie sich verbundene Nodes auf die Gruppierung und Sortierung auswirken, finden Sie unter [Node-Vorlagen-Operationen](#).

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 Dialogfeld [Extras | Optionen](#) ändern.

8.3 Benutzerdefinierte Elemente, XML-Textblöcke

Mit Hilfe von [benutzerdefinierten Elementen](#) und [benutzerdefinierten XML-Textblöcken](#) 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.

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.

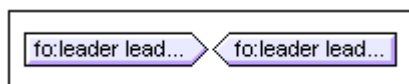
Anmerkung: Benutzerdefinierte Elemente werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

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 (*siehe Abbildung unten*).

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.

Benutzerdefinierte XML-Textblöcke

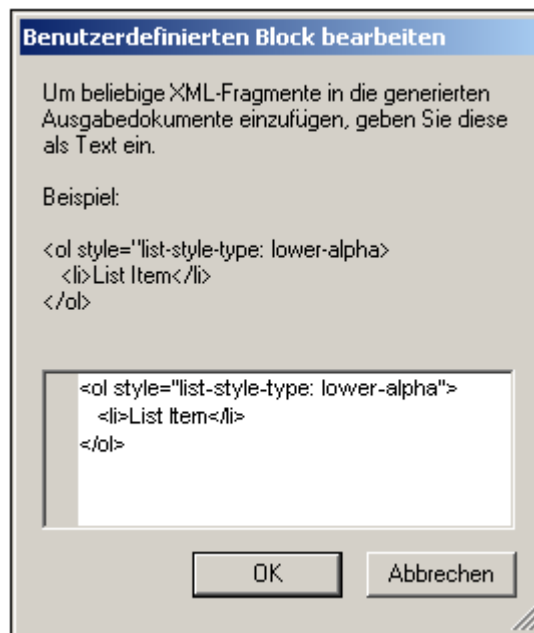
Ein benutzerdefinierter XML-Textblock ist ein XML-Fragment, das in den vom SPS generierten XSLT-Code eingefügt wird. Er wird als eigenständiger Block, zu dem kein Design-Element hinzugefügt werden kann, in das SPS Design eingefügt. Ein solcher XML-Textblock sollte daher an der Stelle im Stylesheet, an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

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.

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.

8.4 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 dieser 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 der Authentic-Ansicht und 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>
    <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>
      <name>Name</name>
      <phone>Phone</phone>
    </person>
    <person>
      <name>John Merrimack</name>
      <phone>6517890</phone>
    </person>
    <person>
      <name>Joe Concord</name>
      <phone>6402387</phone>
    </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:

- 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

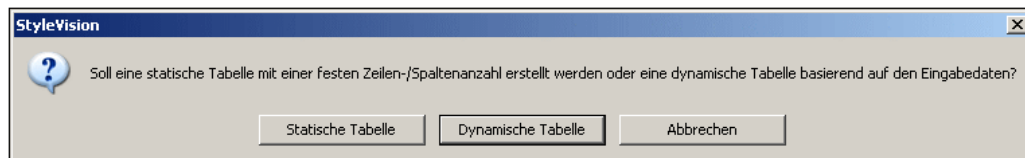
Für denjenigen, der das Dokument entwirft, sind die folgenden Punkte zu beachten:

- **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.

Statische Tabellen

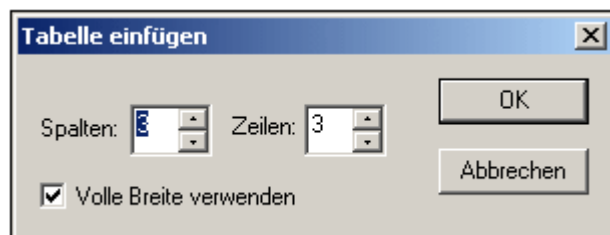
Um eine statische Tabelle einzufügen, gehen Sie folgendermaßen vor:

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.

Person	Telephone		Fax	
	Office	Home	Office	Home

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. Sie können **nicht für CALS/HTML-Tabellen** verwendet werden, da [CALS/HTML-Tabellen](#) nicht auf diese Art formatiert werden.

Dynamische Tabellen

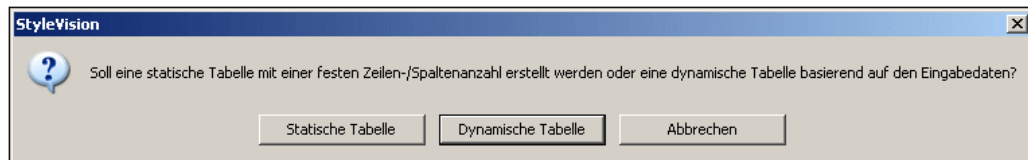
Um eine dynamische Tabelle einzufügen, gehen Sie folgendermaßen vor:

1. Verwenden Sie einen der folgenden Befehle: **Tabelle | Tabelle einfügen** oder **Einfügen | Tabelle** oder klicken Sie in der Symbolleiste "Design-Element einfügen" auf



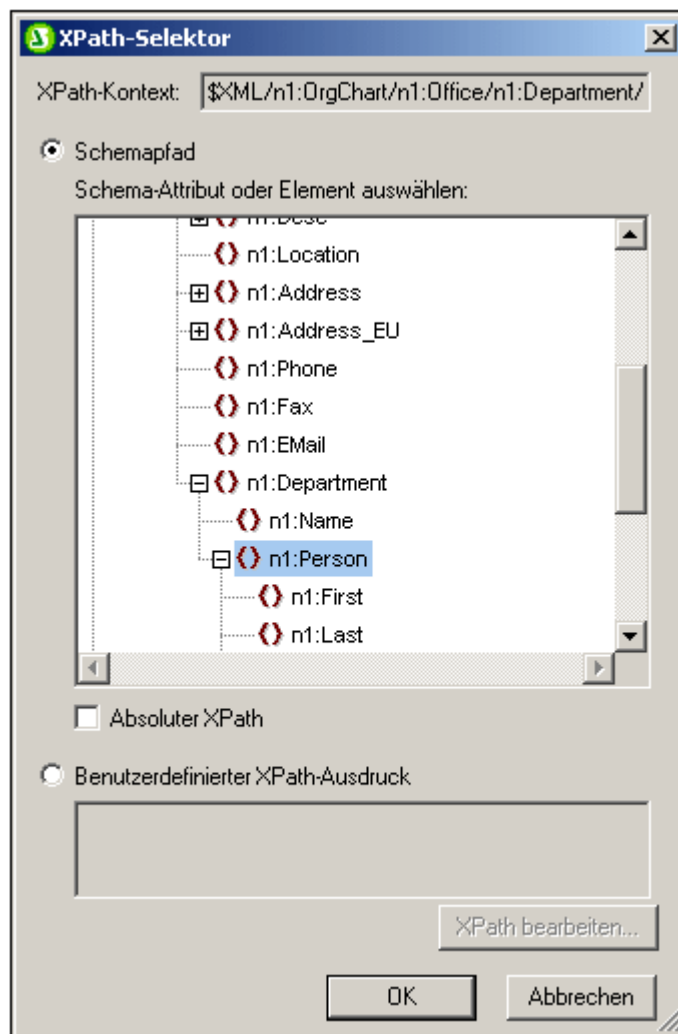
die Schaltfläche **Tabelle einfügen**.

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.



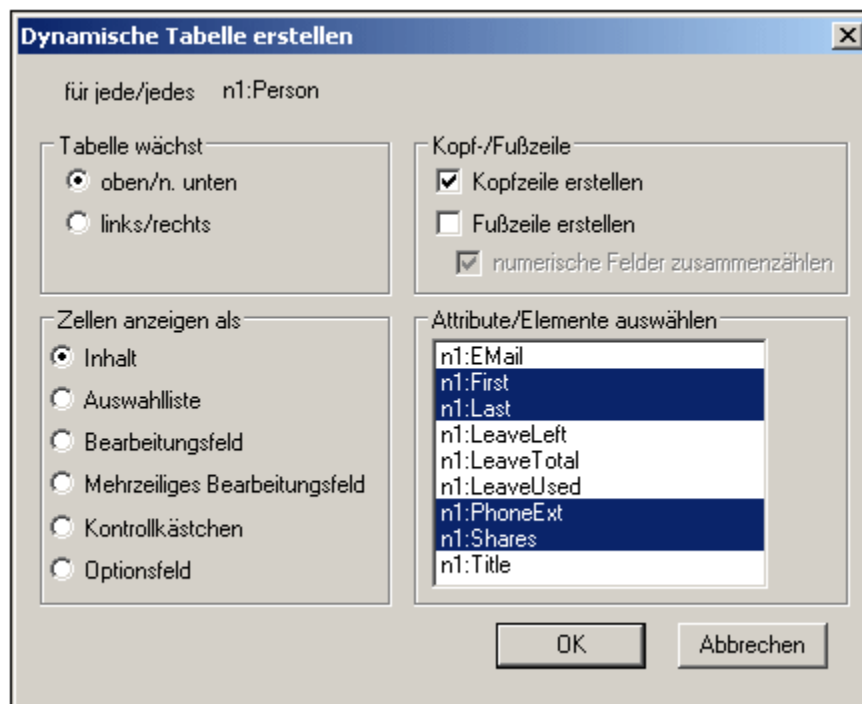
Klicken Sie auf **Dynamische Tabelle**

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 `nl:Department` und der Node `nl: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.



5. Die Child-Elemente und -Attribute des Elements, das in des Design-Fenster gezogen wurde, werden in der Liste "Attribute/Elemente 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:

name	street	city	state	zip
ipo:name (contents) ipo:name	ipo:street (contents) ipo:street	ipo:city (contents) ipo:city	ipo:state (contents) ipo:state	ipo:zip (contents) ipo:zip

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

name	 (contents) 
street	 (contents) 
city	 (contents) 
state	 (contents) 
zip	 (contents) 

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 eine 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.

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 Dateneingabelemente 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. Sie können **nicht für CALS/HTML-Tabellen** verwendet werden, da [CALS/HTML-Tabellen](#) nicht auf diese Art formatiert werden.

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**).

	Header-C1		Header-C2
	O n1:Name (Inhalt) O n1:Name		O n1:Person (Inhalt) O n1:Person
	Footer-C1		Footer-C2

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*).

		O n1:Person	
	Header-R1	O n1:First (Inhalt) O n1:First	Footer-R1
	Header-R2	O n1:Last (Inhalt) O n1:Last	Footer-R2

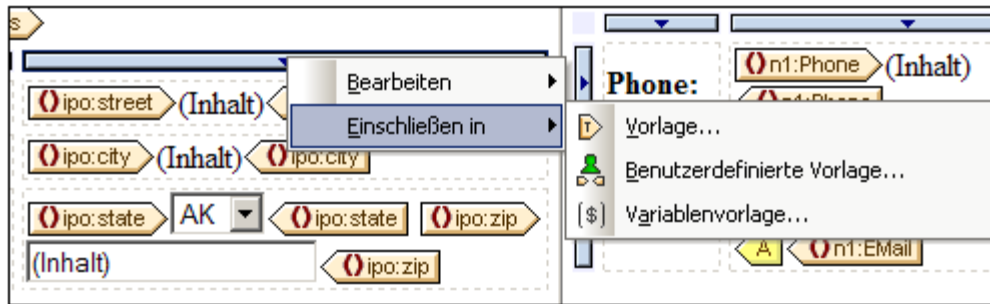
Nach Auswahl einer Spalte, Zeile oder Tabelle können in den Fenstern "Stile" und "Eigenschaften" Stile und/oder Eigenschaften für die Auswahl definiert werden.

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](#) 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](#) einschließen, können Sie durch Filtern der Zeile bzw. Spalte auch interessante Ergebnisse erzielen (z.B. mit Hilfe von automatischen Berechnungen).

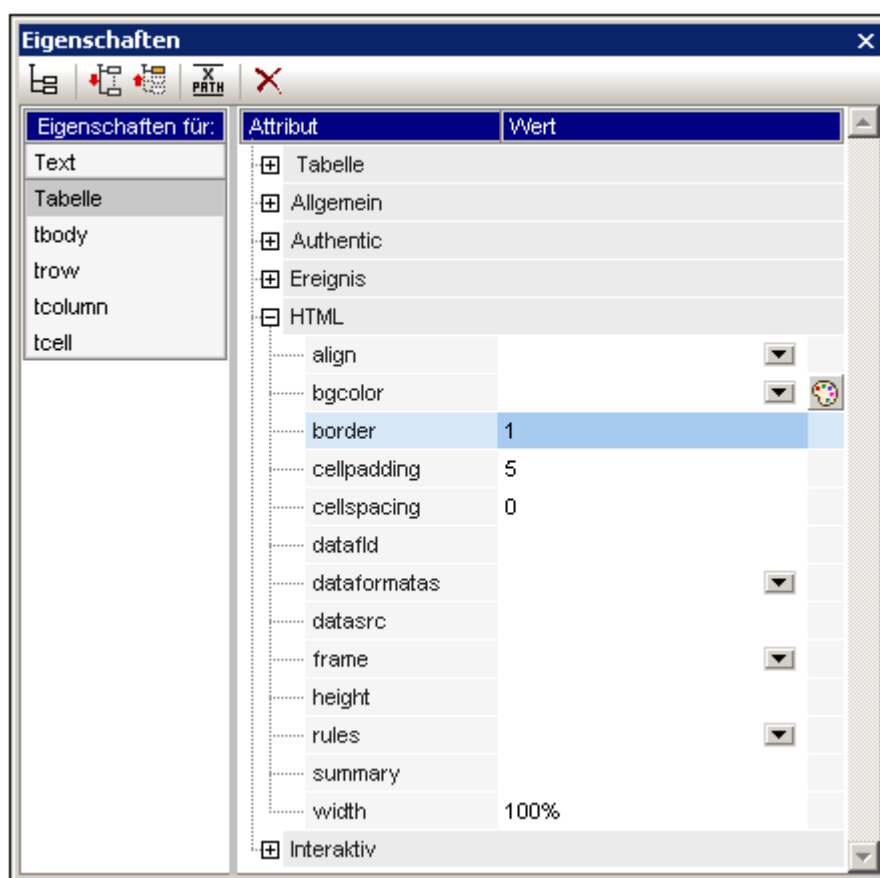
Formatieren von Tabellen

Statische und dynamische Tabellen

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

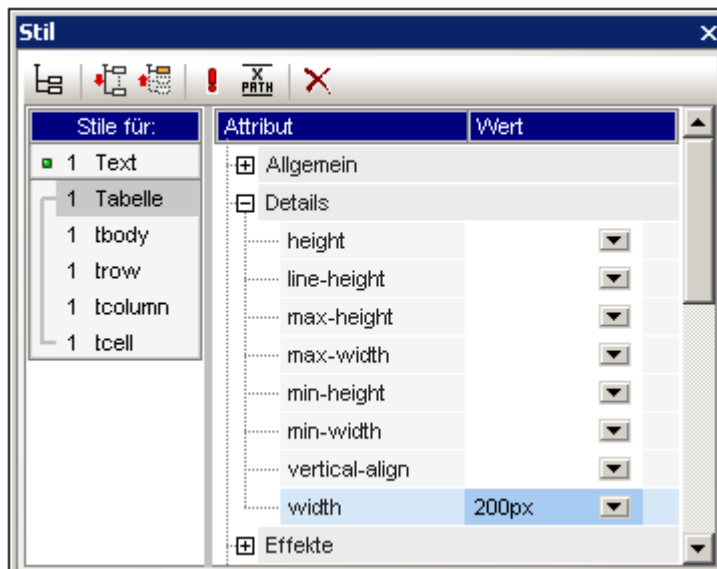
Seitenleiste "Eigenschaften"

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.



Seitenleiste "Stil"

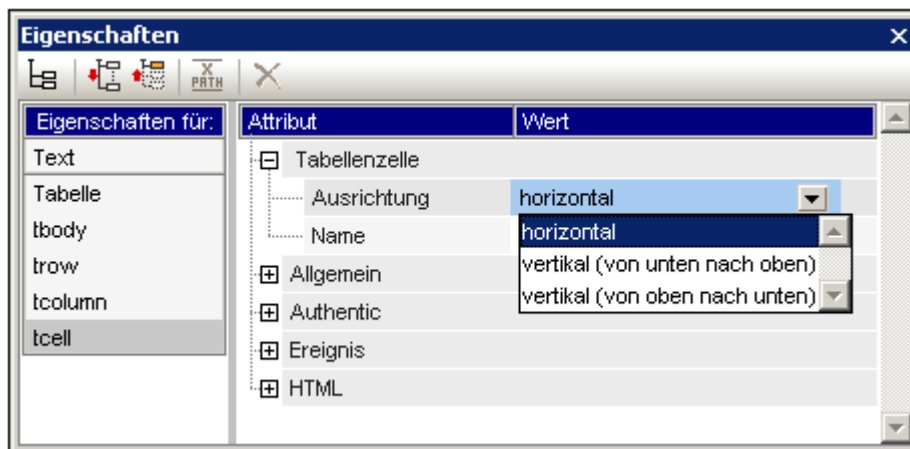
Die CSS-Tabellenformatierungseigenschaften stehen in der Seitenleiste "Stil" (*Abbildung unten*) zur Verfügung. CSS-Eigenschaften sind für die Tabellenkomponente und ihre Unterkomponenten verfügbar (Body, Zeile, Spalte und Zelle).



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 dazu 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 auch auf Text in [Textfeldern](#) 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 "Stile"
Tabelle	border, frame, rules; cellpadding, cellspacing; bgcolor; height, width (<i>wird durch height, width in der Seitenleiste "Stile" überschrieben, falls vorhanden</i>); align	Umrandungen (border) und Abstände (padding) in Feldstilen; height, width in der Gruppe "Details" (<i>überschreibt height und width in der Seitenleiste "Eigenschaften"</i>); Farbe, Schriftart und Textstile
Body	align, valign	height, vertical-align; Farbe, Schriftart und Textstile
Column	align, valign	width, vertical-align; Farbe, Schriftart und Textstile; Feldstile
Row	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](#) einfügen. Beachten Sie allerdings: Wenn Sie die 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: `tr`.
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" 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) in das `table`-Element ein. 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 abgeteilt werden soll, müssen Sie diese Option explizit für die jeweiligen Blöcke definieren.

Anzeigen von Zeilen und Spalten

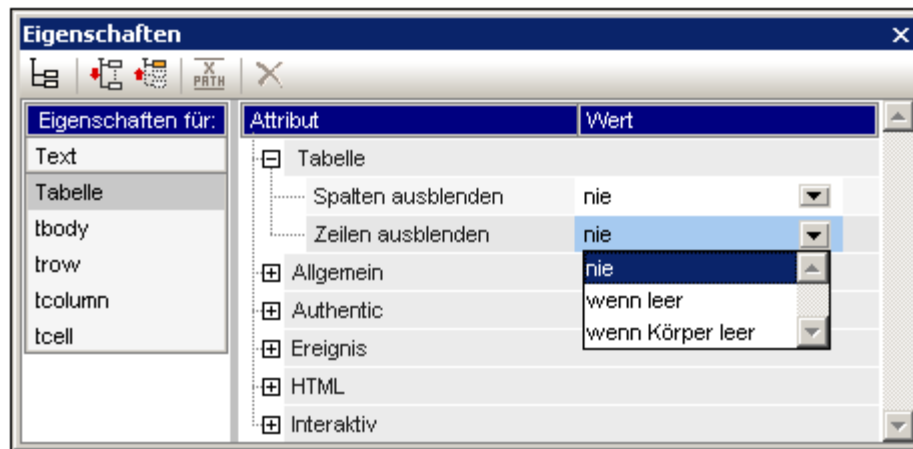
Für Tabellen stehen **nur** in der **HTML-Ausgabe** die folgenden Optionen zur Anzeige von Zeilen und Spalten zur Verfügung. Diese Funktionen werden in der **Authentic-Ansicht nicht unterstützt**. Um diese Funktionen verwenden zu können, muss als XSLT-Version des SPS XSLT 2.0 ausgewählt sein.

- 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 Eingabehilfe "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).

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](#)
2. [Definieren Sie die Formatierungsstile für die Tabelle](#)
3. [Fügen Sie die CALS/HTML-Tabellen in das SPS-Design ein](#)

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>
```

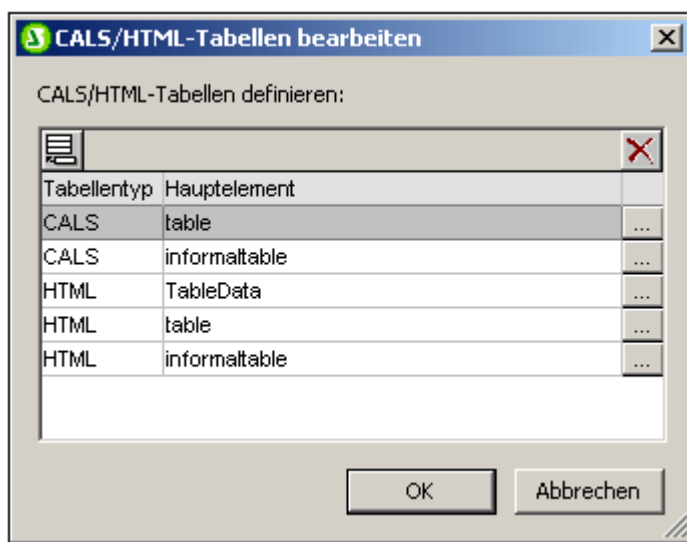
Alternativ dazu könnte die XML-Datenstruktur der einer HTML-Tabelle entsprechenden, mit dem Unterschied, dass die Elementnamen sich von denen im HTML-Tabellenmodell unterscheiden. Beispiel:

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

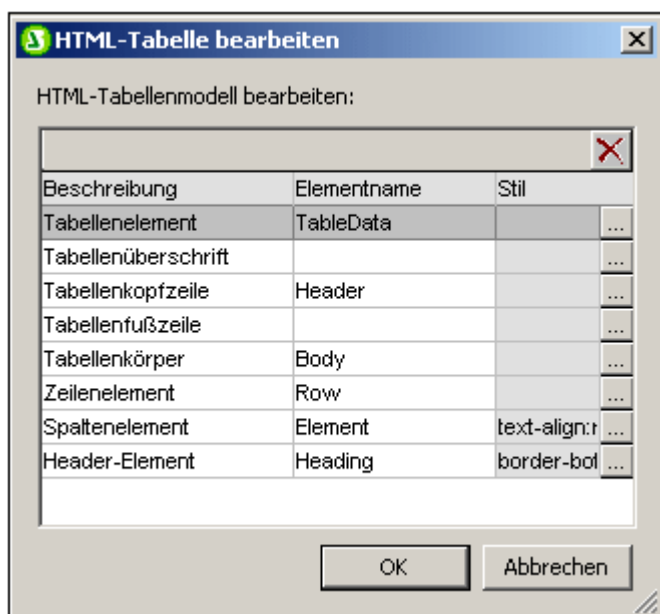
Anhand dieser im XML-Dokument definierten Tabellenstruktur kann auf direktem Weg eine Tabelle in den verschiedenen Ausgabeformaten generiert werden. Dazu muss diese XML-Datenstruktur als CALS- oder HTML-Tabelle definiert werden. Wenn die XML-Datenstruktur nicht als CALS- oder HTML-Format (Standardeinstellung) 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.



3. Es wird ein Dialogfeld angezeigt, in dem die Elemente des ausgewählten Tabellentyps angezeigt werden (*Abbildung unten*).



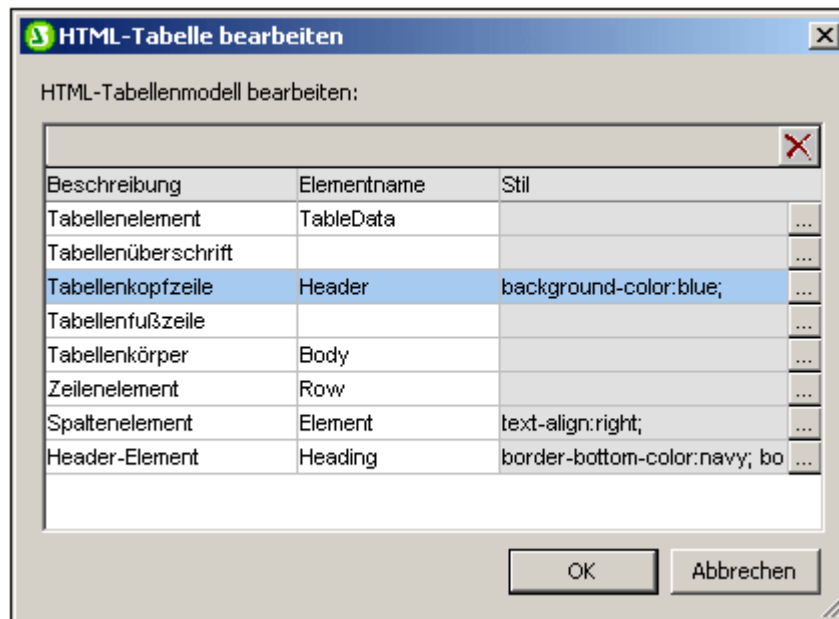
4. Die in diesem Dialogfeld aufgelisteten Elemente sind standardmäßig die Elementnamen im ausgewählten Tabellenmodell. Wenn das SPS-Schema Elemente dieses Namens enthält, so werden diese schwarz angezeigt (*wie in der Abbildung oben*). Wenn einer der aufgelisteten Elementnamen im Schema nicht vorhanden 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.
5. Klicken Sie auf **OK**, um diese XML-Datenstruktur als CALS- oder HTML-Tabelle zu definieren.
6. 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.
7. 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](#), so wird die Datenstruktur als Tabelle an die Ausgabegesendet. Um eine CALS/HTML-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*).



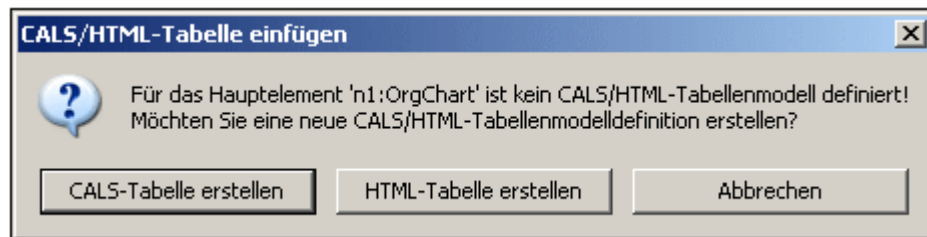
Um einem bestimmten Element einen Stil zuzuweisen, klicken Sie auf die Schaltfläche **Stile hinzufügen** für das jeweilige Element und weisen Sie die gewünschten Stile über die daraufhin angezeigte [Seitenleiste "Stile"](#) zu. Jeder Stil wird als eigenes CSS-Attribut zum jeweiligen Element hinzugefügt.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

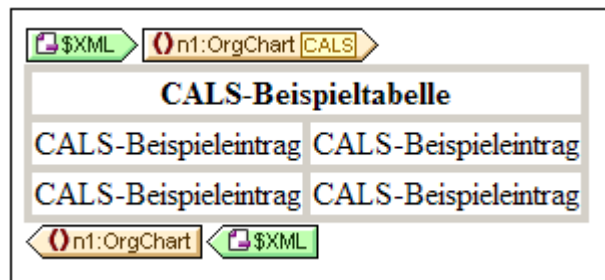
Einfügen einer CALS/HTML-Tabelle in das Design

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](#), so wird das Dialogfeld "CALS/HTML-Tabelle bearbeiten" (Abbildung oben) aufgerufen und Sie können das CALS- oder HTML-Tabellenelement 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](#) 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`.

8.5 Listen

Im SPS können zwei Arten von Listen erstellt werden:

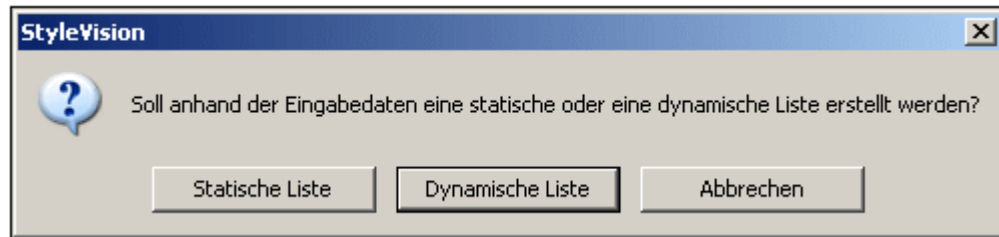
- [Statische Listen](#): Dies sind Listen, deren Inhalt direkt in das SPS eingegeben wird. Die Listenstruktur wird nicht dynamisch anhand der Struktur des XML-Dokuments erstellt.
- [Dynamische Listen](#): Dies sind Listen, deren Struktur und Inhalt dynamisch anhand des XML-Dokuments erstellt wird.

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

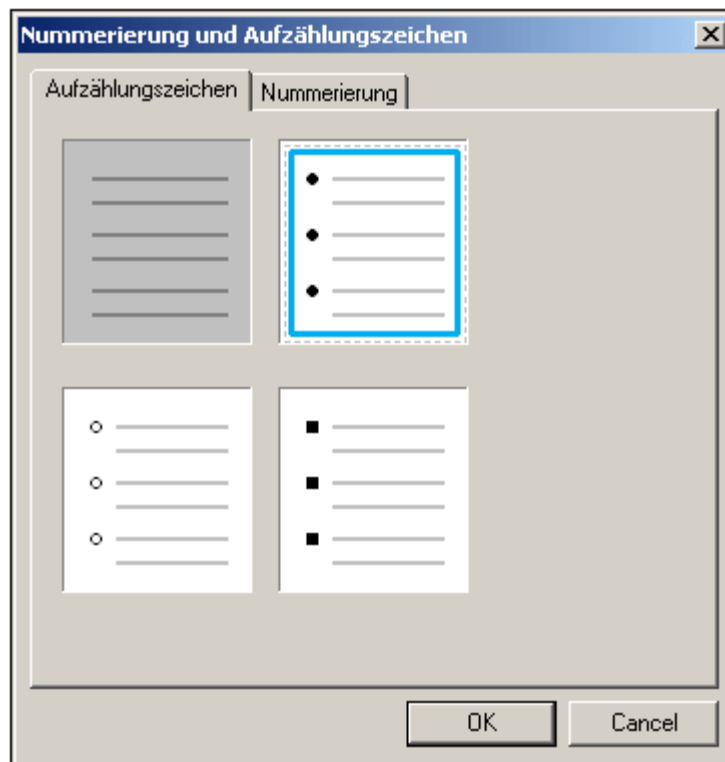
Statische Listen

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](#). 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](#). 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 auf die Schaltfläche "Nummerierung und Aufzählungszeichen" in der [Symbolleiste "Design-Element einfügen"](#) erstellen. Der erste Listeneintrag wird am Cursoreinfügapunkt erstellt.

Umwandeln von statischem Text in eine Liste

Es gibt zwei Methoden, um statischen Text in eine Liste umzuwandeln:

- Markieren Sie den gewünschten Text, klicken Sie auf [Einfügen | Nummerierung und Aufzählungszeichen](#), 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.
- Platzieren Sie den Cursor in ein Textfragment, klicken Sie auf [Einfügen | Nummerierung und Aufzählungszeichen](#), wählen Sie die gewünschte Art der Listenkennzeichnung aus und klicken Sie auf **OK**. Dieses Textfragment, d.h. der Teil, zwischen zwei CR-LF-Trennzeichen, wird als Listeneintrag erstellt.

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.

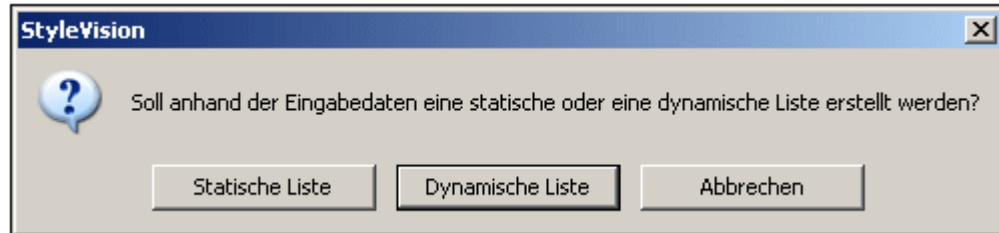
Grundprinzip

- 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.

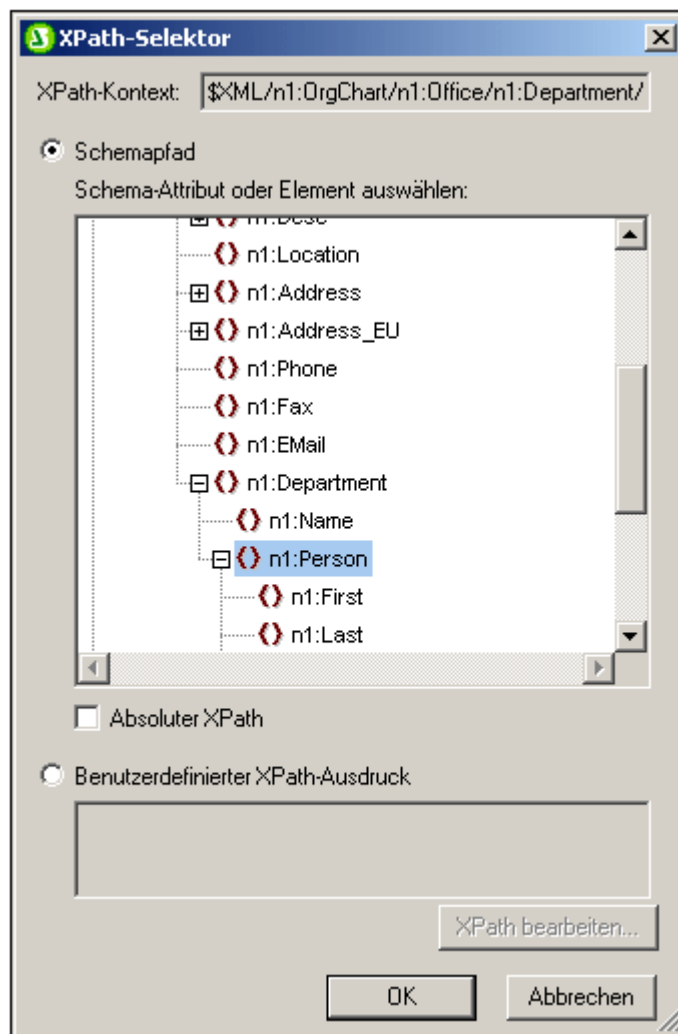
Erstellen einer dynamischen Liste

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](#). Daraufhin wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob Sie eine statische oder eine dynamische Liste erstellen möchten (*Abbildung unten*).

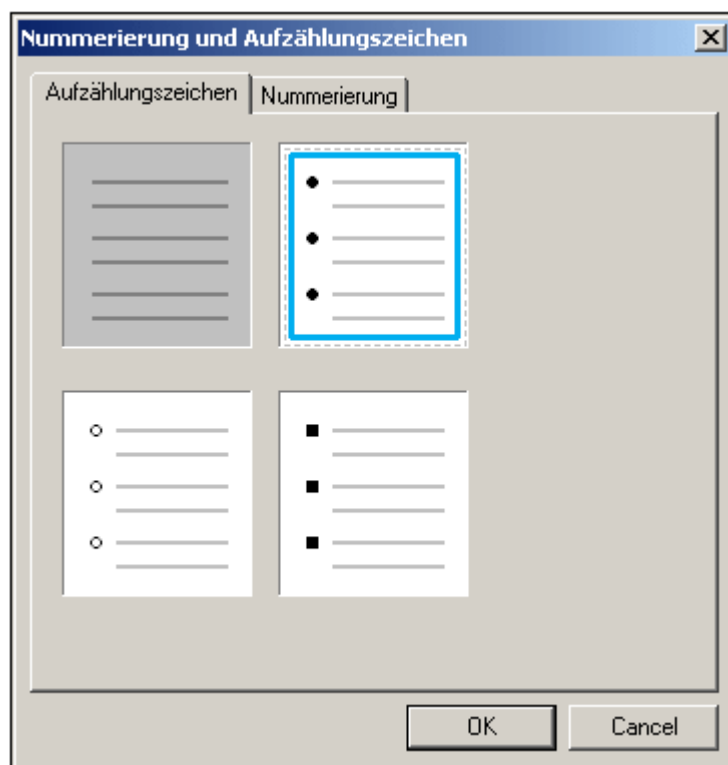


2. Klicken Sie auf **Dynamische Liste**. Daraufhin wird das Dialogfeld "XPath-Selektor" (*Abbildung unten*) angezeigt.
3. Beachten Sie, dass der XPath-Kontext im Dialogfeld "XPath-Selektor" der Kontext an der Einfügestelle ist und dass dieser Kontext im Dialogfeld nicht geändert werden kann. Wählen Sie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



8.6 Grafiken

Grafiken können in einem SPS auf zwei Arten verwendet werden:

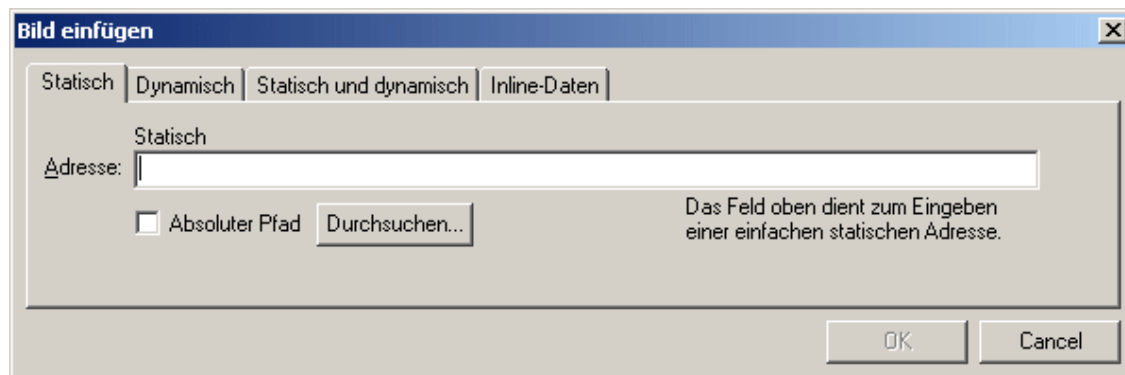
- Als [Bilder im Design-Dokument](#) und
- Als Symbolleistenschaltflächen der Authentic-Ansicht, um Markup-Tags zum XML-Dokument hinzuzufügen ([benutzerdefinierte Symbolleisten-Schaltflächen](#)).

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-URLs](#). Im Abschnitt [Referenz | Menü "Authentic" | Benutzerdefinierte Schaltflächen](#) finden Sie eine Beschreibung, wie Symbolleistenschaltflächen für die Authentic-Ansicht definiert werden können.

Bilder: URIs und Inline-Daten

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

Um ein Bild einzufügen, klicken Sie auf den Menübefehl [Einfügen | Bild einfügen](#), 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" eingegeben wird.
- Das Bild ist als Base 16- oder Base 64-Text in einer XML-Datei kodiert.

Einfügen einer Bilddatei

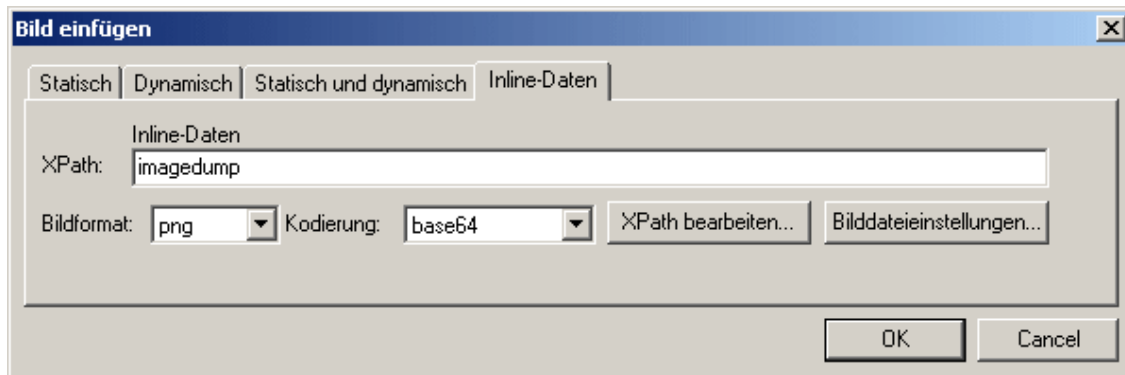
Ein Bilddatei wird in das Design eingefügt, indem ihre URI angegeben wird. Diese Datei wird während der Laufzeit aufgerufen und im Dokument platziert. Die Bild-URI kann auf drei Arten im Dialogfeld "Bild einfügen" (*Abbildung oben*) eingegeben werden:

- auf dem Register "Statisch" wird die URI direkt als absolute oder relative URI, z.B. `nanonull.gif` (*relative URI*; [siehe Abschnitt unten](#)) 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. 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.
- 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.

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 und der Authentic-Ansicht in StyleVision sowie in der Authentic-Ansicht in Altova-Produkten können Sie im Dialogfeld [Eigenschaften](#) 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.
- Bei der RTF-Ausgabe wird die URI des Bildes als Objektlink an die RTF-Datei übergeben und von der RTF-Applikation (normalerweise von MS Word) beim Öffnen der Datei aufgerufen. Wenn die URI relativ ist, muss der Pfad relativ zum Ordner der RTF-Datei sein. Zur Darstellung der RTF-Vorschau in StyleVision wird im selben Ordner wie die SPS-Datei eine temporäre RTF-Datei gespeichert. Daher müssen zur Darstellung der RTF-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 "Bildeigenschaften" 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".

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
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`, befindet sich im Ordner:

```
C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/
StyleVisionExamples/Tutorial/Images
```

Anpassen der Größe von Bildern in der RTF-Ausgabe

Das Anpassen der Größe eines Bildes in der RTF-Ausgabe wird nur für JPG- und PNG-Bilder unterstützt. Beachten Sie dabei die folgenden Punkte:

- Die Anpassung der Größe von Bildern wird nur in Designs unterstützt, für die XSLT 2.0 und nicht XSLT 1.0 verwendet wird.
- Die Bildattribute `height` und `width` müssen in der Gruppe *Details* der Seitenleiste "Stile" definiert werden. Die Attribute `height` und `width` in der Gruppe *HTML* der Seitenleiste "Eigenschaften" werden nicht verwendet.
- Es werden nur absolute Maßeinheiten (px, cm, Zoll, usw.) unterstützt. Prozentwerte

werden nicht unterstützt.

- JPG- und PNG-Bilder werden in die RTF-Dateien eingebettet. Dies wird mittels einer proprietären Altova XSLT 2.0-Erweiterungsfunktion implementiert.

Einbettung von Bildern in RTF

Bilder können in der RTF-Ausgabe eingebettet werden (bei Verwendung von XSLT 2.0) oder verknüpft werden. Diese Einstellung wird einzeln für jedes SPS%> vorgenommen. Um Bilder einzubetten, gehen Sie folgendermaßen vor:

1. Öffnen Sie das Dialogfeld "Eigenschaften" (**Datei | Eigenschaften**), wobei das gewünschte SPS aktiv sein muss.
2. Aktivieren Sie das Kontrollkästchen "Bilder einbetten" (Standardeinstellung ist aktiviert). Beachten Sie, dass die Bilder nur eingebettet werden, wenn XSLT 2.0 als XSLT-Version des aktiven SPS eingestellt ist.
3. Nachdem Sie die Einstellung vorgenommen haben, klicken Sie auf **OK**. Die Einstellung wird für das aktive SPS gespeichert.

Um die Einstellung für ein anderes SPS vorzunehmen, öffnen Sie das SPS und machen Sie es zum aktiven und wiederholen Sie die oben angeführten Schritte.

Wenn das Kontrollkästchen deaktiviert ist, werden die Bilder entsprechend dem Bilddateipfad, der unter den Bildeigenschaften definiert ist, mittels Verknüpfung inkludiert (wählen Sie das Bild aus und wählen Sie die Url in der [Eingabehilfe "Eigenschaften"](#) aus). Informationen darüber, wie Pfade aufgelöst werden, finden Sie im Abschnitt [Bild-URLs](#)

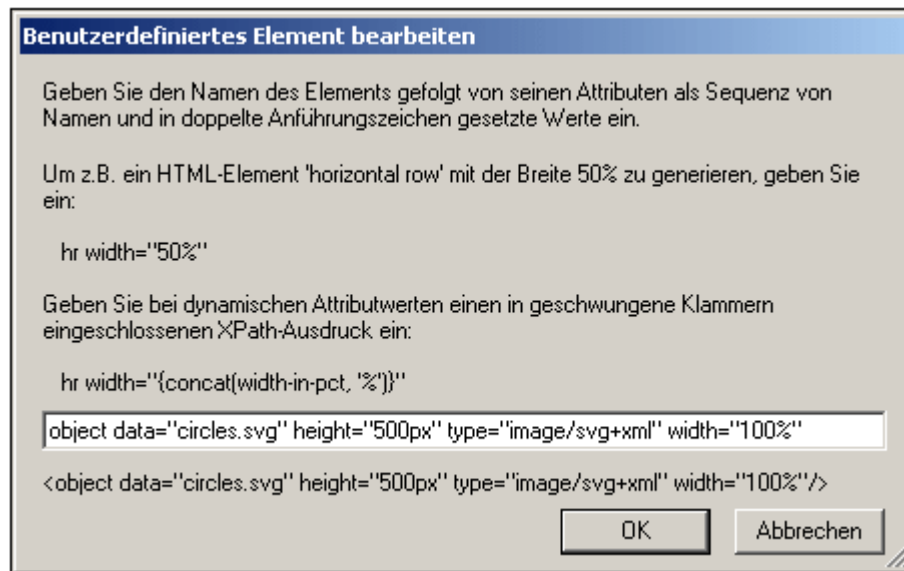
Anmerkung: Das RTF-Format unterstützt eingebettete Bilder nur für EMF, JPG, und PNG-Dateien.

SVG-Bilder in HTML

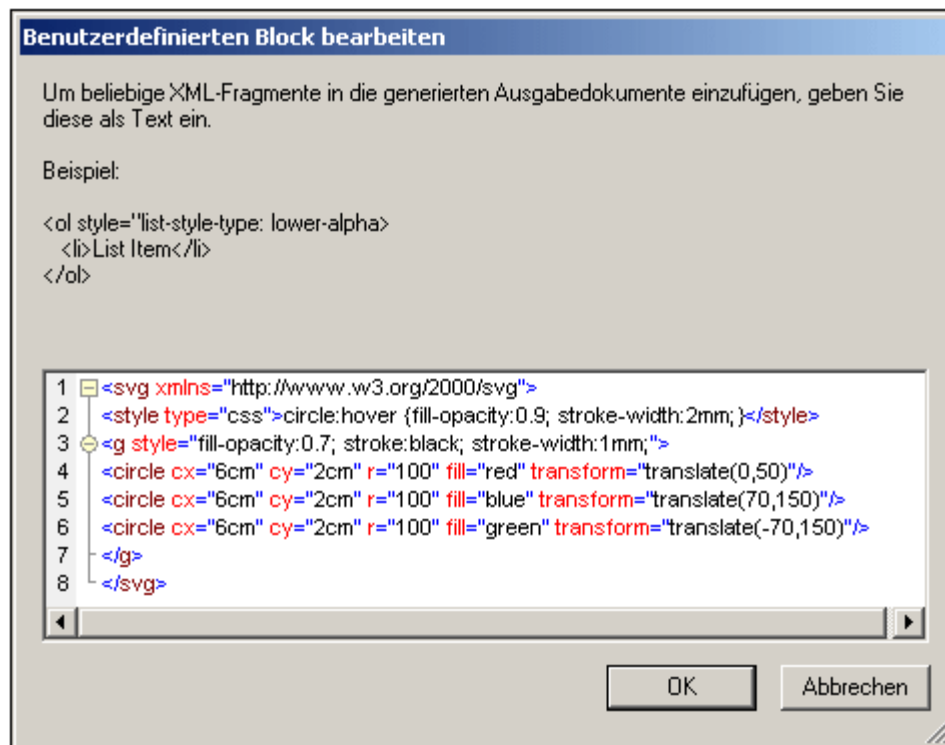
Wenn als Bild eine externe SVG-Datei mit Code für Maus-Events verwendet wird, so wird die SVG-Datei innerhalb des Bilds dargestellt und ist nicht mehr interaktiv. Um diese Einschränkung zu umgehen, können Sie die externe SVG-Bilddatei als Objekt verwenden oder das SVG-Codefragment als benutzerdefinierten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



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](#): c: / Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples\Tutorial\Images\Images.sps. Das Bild-Dokument (Images.xml und Images.sps) besteht aus drei Teilen:

- Im ersten Teil wird gezeigt, wie Textstatussymbole in der Authentic-Ansicht verwendet werden (können nur in der Enterprise und der Professional Edition erstellt werden). Wenn Sie die Datei Images.xml in der Authentic-Ansicht von XMLSpy, Authentic Desktop oder Authentic Browser öffnen, können Sie die Verwendung von Textstatussymbolen ausprobieren. Beachten Sie, dass Textstatussymbole in der Authentic-Vorschau von StyleVision nicht zur Verfügung stehen, daher können Sie diese Funktion in StyleVision nicht ausprobieren. Eine Anleitung zur Erstellung von Textstatussymbolen finden Sie im Abschnitt [Referenz | Menü "Authentic" | Benutzerdefinierte Symboleleisten-Schaltflächen](#) dieses Benutzerhandbuch.
- 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 Vorschauenstern ansehen.

8.7 Formularsteuerelemente

Nodes im XML-Dokument können als Dateneingabeelemente erstellt werden (z.B. als Eingabefelder und Auswahllisten). Dateneingabeelemente sollen die Bearbeitung des Dokuments in der Authentic-Ansicht erleichtern. Z.B. zeigt ein Eingabefeld in der Authentic-Ansicht an, dass an dieser Stelle eine Eingabe erwartet wird. In einer Auswahlliste steht dem Benutzer eine Reihe von Werten zur Verfügung, aus denen er auswählen kann. Bei Eingabe von Daten in ein Dateneingabeelement, werden diese Daten als Elementinhalt oder als Attributwert in das XML-Dokument eingefügt. In der HTML- und RTF-Ausgabe wird ein Dateneingabeelement so dargestellt, wie es in der Authentic-Ansicht angezeigt wird, oder es wird auf eine möglichst ähnliche Art wiedergegeben. Beachten Sie, dass Dateneingabeelemente Eingaben in das XML-Dokument zulassen und dass dies in der HTML-Ausgabe nicht funktioniert.

Allgemeine Funktionsweise

Im Folgenden finden Sie eine Liste von in StyleVision verfügbaren Dateneingabeelementen sowie eine Erklärung, wie die Daten bei jedem dieser Elemente in das XML-Dokument eingegeben werden.

Dateneingabeelement	Daten in der XML-Datei
Eingabefeld (Textfeld)	Text, der vom Benutzer eingegeben wird
Mehrzeiliges Eingabefeld	Text, der vom Benutzer eingegeben wird
Auswahlliste	Der Auswahl des Benutzers wird ein Wert zugeordnet.
Kontrollkästchen	Der Auswahl des Benutzers wird ein Wert zugeordnet
Optionsfeld	Der Auswahl des Benutzers wird ein Wert zugeordnet
Schaltfläche	Der Auswahl des Benutzers wird ein Wert zugeordnet

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.

Allgemeine Verwendung

Um ein Dateneingabeelement 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:

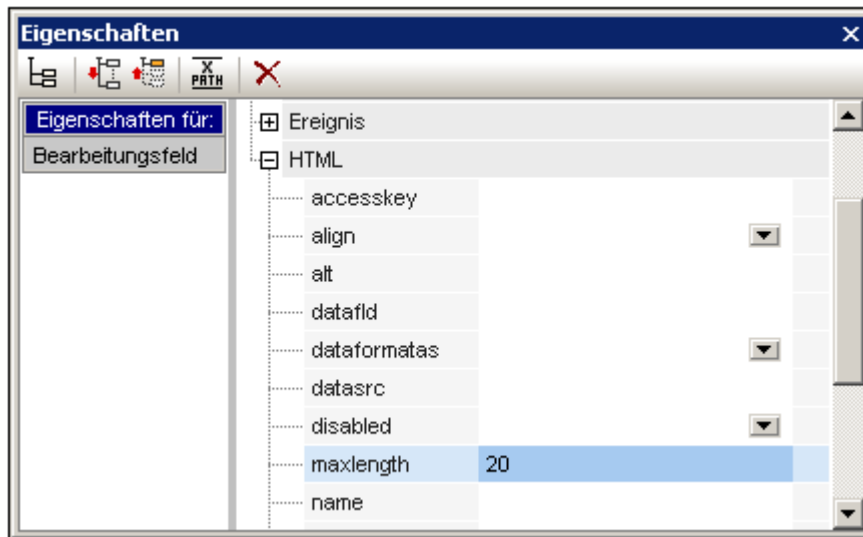
- Daten können nur in der Authentic-Ansicht in Dateneingabeelemente eingegeben werden.
- Dateneingabeelemente können auch erstellt werden, indem Sie den aktuellen Komponententyp eines Node in ein Dateneingabeelement ändern. Klicken Sie dazu mit der rechten Maustaste auf den Node und wählen Sie **Ändern in**.
- In der HTML- und RTF-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.
- Bei einigen Dateneingabeelementen wie z.B. Kontrollkästchen, die in gedruckter Form nicht korrekt wiedergegeben werden können, wird eine andere Darstellung implementiert.

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 Text, den der Benutzer der Authentic-Ansicht in diese Felder eingibt, wird in den XML-Node eingegeben, für den das Feld erstellt wurde. Der Inhalt dieses Node wird im Eingabefeld oder im mehrzeiligen Eingabefeld angezeigt.

Bearbeiten der Eigenschaften von Eingabefeldern

Sie können die HTML-Eigenschaften von Eingabefeldern ändern, indem Sie das Eingabefeld auswählen und seine HTML-Eigenschaften in der Seitenleiste "Eigenschaften" (siehe Abbildung unten) bearbeiten.

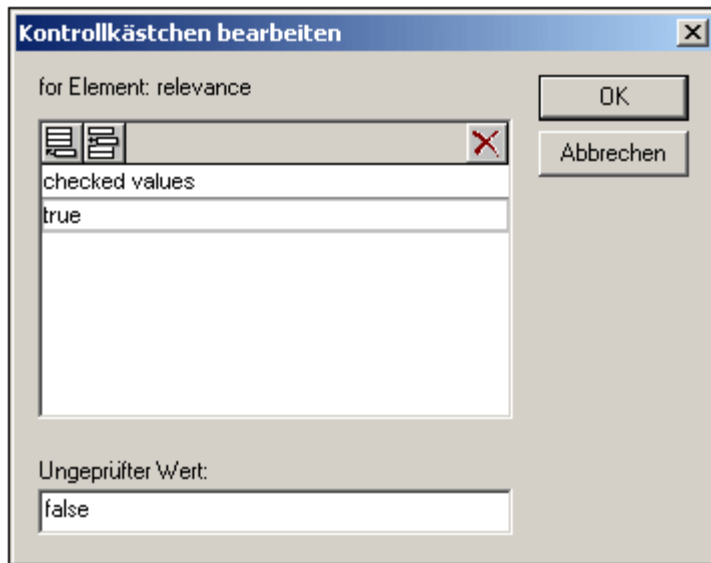


So können Sie z.B. wenn das Eingabefeld ausgewählt ist, im Fenster "Eigenschaften" den Eintrag `editfield` auswählen und in der Eigenschaftsgruppe `HTML` die Eigenschaft `maxlength` definieren, indem Sie auf ihr Wertefeld doppelklicken und einen Wert eingeben.

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

Kontrollkästchen

Sie können ein Kontrollkästchen als Dateneingabeelement erstellen und dadurch die Eingabe des Benutzers auf eine von zwei Auswahlmöglichkeiten einschränken. Im Dialogfeld "Kontrollkästchen bearbeiten" (siehe unten) können Sie die XML-Werte angeben, auf die das angeklickte bzw. deaktivierte Kontrollkästchen gemappt werden soll.



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:

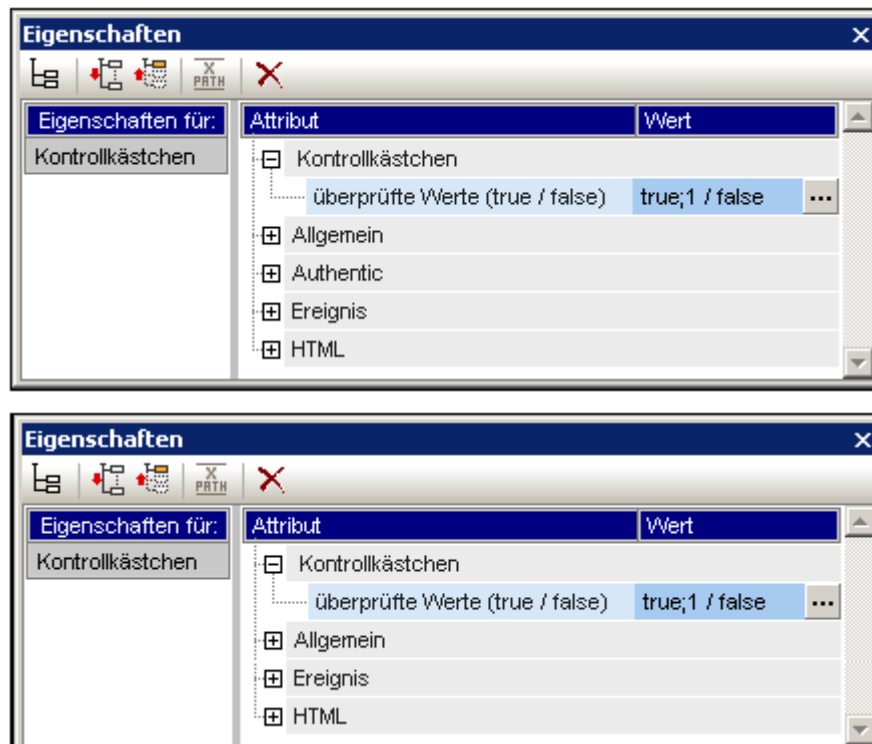
Wenn in der Authentic-Ansicht ein neues `Name` (oder Kontrollkästchen)-Element erstellt wird, ist der XML-Wert leer (es wird nicht der Wert für das deaktivierte Kontrollkästchen verwendet). Der Wert "deaktiviert" wird erst eingegeben, nachdem das Kontrollkästchen zuerst aktiviert und dann deaktiviert wurde. Um einen Standardwert in einem Node zu haben, erstellen Sie eine XML-Vorlagendatei, die den Standardwert enthält.

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.



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.

Auswahllisten

Eine Auswahlliste ist eine Dropdown-Liste, über die dem Benutzer der Authentic-Ansicht eine Liste von Optionen zur Auswahl zur Verfügung steht. Die ausgewählte Option ist auf einen Wert gemappt, der in das XML-Dokument eingegeben wird. Die Zuordnung des Eintrags in der Dropdown-Liste zum XML-Wert ist im SPS definiert.

Die Zuordnungen im Dialogfeld "Auswahlliste bearbeiten" können auf drei Arten vorgenommen werden:

- über die Schema-Enumerationen für den ausgewählten Node. In diesem Fall ist der sichtbare Eintrag (in der Dropdown-Liste) mit dem XML-Wert identisch.
- 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.
- über die Ergebnissequenz eines XPath-Ausdrucks zum aktuellen Node. Die Einträge in der Ergebnissequenz werden als die Einträge der Dropdown-Liste angezeigt und der Listeneintrag, der vom Benutzer der Authentic-Ansicht ausgewählt wird, wird als Wert des Node eingetragen. 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 in der Authentic-Ansicht 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..

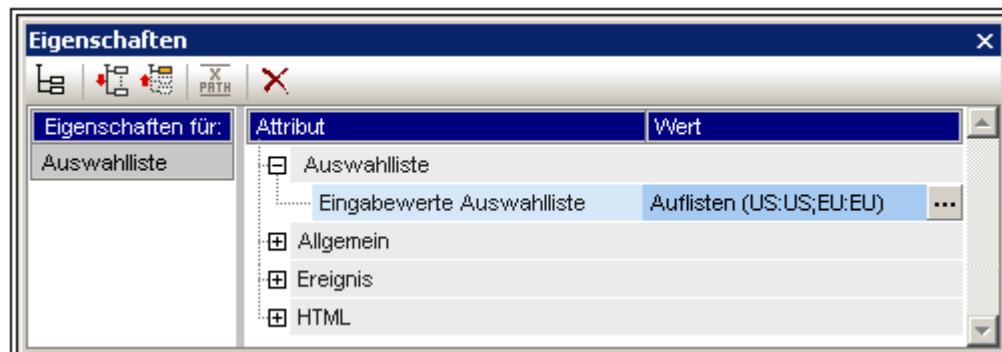
Aufrufen des Dialogfelds "Auswahlliste bearbeiten"

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.

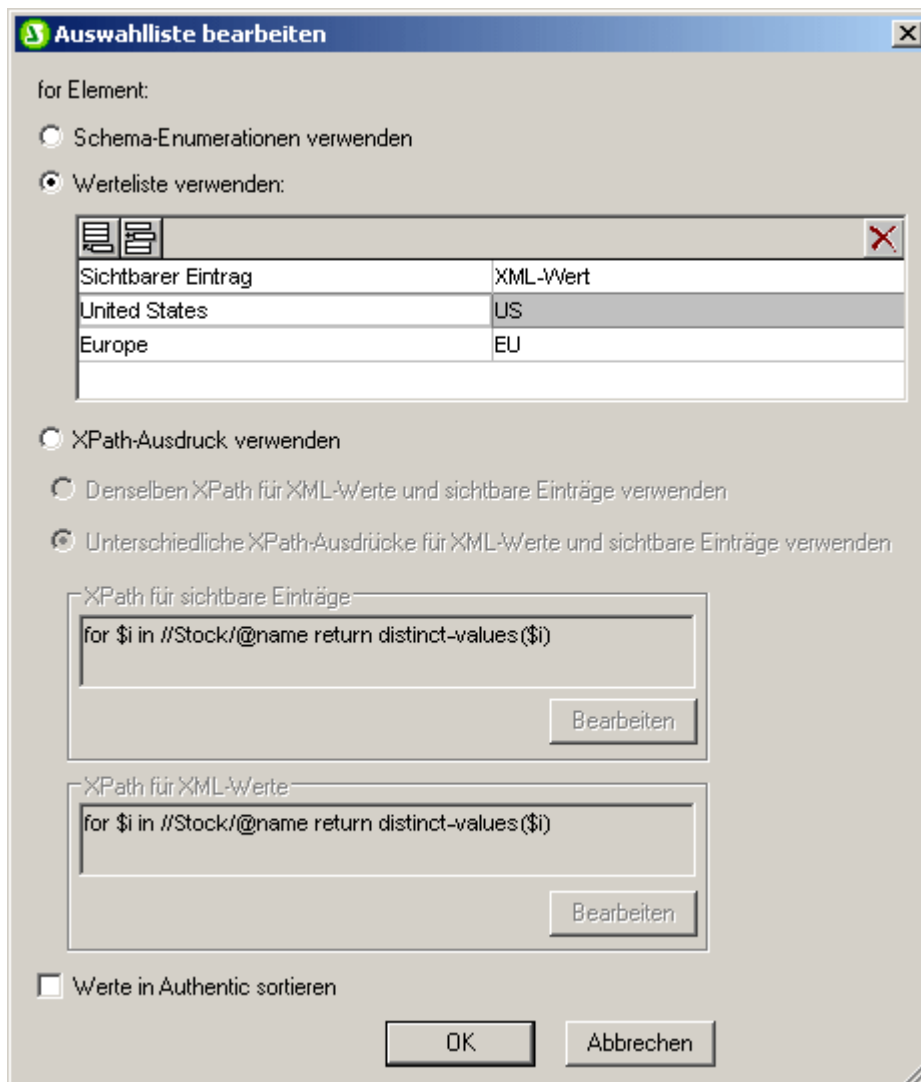




3. Klicken Sie auf die Bearbeiten-Schaltfläche  der Eigenschaft `Ursprung des Inhalts`. Daraufhin wird das Dialogfeld "Auswahlliste bearbeiten" angezeigt.

Verwenden des Dialogfelds "Auswahlliste bearbeiten"

Unten sehen Sie das Dialogfeld "Auswahlliste bearbeiten".



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.

Sichtbarer Eintrag	XML-Wert
Area Charts	Area Charts
Bar Chart 2D	Bar Chart 2D
Bar Chart 3D	Bar Chart 3D
Bar Chart 3D Grouped	Bar Chart 3D Grouped

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. Wenn Die Einträge aus einer Datenbank stammen, muss das Datenbankschema als Hauptschema definiert werden. Wenn die Einträge aus einer Datenbank stammen, bei der es sich nicht um das Hauptschema handelt, muss - auch wenn die Vorlage leer sein muss - in das Design eine Vorlage für die Datenbankzeile eingefügt werden, an der der XPath-Ausdruck ausgewertet wird.

Stellen Sie in diesem Fall außerdem sicher, dass alle Instanzen der betreffenden Zeile [abgerufen werden](#).

- Sie können die HTML-Eigenschaften einer Auswahlliste ändern, indem Sie sie auswählen und die HTML-Eigenschaften anschließend in der Seitenleiste "Eigenschaften" ändern.

Optionsfelder, Schaltflächen

Es gibt zwei Arten von Schaltflächen: Optionsfelder und Schaltflächen. Mit Hilfe von Optionsfeldern kann der Benutzer der Authentic-Ansicht Daten in die XML-Datei eingeben. Schaltflächen gestatten die Dateneingabe in der Authentic-Ansicht nicht, eignen sich jedoch zum Auslösen von Ereignissen in der HTML-Ausgabe.

Optionsfelder

Wenn Sie Optionsfelder in das SPS einfügen, kann der Benutzer zwischen mehreren Alternativen wählen. Jedes eingefügte Optionsfeld ist auf einen XML-Wert gemappt. Der Benutzer wählt ein Optionsfeld aus. Die Optionsfelder für einen Node schließen einander gegenseitig aus; es kann immer nur ein Optionsfeld ausgewählt werden. Der dazugehörige XML-Wert wird als Wert des Node eingegeben. Diese Funktion wird folgendermaßen verwendet: Es wird ein Node erstellt, bei dem eine mehrmalige Dateneingabe über ein Optionsfeld erforderlich ist. Geben Sie für jedes Optionsfeld (i) statischen Text für den Wert ein, den der Benutzer auswählt, und (ii) einen unterschiedlichen XML-Wert für jedes Optionsfeld. Um den XML-Wert eines Optionsfelds zu bearbeiten, wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag "Optionsfeld" aus, klicken Sie anschließend auf die Bearbeiten-Schaltfläche der Eigenschaft *Optionsfeld*. Daraufhin wird das Dialogfeld "Optionsfeld bearbeiten" angezeigt.

Schaltflächen

Über die Option "Schaltfläche" können Sie eine Schaltfläche einfügen und den Text auf der Schaltfläche definieren. Dies ist hilfreich, wenn Sie in der generierten HTML-Ausgabe Scripts mit Schaltflächen-Ereignissen verknüpfen möchten. Beachten Sie jedoch, dass eine Schaltfläche nicht auf einen XML-Wert gemappt wird und die Eingabe von Daten in der Authentic-Ansicht nicht gestattet. [Der Wert des Parent Node einer Schaltfläche kann allerdings vom Benutzer der Authentic-Ansicht ausgewählt werden](#) und das SPS kann so konzipiert werden, dass die Darstellung sich ändert, je nachdem, welchen Wert der Benutzer der Authentic-Ansicht auswählt.

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.

8.8 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-URLs für Hyperlinks erstellt werden können.

Im Abschnitt [Lesezeichen und Hyperlinks](#) wird beschrieben, wie man im Dokument dynamische und statische Hyperlinks anlegt und wie man sowohl zu Lesezeichen als auch zu externen Dokumenten Links erstellt.

8.9 Barcodes

Mit Hilfe des Design-Elements "Barcode" können im Ausgabedokument Barcodes (*Abbildung unten*) generiert werden. [Fügen Sie das Barcode-Design-Element](#) an der gewünschten Stelle im Design-Dokument ein und definieren Sie seine [Eigenschaften](#).

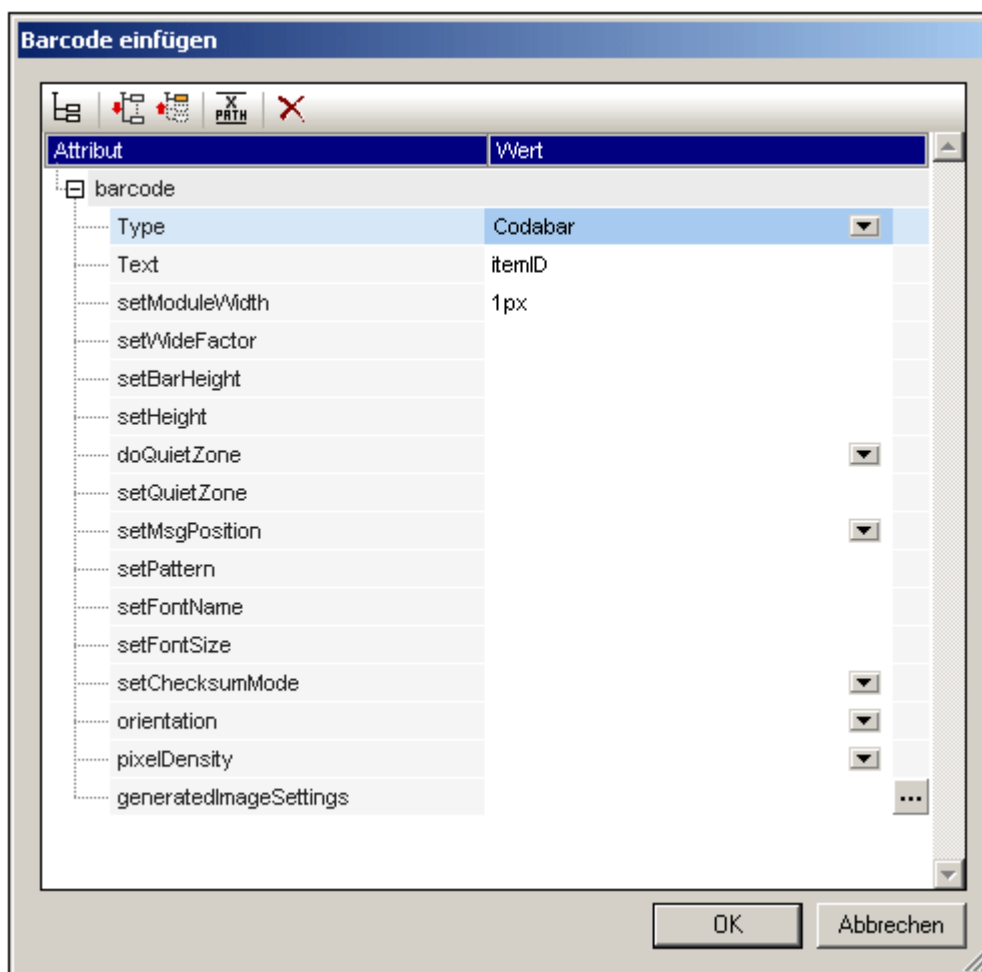


Wichtig: Damit Barcodes funktionieren, muss eine Java Runtime-Umgebung installiert sein. Diese muss zumindest Version 1.4 sein und in einer Bit-Version vorhanden sein, die der Bit-Version des auf Ihrem System installierten StyleVision-Pakets entspricht: 34-Bit oder 64-Bit.

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 das Barcode-Symbol aus der Symbolleiste an die gewünschte Stelle im Design ziehen. Sie können auch ein Element aus der Schema-Struktur in die Design-Ansicht ziehen und dann 'Barcode erstellen' auswählen. Daraufhin wird das Barcode-Dialogfeld "Eigenschaften" angezeigt (*Abbildung unten*).



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](#). 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](#) 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 *Wertefeld* der Eigenschaft ein. Der XPath-Ausdruck wird innerhalb des aktuellen Kontext-Node ausgewertet.
3. Nachdem Sie die Eigenschaften definiert haben, klicken Sie auf **OK**. Daraufhin wird der Barcode 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 *Message* und *Type* müssen definiert werden; die anderen Eigenschaften sind optional.

- *Type*: Das Barcode-System, mit dem die Message interpretiert wird, z.B. EAN und UPC.
- *Message*: 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

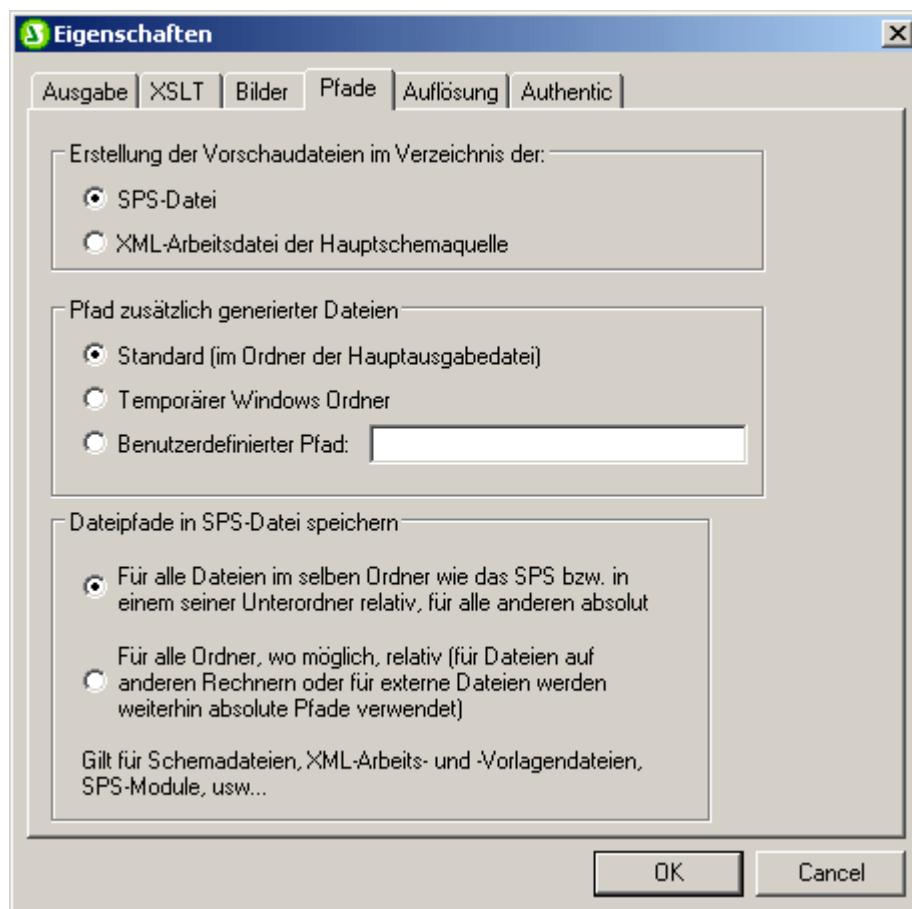
Muster werden verwendet, um die Eingabe-Zeichenfolge im Barcode lesbarer zu machen. In den Mustern wird jedes Zeichen im Text der Eingabenachricht durch einen Unterstrich "_" dargestellt. Alle anderen Zeichen, die im Muster enthalten sind, werden an den zugehörigen Orten in den Text der Ausgabenachricht 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 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 Befehls **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.

8.10 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](#) 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](#) 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](#) können gezeichnet, formatiert, positioniert und im Stapel der Layout-Objekte (Layout-Felder und andere Linien) nach vorne oder hinten gebracht werden.

Layout Container

Ein Layout Container hat die folgenden Eigenschaften:

- Er kann innerhalb eines Dokumentablaufs, also innerhalb einer Vorlage, [eingefügt](#) 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](#) weiter unten.
- Ein [Layout-Raster](#) und eine [Zoom-Funktion](#) erleichtern das Positionieren von Objekten im Layout Container.
- Er kann [Stileigenschaften](#) wie z.B. Ränder, Hintergrundfarben, Schriftarteigenschaften, usw. haben.
- Er kann [Layout-Felder und Linien](#) enthalten, aber kein anderes Design-Element. (Alle Design-Elemente müssen in Layout-Felder platziert werden.)
- Er kann eine [Grafikvorlage](#) 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

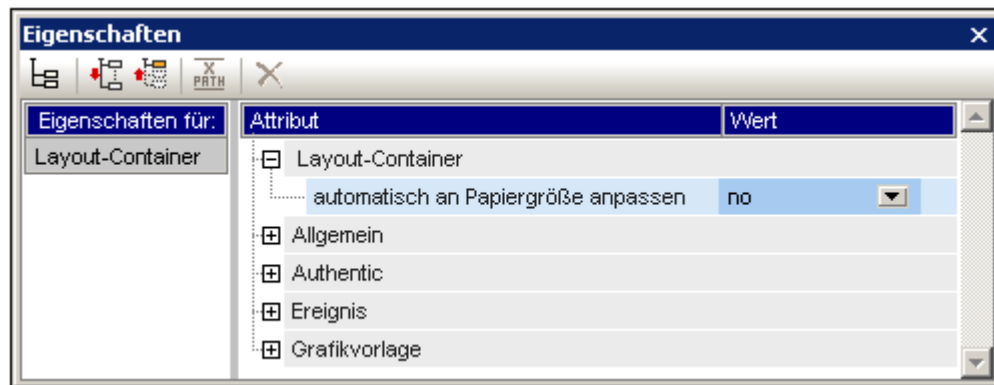
Um einen Layout Container einzufügen, platzieren Sie den Cursor an die gewünschte Stelle im Design und klicken Sie in der [Symbolleiste "Design-Elemente einfügen"](#) auf die Schaltfläche **Layout Container einfügen**. Daraufhin erscheint ein Dialogfeld, in dem Sie gefragt werden, ob Sie den Layout Container 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 wie 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:

- *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" ([Extras | 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.

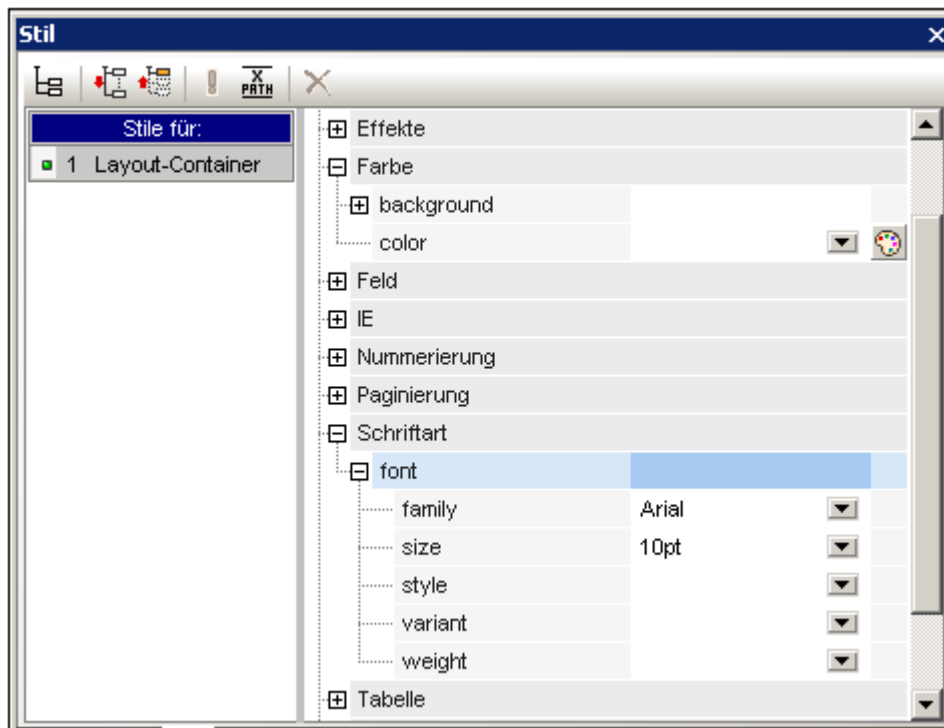
Zoomen

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*).

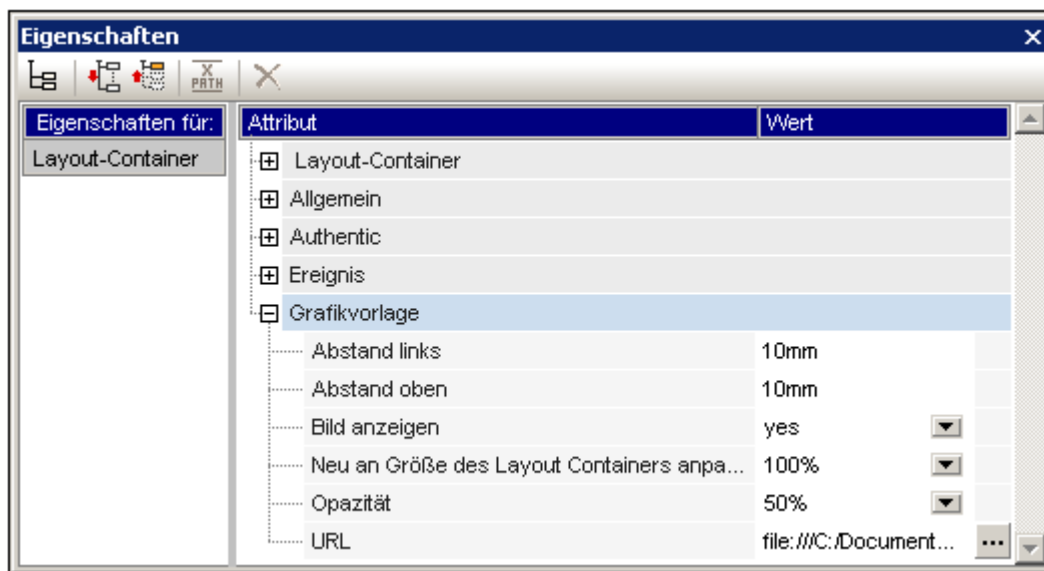
Layout Container-Inhalt

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 Ausgabeanzeige, 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.

Anmerkung: Wenn Design-Element Markup-Tags - wie z.B. Vorlagen-Node-Tags - in ein Layout-Feld eingefügt werden, wirkt sich dies auf die Abstände im Layout aus, da die Tags Platz im Layout einnehmen. Um diese Störquelle zu vermeiden und damit das Design besser mit der Grafikvorlage übereinstimmt, blenden Sie die Tags mit Hilfe der Funktion "Design-Markup-Symbole ausblenden" aus.

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 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](#)
- [Auswählen und Verschieben von Layout-Feldern](#)
- [Anpassen der Größe eines Layout-Felds](#)
- [Definieren von Stileigenschaften für ein Layout-Feld](#)
- [Einfügen von Inhalt in das Layout-Feld](#)
- [Stapelreihenfolge von Layout-Feldern](#)

Einfügen eines Layout-Felds

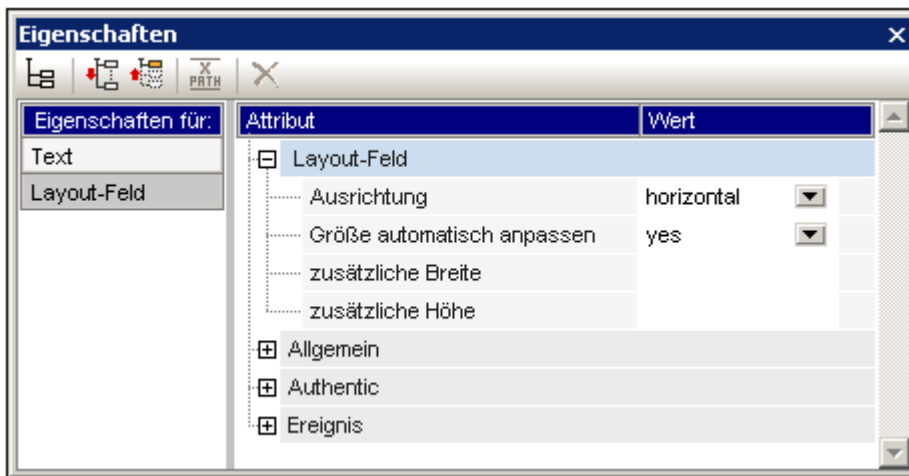
Ein Layout-Feld kann nur in einen [Layout Container](#) eingefügt werden. Um ein Layout-Feld hinzuzufügen, klicken Sie zuerst in der [Symbolleiste "Design-Elemente einfügen"](#) auf die Schaltfläche "Layout-Feld einfügen" und anschließend an die Stelle im Layout Container, an der das Layout-Feld eingefügt 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 darin eingefügten 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](#). 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).

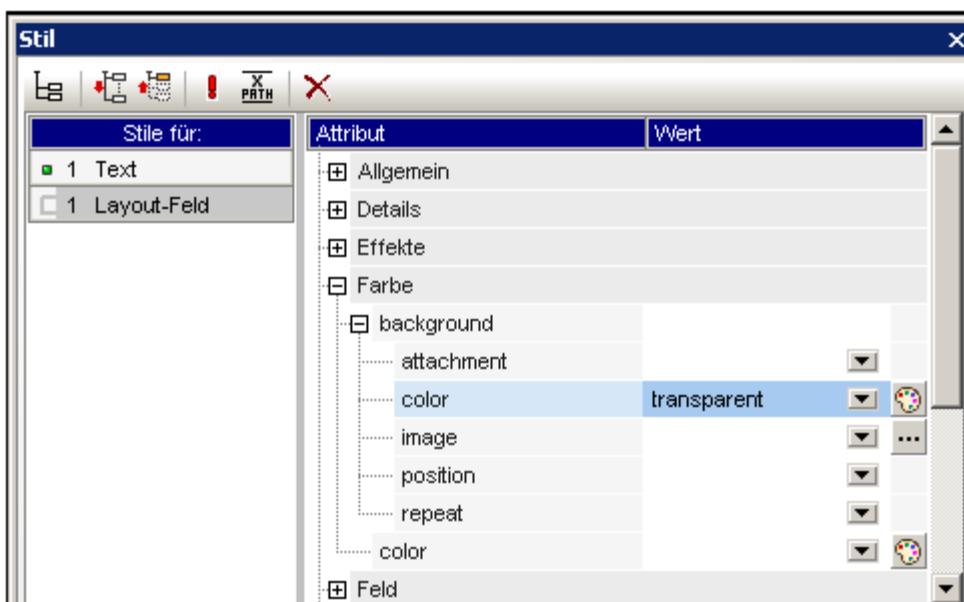


Die Eigenschaften *zusätzliche Höhe* und *zusätzliche Breite* geben die Längen an, die zusätzlich zur optimalen Größe (bei Auswahl der Option "Automatisch an die optimale Größe anpassen") anfallen. Diese zusätzlichen Längen erhält man, wenn man die Größe eines Layout-Felds manuell anpasst. Wenn Sie umgekehrt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



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](#) noch eine [Layout-Linie](#) 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.
- Würden Markup-Tags in einem Layout-Feld angezeigt, würde dies die WYSIWYG-Ansicht beeinträchtigen. Um zu sehen, wie das Layout in der Ausgabe aussieht, können Sie die Markup-Tags mit Hilfe der Funktion "Markup-Symbole ausblenden" ([Schaltfläche in der Symbolleiste](#)) aus der Ansicht ausblenden.
- 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.



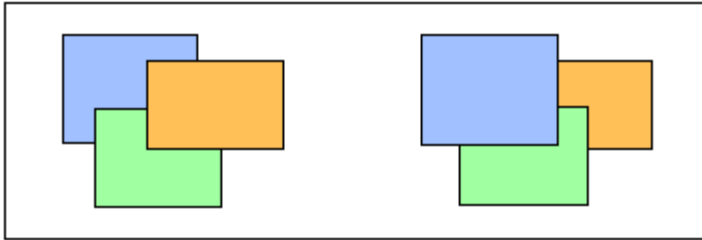
Beachten Sie dazu die folgenden Punkte:

- 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](#) 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

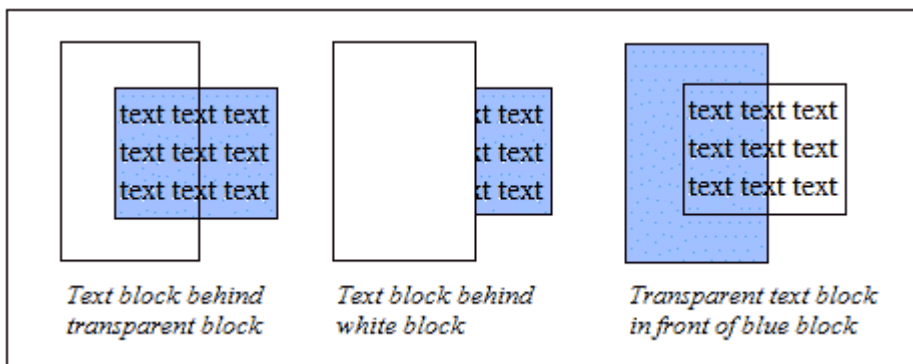
sichtbar; die anderen werden zum Teil oder zur Gänze verdeckt.



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](#) hinzufügen. Jede Linie kann relativ zu den anderen Elementen im Stapel verschoben werden.

Linien

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

Einfügen einer Linie

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

1. Klicken Sie in der Symbolleiste [Design-Elemente einfügen](#) 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.

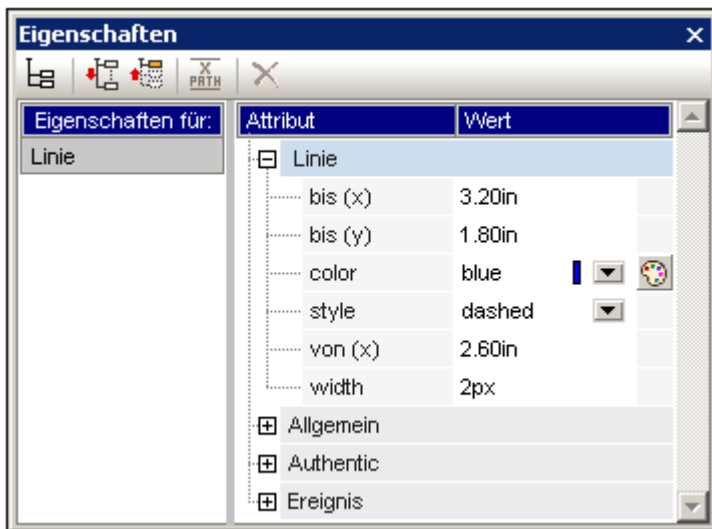
Auswählen, Verschieben, Anpassen der Größe einer Linie

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.

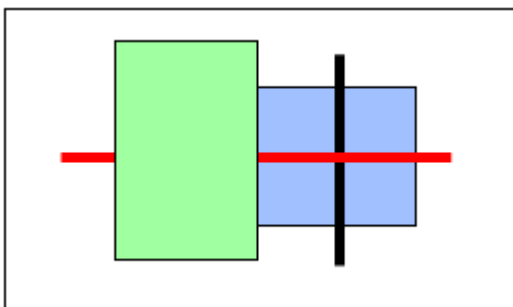


Die folgenden Linieneigenschaften können in der Seitenleiste "Eigenschaften" bearbeitet werden:

- *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.

8.11 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

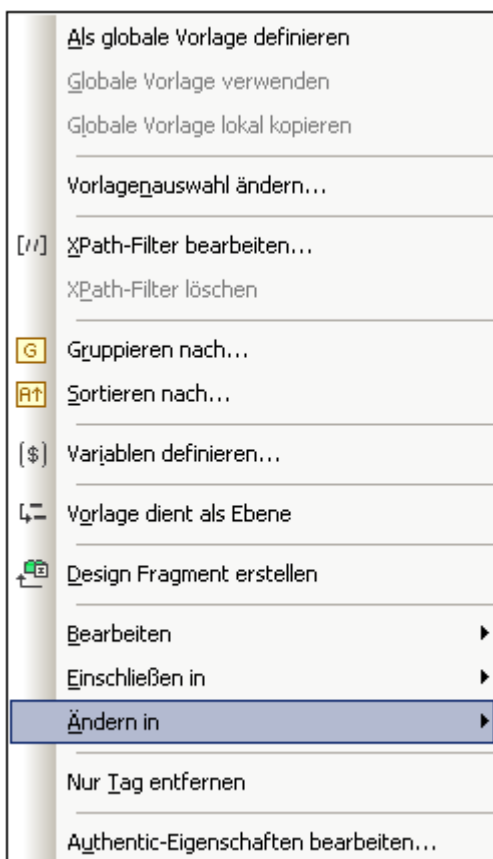
Sie können entweder einen Node oder dessen Inhalt ändern. In der Abbildung auf der linken Seite ist der Node ausgewählt. In der Abbildung auf der rechten Seite ist der Inhalt des Node ausgewählt.



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.

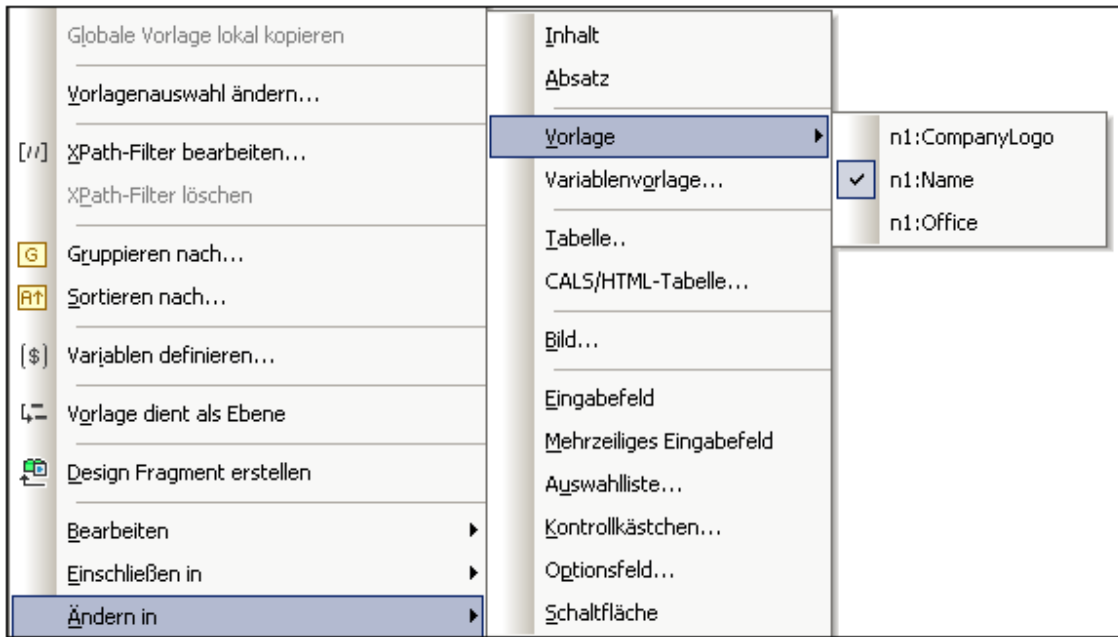
Der Befehl "Ändern in"

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](#) 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*).



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

Kapitel 9

SPS-Datei: Struktur

9 SPS-Datei: Struktur

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

- [Schemaquellen](#)
- [Modulare SPSs](#)
- [Vorlagen und Design-Fragmente](#)

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 `Person` 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](#).

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](#).

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](#).

Dieses Baukastensystem (zusammengesetzt aus mehreren Vorlagen) wird noch durch die StyleVision-Funktion "Design-Fragmente" verbessert. Mit dieser Funktion kann einem wiederverwendbaren Dokument-Fragment eine bestimmte 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](#) näher beschrieben.

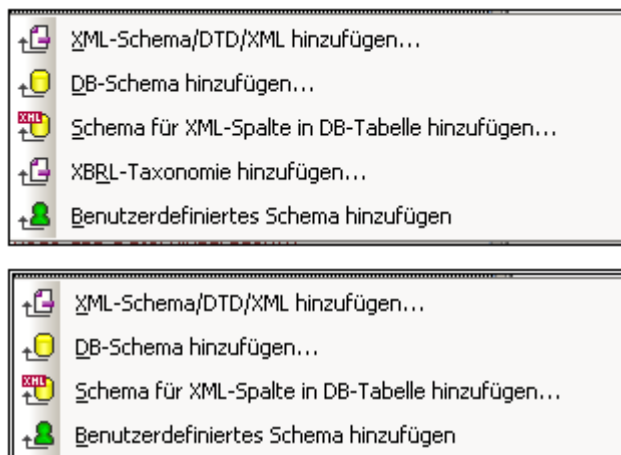
9.1 Schemaquellen

Die Schemaquellen bilden den Ausgangspunkt für das Design und die Designstruktur 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"](#) 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 eine Datenbank oder ein benutzerdefiniertes Schema handeln. Ein wichtiger Punkt ist auch die Frage, ob globale Vorlagen verwendet 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. Bei Auswahl einer Datenbank während des Imports können Sie auswählen, welche Tabellen aus der Datenbank importiert werden sollen. Diese Auswahl bestimmt die Struktur des XML-Schemas, das anhand der Datenbank generiert wird.

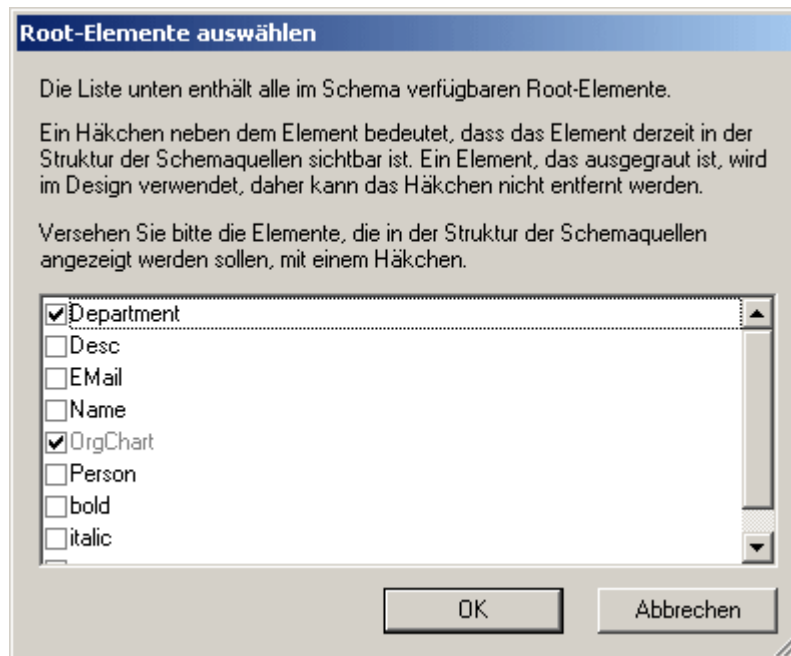
Anmerkung: Wenn Sie einen Namespace zu einem SPS oder einem anhand eines SPS generierten XSLT-Stylesheet hinzufügen möchten, muss der Namespace zum `schema`-Element der obersten Ebene des XML-Schemas, auf dem das SPS basiert, hinzugefügt werden.

Root-Elemente

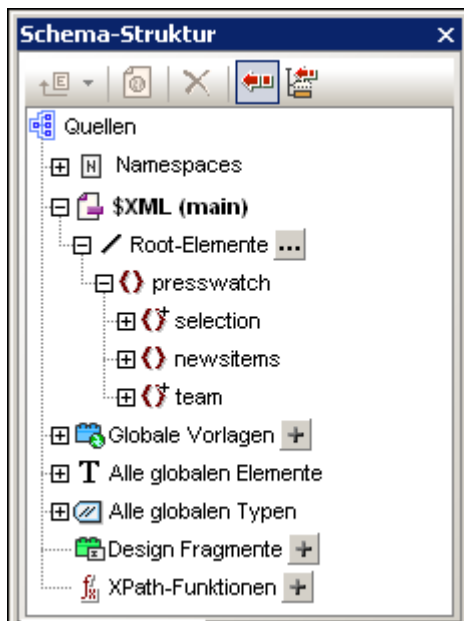
Wenn eine Schemaquelle mehrere [globale Elemente](#) hat, so können mehrere Root-Elemente (

[Dokumentelemente](#)) 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-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](#)) 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](#)) zur Verfügung stehen sollen und klicken Sie auf OK. Die Schemastruktur würde dann in etwa folgendermaßen aussehen:



Sie können nun für das in der Abbildung oben gezeigte SPS zwei Vorlagen erstellen, je eine für jedes der beiden [Dokumentelemente](#). Das SPS kann dann mit verschiedenen XML-Instanzen verwendet werden, die gemäß dem Schema `NanonullOrg.xsd` gültig sind, einige davon haben `Department` als Dokumentelement, andere `OrgChart`. Abhängig vom jeweiligen Instanzdokument wird die entsprechende Vorlage verwendet.

DTDs und XML-Schemas

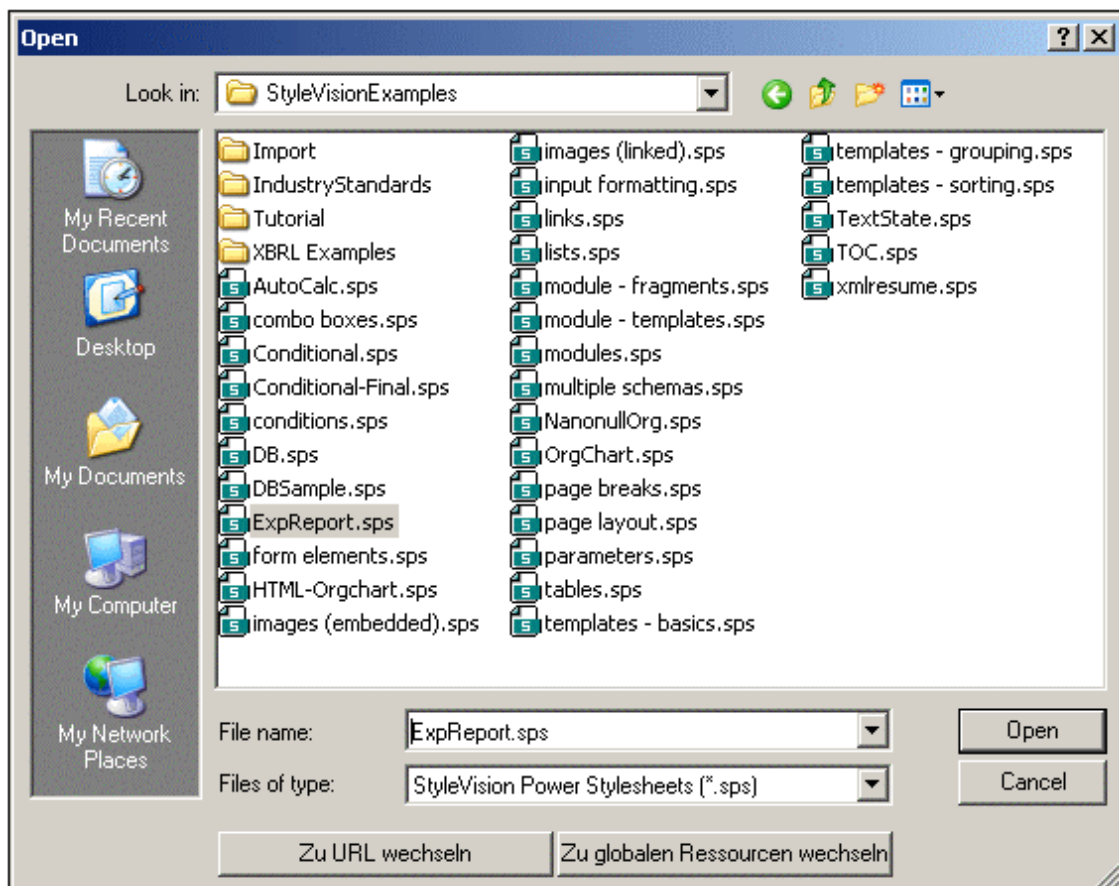
Ein SPS kann auf einem XML-Schema oder einer DTD basieren. Ein XML-Schema oder eine DTD kann auf eine der folgenden Arten als Schemaquelle erstellt werden:

- 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"](#) 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 von Dateien über URLs und globale Ressourcen

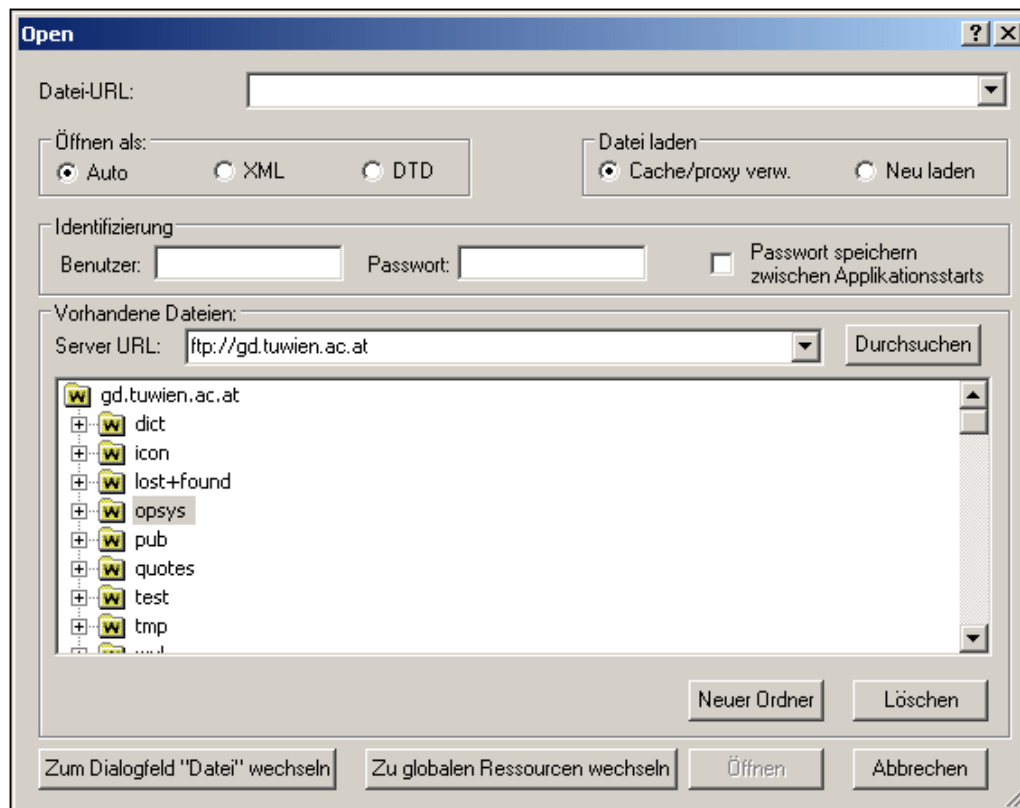
Es gibt mehrere Dialogfelder zum Öffnen und Speichern von Dateien, über die Sie die gewünschte Datei entweder über eine URL oder eine globale Ressource (*siehe Abbildung unten*) auswählen bzw. speichern können. Wählen Sie den Befehl **Zu URL wechseln** oder **Zu globalen Ressourcen wechseln**, um das entsprechende Dialogfeld aufzurufen.



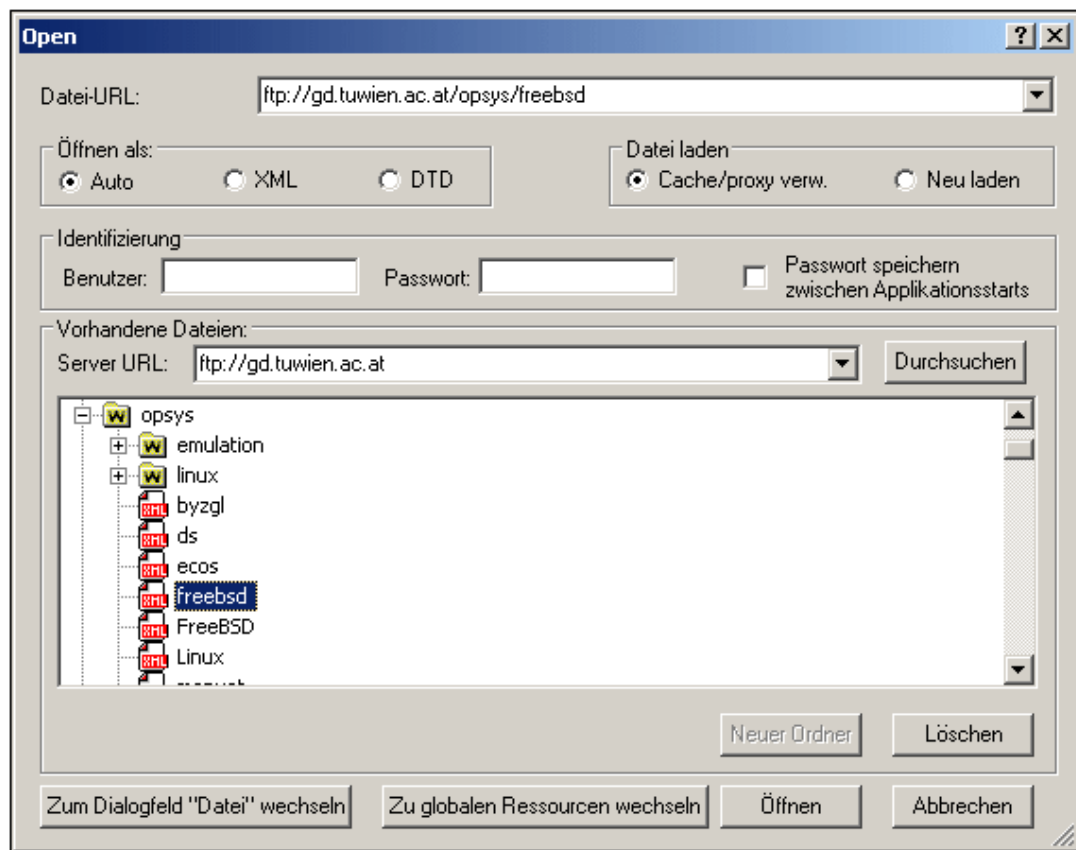
Auswählen von Dateien über URLs

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.



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.

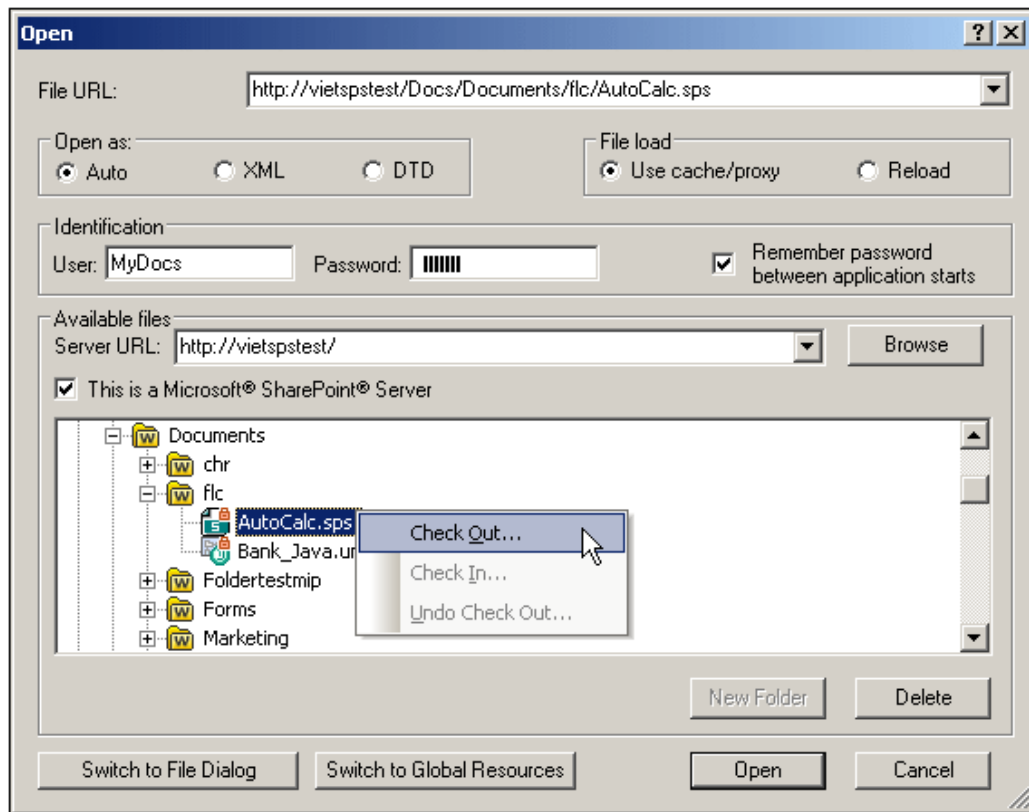
Anmerkung: Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.

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.

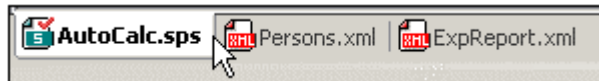


Wenn Sie mit der rechten Maustaste auf eine Datei klicken, erscheint ein Kontextmenü mit Befehlen, die für diese Datei zur Verfügung stehen (*Abbildung oben*).

- 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 auf das Dateiregister im Hauptfenster Ihrer Applikation klicken (*Abbildung unten*), einchecken.



- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht ausgecheckt 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 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.

Öffnen und Speichern von Dateien über globale Ressourcen

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

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.

Datenbankschemas

Ein SPS kann auf einem anhand einer Datenbank generierten Schema basieren. Ein Datenbankschema kann auf eine der folgenden Arten als Schemaquelle erstellt werden:

- Das Datenbankschema wird für ein neues SPS generiert, wenn das SPS direkt von einer Datenbank erstellt wird (**Datei | Neu | Neu von Datenbank** oder **Datei | Neu | Neu von XML-Spalte in IBM DB2**).
- Das Datenbankschema wird (in der [Seitenleiste "Design-Übersicht"](#)), zu einem neuen leeren SPS hinzugefügt, das keine Schemaquellen hat .

Mit dem jeweiligen Befehl zur Generierung eines Datenbankschemas wird anhand der ausgewählten Datenbank ein temporäres XML-Schema generiert. Das Schema wird in der [Seitenleiste "Schema-Struktur"](#) als Schemaquelle erstellt oder der Befehl lädt ein datenbankbasiertes Schema (DB-Schema) aus einer XML-Datenbank. Als Dokumentelement wird ein Element namens `DB` erstellt, das nicht in der Datenbank enthalten ist. Die Datenbankstruktur wird innerhalb dieses Dokumentelements erstellt. Während der Schemaerstellung werden Sie aufgefordert, auszuwählen, welche Tabellen aus der Datenbank importiert werden sollen. Diese Datenbanktabellen werden im XML-Schema als Child-Elemente des `DB` Elements erstellt und auch als Einträge in der Liste der globalen Vorlagen.

Die Erstellung des XML-Schemas anhand der Datenbank erfolgt in zwei Schritten:

- [Herstellen einer Verbindung zu einer Datenbank](#). Dabei wird nach der Datenbankdatei (wenn es sich um eine MS Access Datenbank handelt) gesucht oder ein Connection String (alle anderen Datenbanken mit Ausnahme von MS Access) wird erzeugt.
- [Auswählen der Datenbankdaten](#). In diesem Schritt wird die Tabelle oder Datenstruktur aus der Datenbank ausgewählt. Bei Nicht-XML-Datenbanken basieren das temporäre XML-Schema und die XML-Datei, die erzeugt werden, auf den ausgewählten Datentabellen.

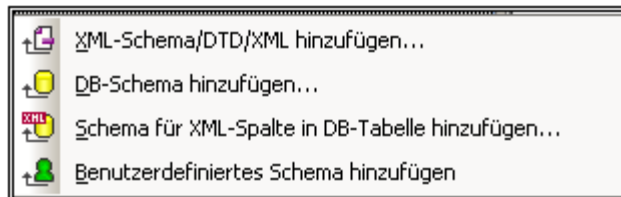
Nachdem die Schemaquelle in Fenster "Schema-Struktur" angezeigt wird, können Sie nun mit dem Entwurf des SPS beginnen. Jedes Mal, wenn Sie auf das Register "Ausgabevorschau" klicken, wird eine temporäre XML-Datei erstellt. Die XML-Datei wird entsprechend dem generierten XML-Schema strukturiert und enthält die Daten aus den ausgewählten Datenbanktabellen.

Benutzerdefinierte Schemas

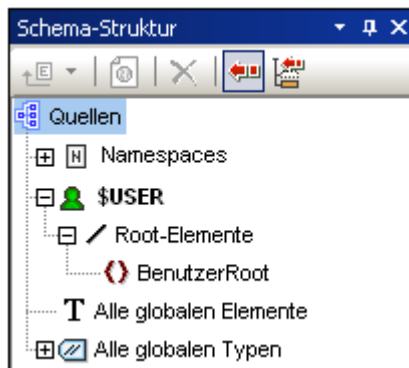
In der [Seitenleiste "Schema-Struktur"](#) können Sie ein benutzerdefiniertes Schema schnell erstellen. Dies ist vor allem dann nützlich, wenn Sie ein XML-Dokument haben, das auf keinem Schema basiert und Sie möchten ein SPS für dieses XML-Dokument erstellen.

Um ein benutzerdefiniertes Schema zu erstellen und in der Seitenleiste "Schema-Struktur" hinzuzufügen, gehen Sie folgendermaßen vor:

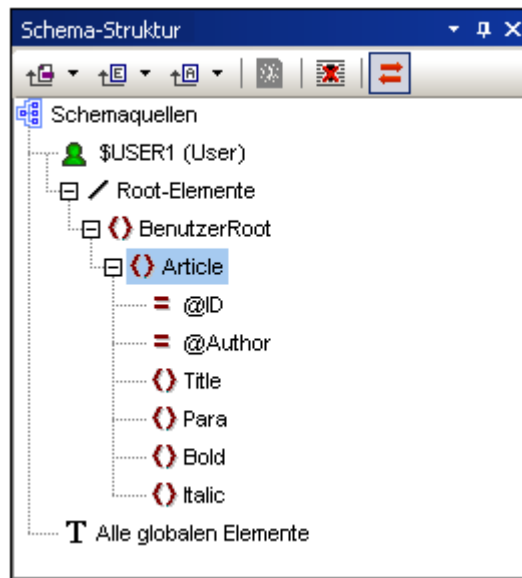
1. Klicken Sie auf **Datei | Neu | Neu (leer)**. Klicken Sie in der [Seitenleiste "Design-Übersicht"](#) (unter der Überschrift "Quellen") auf den Befehl **Neue Quelle hinzufügen** und wählen Sie den Befehl **Benutzerdefiniertes Schema hinzufügen** (*Abbildung unten*).



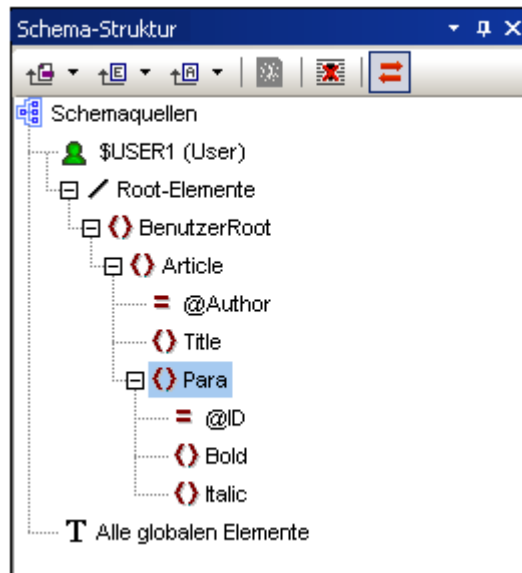
Das neue Schema wird erstellt und erhält die Bezeichnung `$USER` (*Abbildung unten*).



2. In der Baumstruktur des Root-Elements gibt es ein einziges [Root-Element \(Dokumentelement\)](#) mit der Bezeichnung `BenutzerRoot`.
3. Doppelklicken Sie auf `BenutzerRoot` und benennen Sie es um, damit der Name dem [Dokumentelement](#) 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 in der Symbolleiste der [Seitenleiste "Schema-Struktur"](#) auf das Symbol "Neues Element anhängen"  bzw. "Neues Attribut anhängen" . Alternativ dazu können Sie rechtsklicken und den gewünschten Befehl aus dem Kontextmenü auswählen. Nachdem das neue Element oder Attribut zur Struktur hinzugefügt wurde, geben Sie den gewünschten Namen ein. Beachten Sie, dass die Symbole "Neues Element anhängen"  und "Neues Attribut anhängen"  über Dropdown-Menüs verfügen, die durch Rechtsklick auf den Nach unten-Pfeil in dem jeweiligen Symbol aufgerufen werden können. Die Dropdown-Menüs enthalten Einträge, über die Sie relativ zum ausgewählten Node auf verschiedenen Ebenen Nodes hinzufügen können. 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.



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.

9.2 Zusammenführen von XML-Daten aus mehreren Quellen

Wenn als XSLT-Version des SPS die Version XSLT 2.0 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](#) unter Verwendung der `doc()` Funktion in der Ausgabe zusammenführen (siehe *Beispiel unten*).

Beispiel

Der [Ordner "Eigene Dateien"](#), C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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.

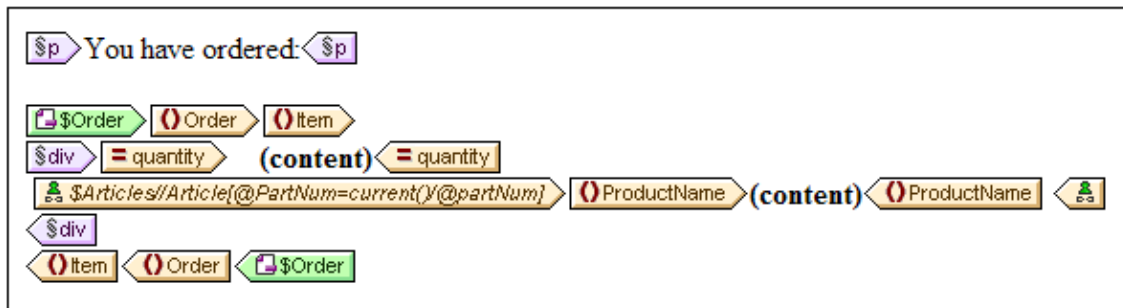
```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeArticles.xsd">
  <Article PartNum="833-AA">
    <ProductName>Lapis necklace</ProductName>
    <Price>99.95</Price>
  </Article>
  <Article PartNum="748-OT">
    <ProductName>Diamond heart</ProductName>
    <Price>248.90</Price>
  </Article>
```

```

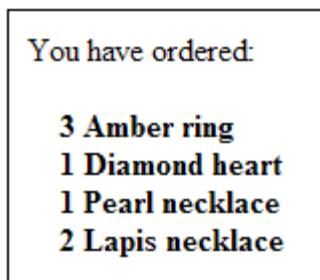
<Article PartNum="783-KL">
  <ProductName>Uncut diamond</ProductName>
  <Price>79.90</Price>
</Article>
<Article PartNum="238-KK">
  <ProductName>Amber ring</ProductName>
  <Price>89.90</Price>
</Article>
<Article PartNum="229-OB">
  <ProductName>Pearl necklace</ProductName>
  <Price>4879.00</Price>
</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](#) 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]

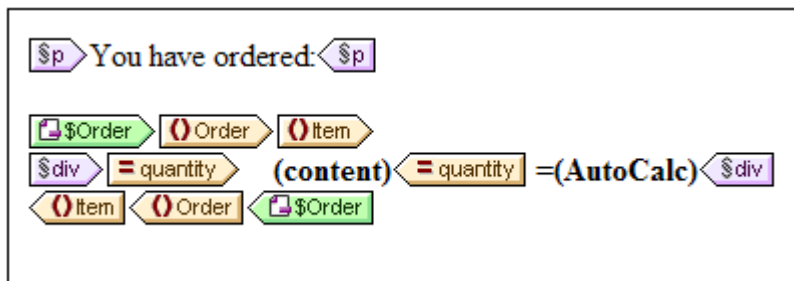


Mit dieser Vorlage wird ein Ausgabedokument erzeugt, das in etwa wie in der Abbildung unten aussieht.



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 Sie außerdem, wie die Vorlage ProductName innerhalb des Kontexts der Vorlage /Articles/Article ausgewählt wird.

Dasselbe Ergebnis wie oben könnte auch mit Hilfe einer [automatischen Berechnung](#) erzielt werden (siehe Abbildung unten).



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).

9.3 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.

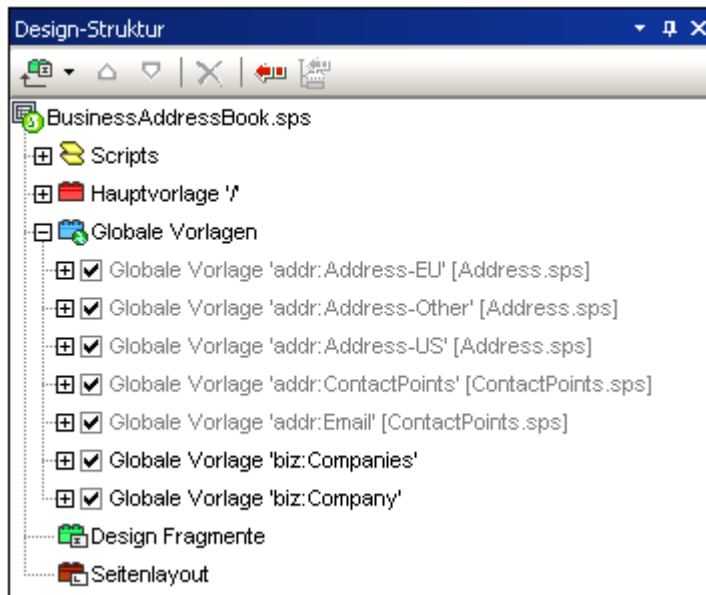
In jedem beliebigen SPS kann ein oder mehrere SPSs als Module hinzugefügt werden. Einige Komponententypen (oder Objekte) in diesen Modulen stehen dann dem importierenden (oder referenzierenden) SPS zur Verfügung.

Verfügbare Modulobjekte

Im Abschnitt [Verfügbare Modulobjekte](#) 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](#) zum Haupt-SPS als Modul hinzu. Alle globalen Vorlagen, Design-Fragmente, JavaScript-Funktionen und Seitenlayout-Elemente 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.



Diese Objekte können anschließend ihren jeweiligen Inkludierungsmechanismen entsprechend im referenzierenden SPS wiederverwendet werden. Globale Vorlagen und Layout-Elemente 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. Und JavaScript-Funktionen werden über das Fenster "Eigenschaften" als Event Handler für die ausgewählte Design-Komponente zugewiesen.

Eine Anleitung zum Erstellen und Arbeiten mit einem modularen SPS finden Sie im Abschnitt [Erstellen eines modularen SPS](#).

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.

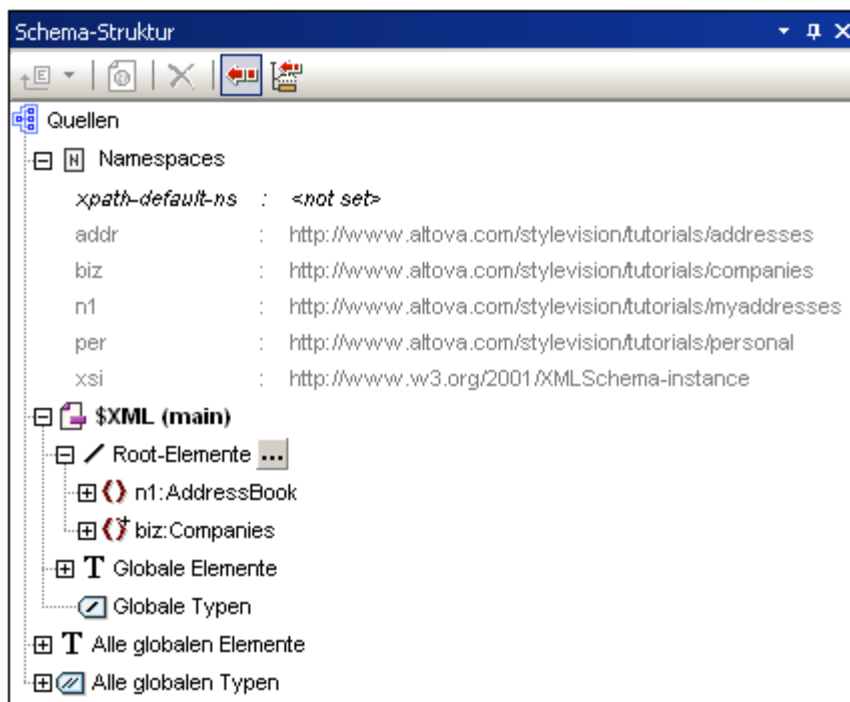
Verfügbare Modulobjekte

In diesem Abschnitt werden die Objekte in [hinzugefügten SPS Modulen](#) aufgelistet, die dem [referenzierenden SPS Modul](#) 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](#). Am Ende dieses 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](#)
- [Globale Vorlagen](#)
- [Design-Fragmente](#)
- [Hinzugefügte Module](#)
- [Scripts](#)
- [CSS-Stile](#)
- [Seitenlayouts](#)
- [Nicht verfügbare modulare Objekte](#)

Namespace-Deklarationen

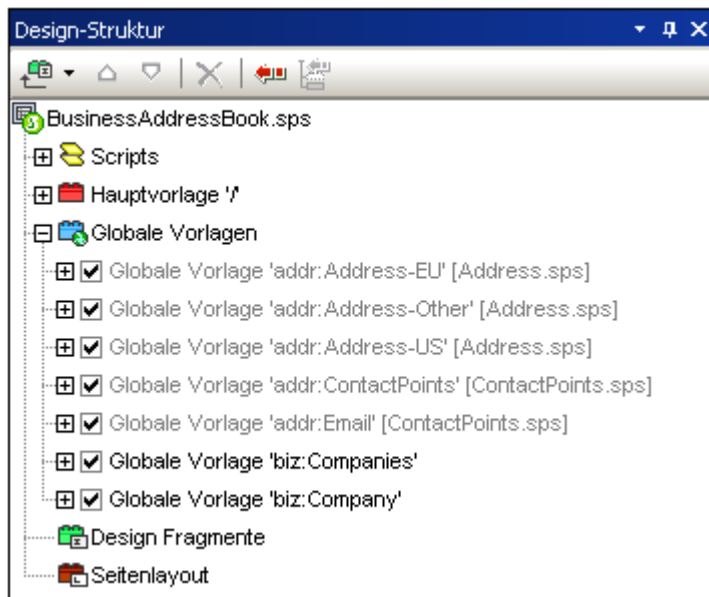
Jedes SPS enthält eine Liste von Namespace URIs und ihren Präfixen. Wenn ein SPS als Modul hinzugefügt wird, werden die Namespaces darin mit den Namespaces in der/den Schemaquelle(n) des referenzierenden SPS verglichen. Wenn eine Namespace URI im hinzugefügten SPS mit einer Namespace URI in der/den Schemaquelle(n) des referenzierenden SPS übereinstimmt, wird das in der Schemaquelle des referenzierenden SPS verwendete Präfix als Präfix für diesen Namespace im referenzierenden SPS übernommen. Wenn für einen Namespace im hinzugefügten SPS keine Übereinstimmung mit einem Namespace 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.

Globale Vorlagen

Die [globalen Vorlagen](#) des hinzugefügten SPS Moduls stehen dem referenzierenden SPS Modul zur Verfügung und werden in der [Seitenleiste "Design-Struktur"](#) angezeigt (*Abbildung unten*).



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](#) im hinzugefügten SPS Modul stehen dem referenzierenden SPS zur Verfügung und werden in der Seitenleiste [Design-Struktur](#) (*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"](#) angezeigt und verwaltet werden.

Scripts

Die Scripts in allen hinzugefügten SPS Modulen stehen im referenzierenden SPS zur Verwendung zur Verfügung und werden in der [Seitenleiste "Design-Struktur"](#) angezeigt. Dabei

werden die Scripts 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"](#) 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"](#) 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 Dokuments 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-Namespace:** Wenn Sie für ein Modul festgelegt wurden, das importiert wird, werden sie nicht im importierenden SPS übernommen.

Erstellen eines modularen SPS

Die Erstellung eines modularen SPS erfolgt grundsätzlich in drei Schritten:

1. Erstellung und Speichern des [hinzuzufügenden SPS Moduls](#).
2. [Hinzufügen des Moduls](#) zum SPS, in dem es verwendet werden soll (d.h. zum referenzierenden SPS Modul).
3. [Aktivieren bzw. Deaktivieren des hinzugefügten Objekts/der hinzugefügten Objekte](#), 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](#) verwendet werden können, muss es sich um [globale Vorlagen](#) handeln. Dies bedeutet, dass die Vorlagen, die wiederverwendet werden sollen, im [hinzuzufügenden SPS Modul](#) 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](#) 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](#), die festlegen, wie Node-Namen erweitert werden.

Sie können die Ausgabe des hinzuzufügenden SPS-Moduls mit Hilfe einer [XML-Arbeitsdatei](#) testen. Die Referenz auf diese XML-Arbeitsdatei wird [vom referenzierenden SPS ignoriert](#).

Hinzufügen des SPS Moduls

Um ein Modul zu einem SPS hinzuzufügen, klicken Sie in der [Design-Übersicht](#) 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 hinzugefügten Moduls zu ändern, wählen Sie es in der Design-Struktur aus und klicken Sie, je nach Wunsch in der Symbolleiste der Design-Übersicht auf die Symbolleisten-Schaltfläche **Nach oben** oder **Nach unten**.

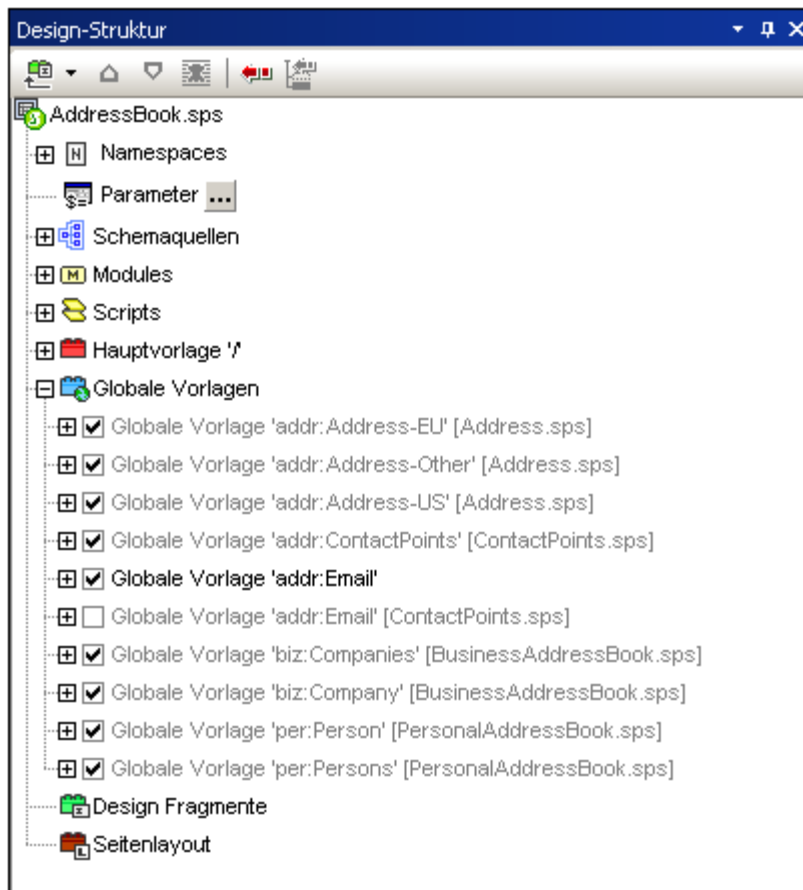
Die Modulreihenfolge spielt bei der Lösung von Konflikten zwischen Scripts, globalen Vorlagen, Design-Fragmenten und Seitenlayouteinträgen keine Rolle.

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 hinzugefügt) werden zum referenzierenden SPS hinzugefügt und unter der entsprechenden Überschrift in der Design-Struktur aufgelistet: Scripts; Globale Vorlagen, Design Fragmente und Seitenlayout. 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 Scripts, 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](#) 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](#).

Globale Vorlagen

Wenn Sie eine [globale Vorlage](#) 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](#).

Scripts

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](#).

Seitenlayoutelemente

Wenn in einem der hinzugefügten Module Seitenlayoutelemente definiert wurden, werden diese unter dem Eintrag "Seitenlayout" des referenzierenden SPS Moduls aufgelistet. Wenn ein Seitenlayoutelement nicht benötigt wird, kann es deaktiviert werden. Wenn es für ein und dasselbe Element z. B. `HeaderOdd` mehrere Optionen gibt, können Sie auswählen, welche der Optionen angewendet werden soll, indem Sie das Kontrollkästchen der entsprechenden Option

aktivieren.

Beispiel: Ein Adressbuch

Der [Ordner \(Eigene\) Dateien](#) C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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.

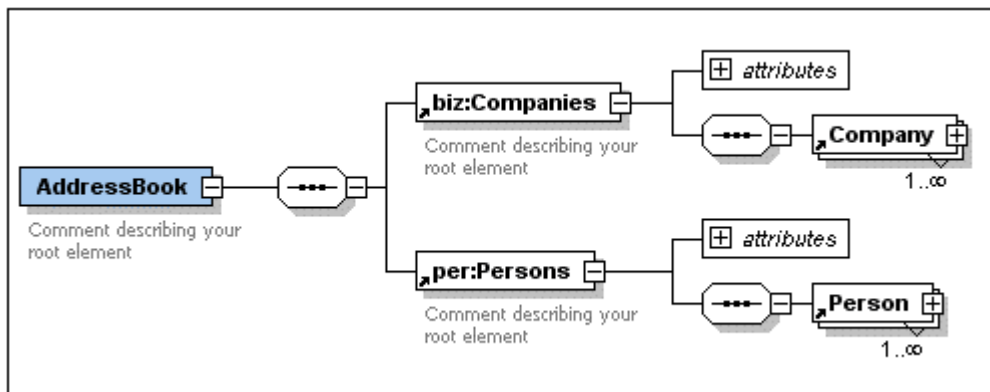
- Das komplette Adressbuch besteht aus zwei Modulen: (i) einem Geschäftsadressbuch und (ii) einem persönlichen Adressbuch. Jedes davon hat ein separates SPS, in dem unterschiedliche Designs definiert sind. Die beiden Module können zu einem einzigen Adressbuch kombiniert werden. In diesem Fall dient die 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:

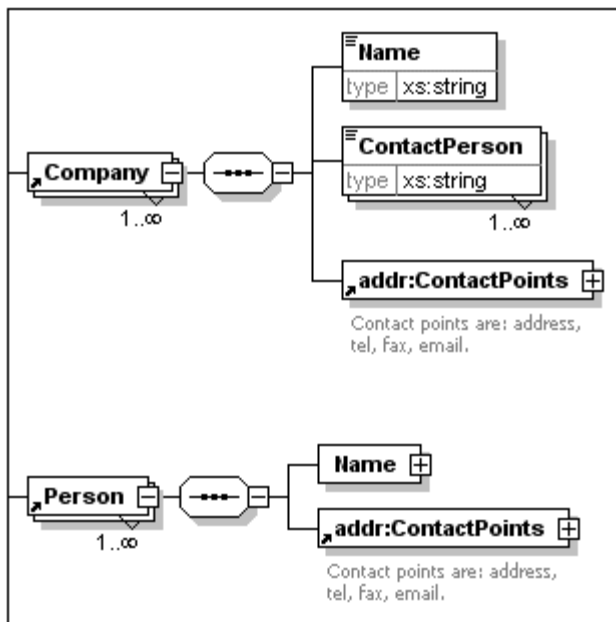
- [die Schema-Dateien](#)
- [die XML-Datenquellen](#)
- [die SPS-Dateien](#)

Die Schema-Dateien

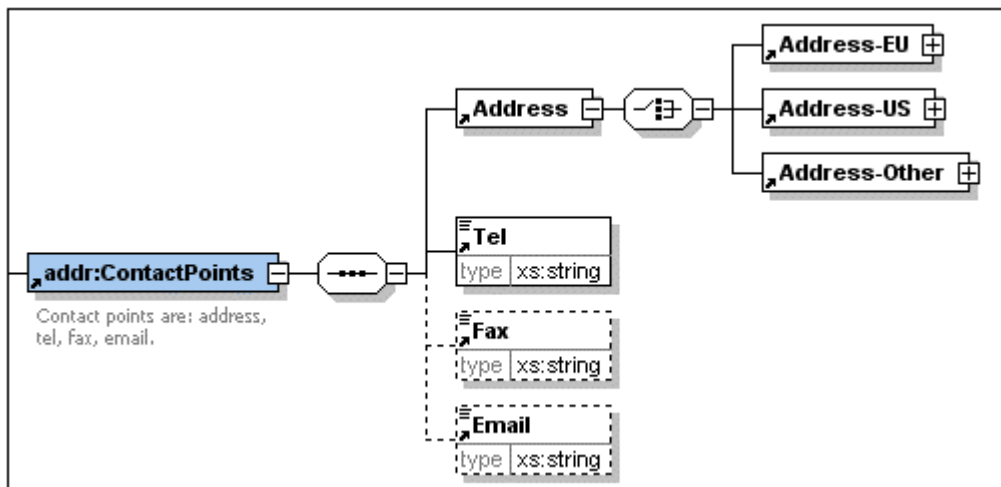
Bei der Erstellung von Schemas für modulare SPSs muss unbedingt beachtet werden, dass die Elemente, die wiederverwendet werden sollen, als globale Elemente erstellt werden. Das Schema für das Adressbuch ist `AddressBook.xsd`. Dieses Schema wurde durch Import der Schemas für das Geschäftsadressbuch (`BusinessAddressBook.xsd`) 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.



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/WSDL-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](#) 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.

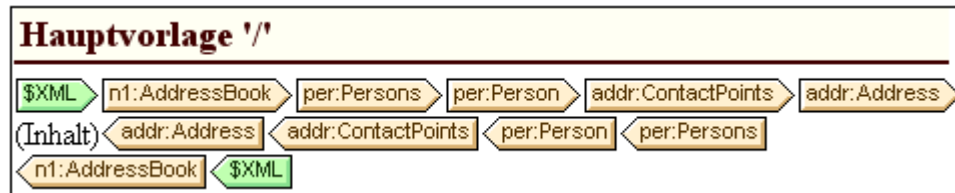
Address.sps

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.

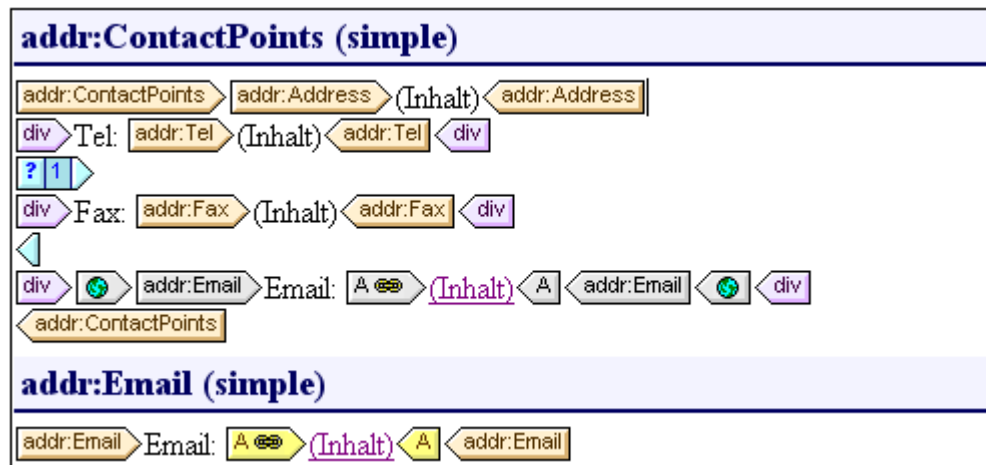


Da nur das Element `Address` verarbeitet wird, wird in den Ausgabevorschauen nur die Ausgabe von `Address` angezeigt. Wenn als Modul `Address.sps` verwendet wird, werden die globalen Vorlagen hinzugefügt und die Hauptvorlage wird ignoriert.

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 importiert 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`.
- `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 `PersonalAddressBook.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.

9.4 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](#) und optional dazu einer oder mehreren [globalen Vorlagen](#). 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](#) 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](#) im Design beschäftigen, sondern nur mit der Struktur. In diesem Abschnitt werden auch zusätzliche Strukturelemente für die Dokumentpaginierung wie z.B. Umschlagseiten, Kopf- und Fußzeilen nicht behandelt. Diese Themen werden im Abschnitt [Erstellen der Druckausgabe](#) abgehandelt.

Hauptvorlage

Die [Hauptvorlage](#) legt die Struktur des Ausgabedokuments fest. D.h. die Reihenfolge des Layouts in der die Hauptvorlage entspricht der Reihenfolge des Layouts in der Authentic-Ansicht und 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](#) 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](#) referenzieren. In diesen Fällen wird die Verarbeitung 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](#) des Schemas. StyleVision bietet auch die Option, mehrere Root-Elemente ([Dokument-Elemente](#)) auszuwählen. Das bedeutet, dass es innerhalb der Hauptvorlage für jedes der aktiven Dokument-Elemente eine [lokale Vorlage](#) 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.

Globale Vorlagen

Eine [globale Vorlage](#) kann für jeden Node oder Typen im Schema oder für einen in einem XPath-Ausdruck definierten Node angelegt werden.

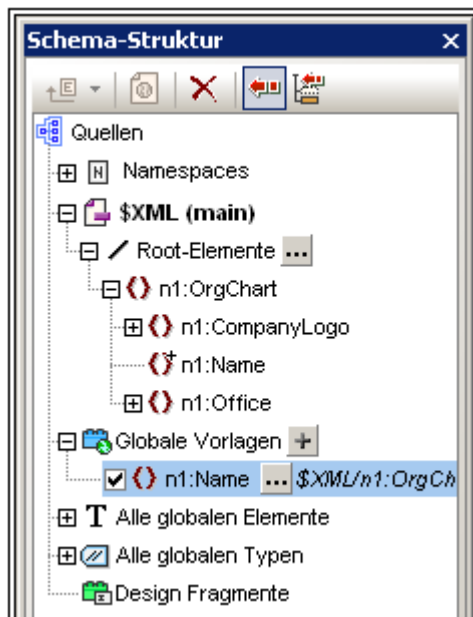
Eine globale Vorlage definiert Anweisungen für den ausgewählten Node oder Typ und wird über die [Hauptvorlage](#), [Design-Fragmente](#) oder andere globale Vorlagen aufgerufen. Das Verarbeitungsmodell ähnelt dem deklarativer Programmiersprachen, d.h. es wird eine einzige Vorlage definiert und mehrmals 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](#) 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](#)) oder ([restlicher Inhalt](#)) innerhalb eines Elements oder Typs in eine [lokale Vorlage](#) eingefügt wird und der restliche Inhalt dieses Elements oder Typs einen Node oder Typ enthält, für den es eine [globale Vorlage](#) 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](#) 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](#)) oder ([restlicher Inhalt](#)), 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.

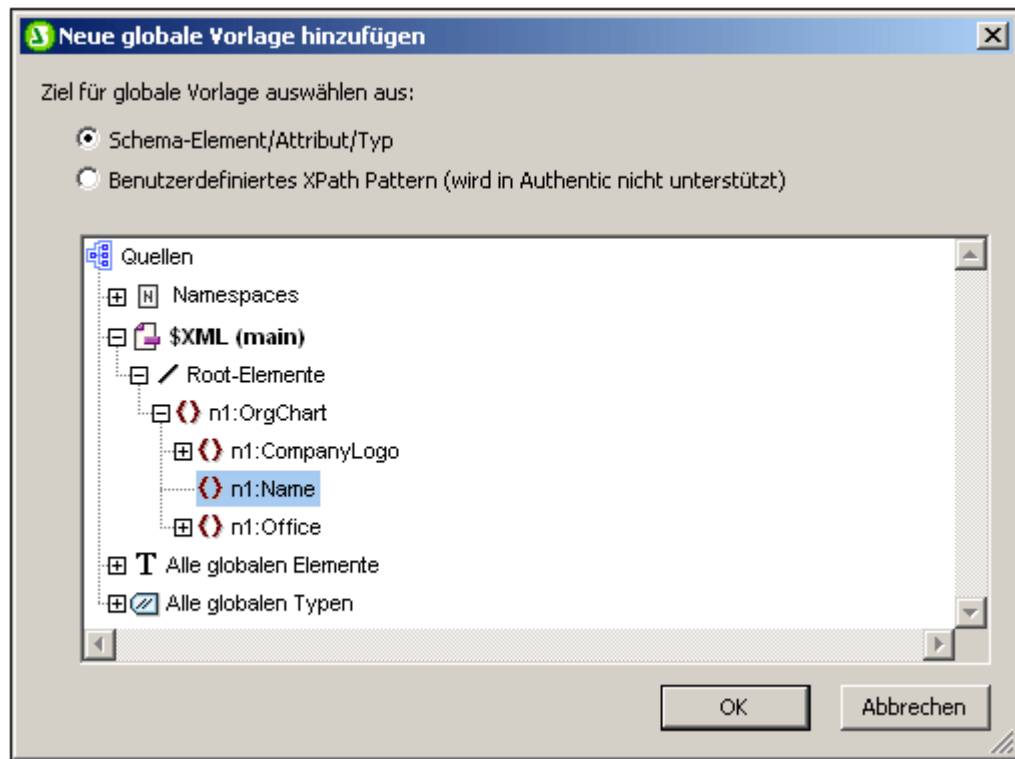
Erstellen einer globalen Vorlage

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. (Beachten Sie, dass globale Vorlagen, die für Nodes erstellt wurden, die durch einen XPath-Ausdruck ausgewählt wurden, in Authentic nicht unterstützt werden.) 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.



- 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 **Neue globale Vorlage hinzufügen** aus. Daraufhin wird das oben beschriebene Dialogfeld "Neue globale Vorlage hinzufügen" angezeigt..
- 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).

Verwenden einer globalen Vorlage

Nachdem eine globale Vorlage erstellt wurde, kann sie verwendet werden, wenn ein Node mit demselben qualifizierten Namen (durch Ziehen mit der Maus) in das Dokument eingefügt wird. 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](#). Daraufhin wird das [Dialogfeld "XPath-Filter bearbeiten"](#) 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**) 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 einer 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.

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.

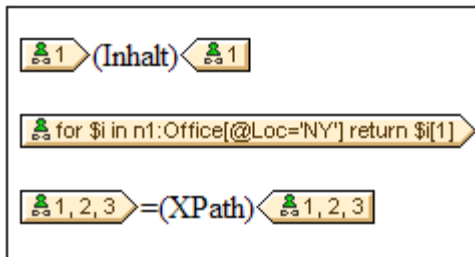
Beachten Sie bitte: Eine einfache globale Vorlage wird automatisch in eine komplexe globale Vorlage konvertiert, wenn **rund um** den Element-Node, für den die globale Vorlage erstellt wurde, ein [vordefiniertes Format](#) oder eine neue Zeile erstellt wird. Dies führt dazu, dass die Verarbeitung für komplexe globale Vorlagen (im vorhergehenden Listeneintrag beschrieben) durchgeführt wird. Um wieder eine einfache globale Vorlage zu verwenden, muss das [vordefinierte Format](#) 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](#)). Um die automatische Konvertierung von einfachen globalen Vorlagen in komplexe globale Vorlagen zu vermeiden, stellen Sie sicher, dass das [vordefinierte Format](#) 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](#), 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](#).

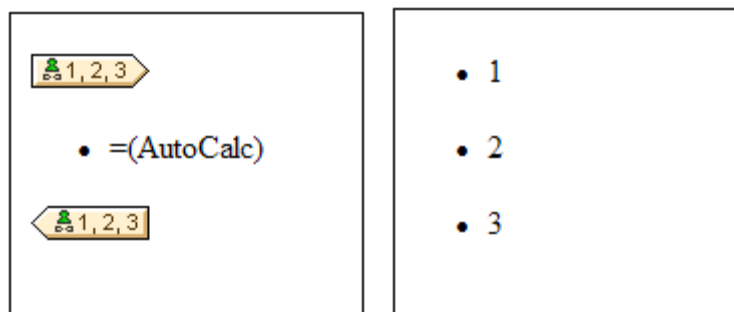
Benutzerdefinierte Vorlagen

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 auf der linken Seite der Tags. Benutzerdefinierte Vorlagen sind sehr nützlich, da sie beim Erstellen von Vorlagen besonders viel Flexibilität bieten. Beachten Sie, dass Inhalt, der durch benutzerdefinierte Vorlagen generiert wurde, in der **Authentic-Ansicht nicht bearbeitet werden kann**.



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, sodass diese in der Authentic-Ansicht bearbeitet werden kann. 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 definierte Bedingung erfüllt. Solche Vorlagen 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 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 weiter oben.

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	<pre><xsl:for-each select="Org"> <xsl:for-each select="Office"> <xsl:for-each select="Dept"> ... </xsl:for-each> </xsl:for-each> </xsl:for-each></pre>	Jedes <code>Office</code> Element hat seine eigene <code>Dept</code> Population. Dadurch kann die Gruppierung und Sortierung innerhalb der einzelnen <code>Office</code> Elemente durchgeführt werden.
Ja	<pre><xsl:for-each select="/Org/Office/Dept"> ... </xsl:for-each></pre>	Die <code>Dept</code> Population umfasst alle <code>Office</code> Elements und alle <code>Org</code> 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](#), 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

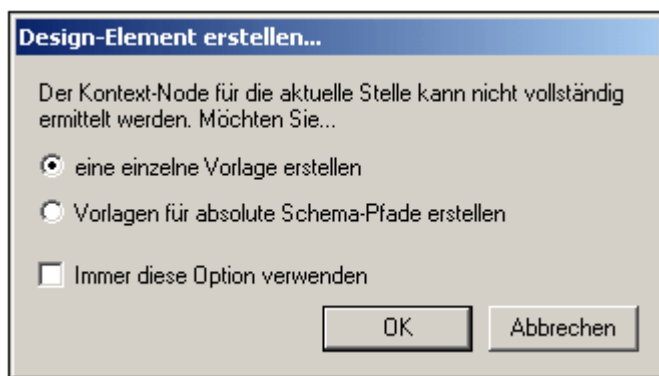
verbundenen und getrennten Nodes) dasselbe, es sei denn ein Gruppierungs- oder Sortierkriterium wurde definiert. Nähere Informationen darüber, wie sich verbundene Nodes auf die Gruppierung und Sortierung auswirken, finden Sie unter [Node-Vorlagen-Operationen](#).

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

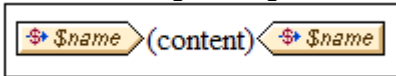
Wenn ein Node aus der Schemastruktur zu einer benutzerdefinierten Vorlage hinzugefügt wird, ist der Kontext für den neuen Node unbekannt, wenn die benutzerdefinierte Vorlage für einen Node oder eine Sequenz erstellt wurde, der/die nicht in den Kontext der Schemaquelle des SPS platziert werden kann. Sie 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 Dialogfeld [Extras | Optionen](#) ändern.

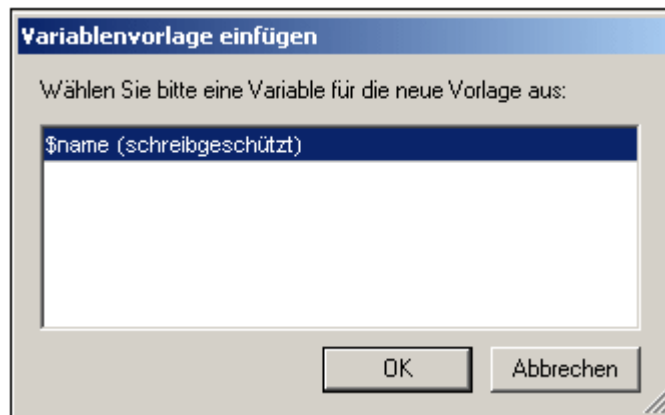
Variablenvorlagen

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.



Um eine Variablenvorlage einzufügen, gehen Sie folgendermaßen vor:

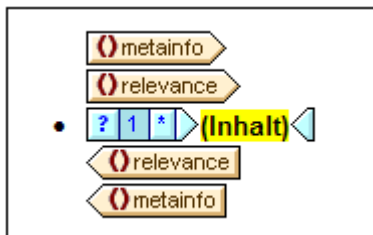
1. Platzieren Sie den Cursor an die Stelle im Design, an der die Vorlage eingefügt werden soll.
2. Klicken Sie auf den Befehl **Einfügen | Variablenvorlage**. Daraufhin wird das Dialogfeld "Variablenvorlage einfügen" angezeigt (*Abbildung unten*).



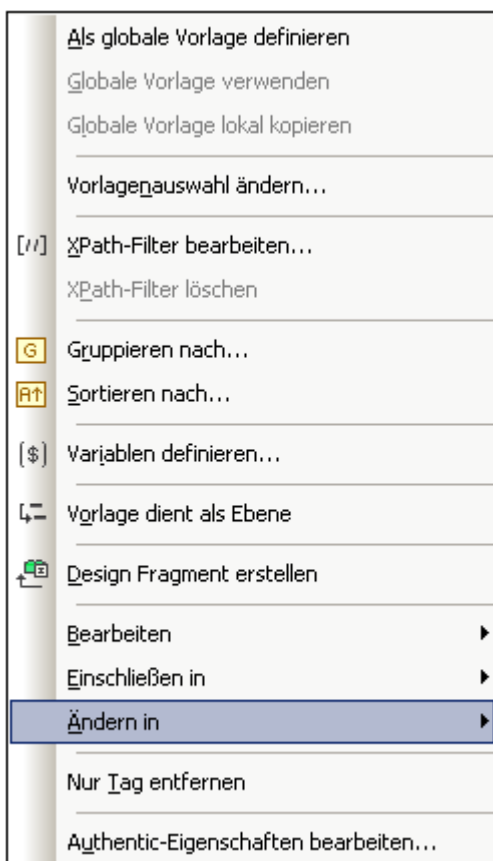
3. Das Dialogfeld enthält eine Liste aller im SPS definierten [benutzerdefinierten Parameter und Variablen](#). Wählen Sie die Variable aus, für die Sie eine Variablenvorlage hinzufügen möchten.
4. Klicken Sie zum Fertigstellen auf **OK**.

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. In der Abbildung unten sehen Sie zwei Node-Vorlagen: `metainfo` und `relevance`.



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 (*siehe Abbildung unten*).



Die Befehle im Kontextmenü werden unten beschrieben:

- [Globale Vorlagen](#)
- [Vorlagenauswahl](#)
- [XPath-Filter](#)
- [Gruppieren nach, Sortieren nach, Variablen definieren, Vorlage dient als Ebene](#)
- [Design Fragment erstellen](#)
- [Nur Tag entfernen](#)

- [Bearbeiten, Einschließen in, Ändern in, Authentic-Eigenschaften](#)

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](#).

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"](#) 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](#) 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](#) 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.

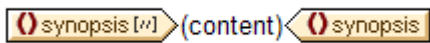
Ein Filter im Design kann zu jeder beliebigen Node-Vorlage und zu mehreren Node-Vorlagen hinzugefügt werden. Auf diese Art können Sie eine Auswahl treffen, die XPath-Ausdrücken wie z.B.: `/Personnel/Office[@country='US']/Person[Title='Manager']` entspricht, um alle Manager in den amerikanischen Büros 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.

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](#) 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](#) 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](#).
- 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](#) 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, Authentic-Eigenschaften bearbeiten

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](#), [Spezieller Absatz](#), [Nummerierung und Aufzählungszeichen](#), [Hyperlink](#), [Bedingung](#), [Inhaltsverz.-Markierung und Ebene](#).
- *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"](#).
- *Authentic-Eigenschaften bearbeiten*: Dem instantiierten Node können in der Authentic-Ansicht bestimmte Eigenschaften zugewiesen werden, die in der Seitenleiste "Eigenschaften", die bei Auswahl dieses Befehls angezeigt wird, bearbeitet werden können. Authentic-Eigenschaften werden im Abschnitt [Authentic Node-Eigenschaften](#) näher beschrieben.

Design-Fragmente

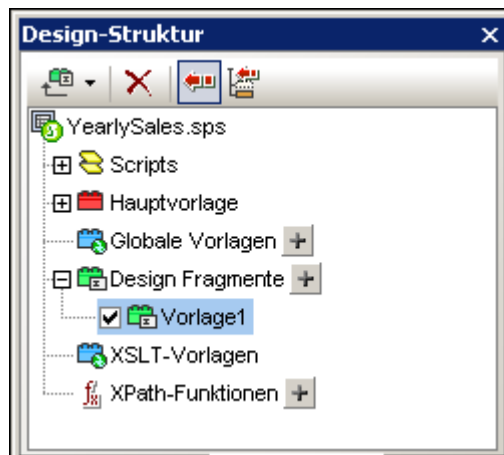
Design-Fragmente dienen zum Erstellen von Komponenten, die an verschiedenen Stellen im Dokument wiederverwendet werden können, ähnlich wie Funktionen, die wiederverwendet werden können. Die Vorgangsweise ist die folgende:

1. [Erstellen Sie das Design-Fragment im Design.](#)
2. [Erstellen Sie den Inhalt des Design-Fragments.](#)
3. [Fügen Sie das Design-Fragment an der gewünschten Stelle in einer Vorlage ein](#)

Erstellen eines Design-Fragments

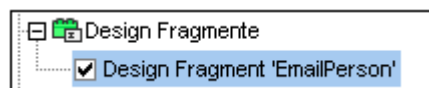
So erstellen Sie ein Design-Fragment:

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. (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](#)-Symbolleiste von StyleVision auf die entsprechenden Schaltflächen zum [Ein-/Ausblenden](#).) 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.

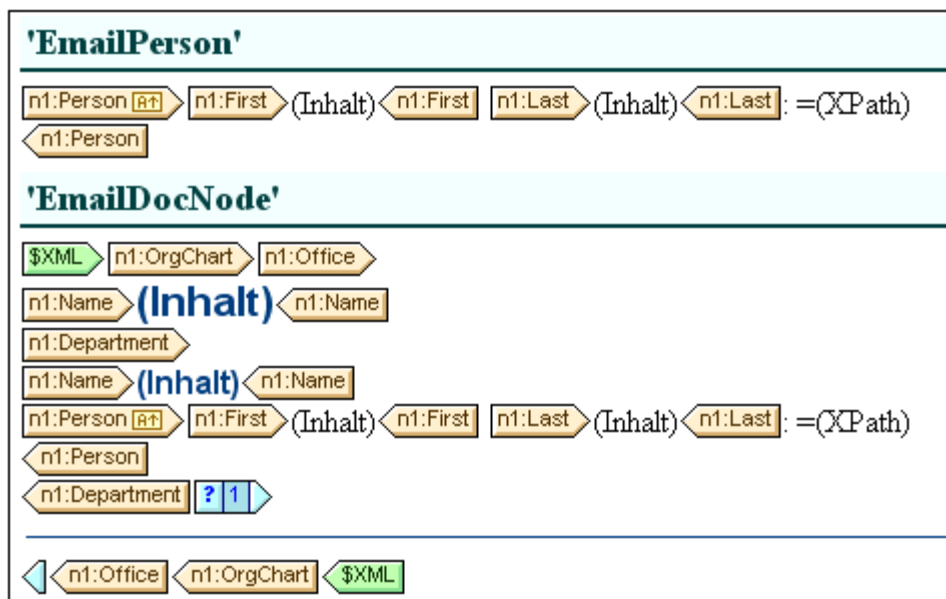
Anmerkung: Wenn Sie ein Design-Fragment anhand einer bestehenden Vorlage erstellen, klicken Sie mit der rechten Maustaste auf die Vorlage und wählen Sie im Kontextmenü den Befehl **Design Fragment erstellen**. Dadurch erstellen Sie anhand der ausgewählten Vorlage an dieser Stelle im Design eine Design-Fragment-Vorlage. 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](#).

Erstellen des Inhalts eines Design-Fragments

Erstellen Sie im Design den Inhalt des Design-Fragments wie bei [jeder globalen Vorlage](#). 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, dass 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](#) `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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](#).

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](#).

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 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 Symboleistenschaltfläche "Entfernen" .

Design-Fragmente in modularen SPSs

Wenn ein [SPS Modul zu einem anderen SPS Modul hinzugefügt wird](#), 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](#).

Beispieldatei

Ein Beispiel-SPS finden Sie im [Ordner \(Eigene\) Dateien](#), C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorials\DesignFragments.

9.5 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.

Anmerkung:

- Importierte XSLT-Vorlagen können in StyleVision nicht geändert werden.
- XSLT-Vorlagen werden in der Authentic-Ansicht nicht unterstützt.

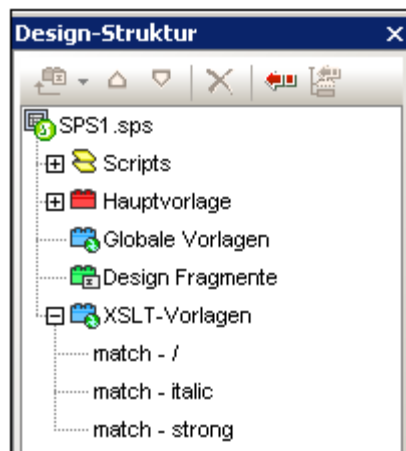
Importieren der XSLT-Datei

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** und anschließend auf den Befehl **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 `xml: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.



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-

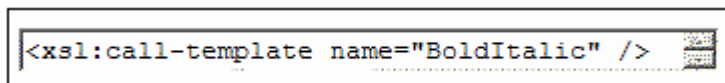
Vorlagen bzw. (ii) dem Wert des `name` Attributs von benannten Vorlagen aufgelistet.

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 importierte Datei (die im XSLT-Code zuerst aufgelistet ist), die niedrigste Priorität, gefolgt von der als zweites importierten Datei, usw.

Benannte Vorlagen

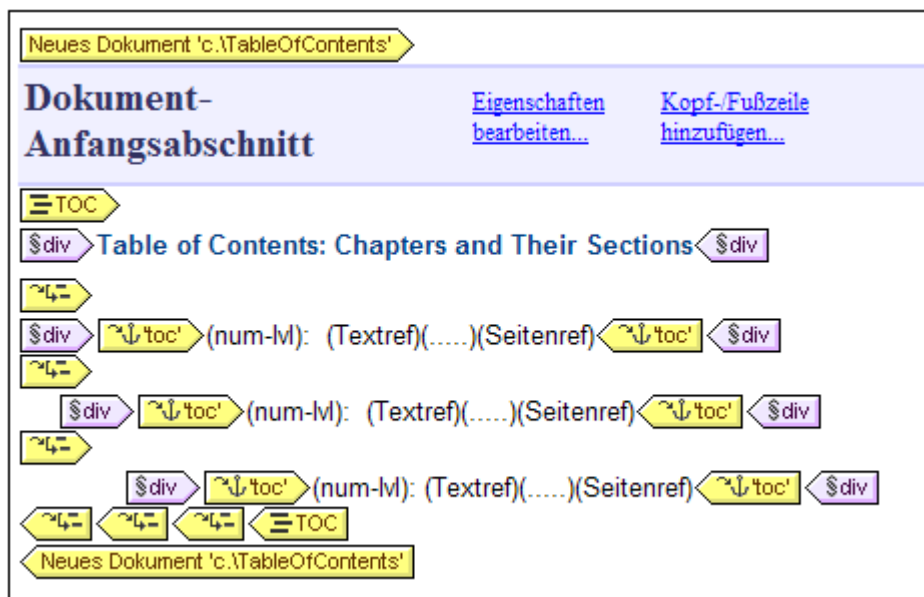
Eine benannte Vorlage kann aus der Design-Struktur an jede Stelle im Design gezogen werden. Sie wird an dieser Stelle als `xsl:call-template` Element (*Abbildung unten*) erstellt, das die benannte Vorlage aufruft.



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.

9.6 Mehrfach-Dokumentausrage

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.



M

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](#).

Die Funktion zur Ausgabe mehrerer Dokument steht sowohl im HTML- als auch im RTF-Format zur Verfügung. In der Authentic-Ansicht, die als Bearbeitungsansicht konzipiert ist, wird der Inhalt in einem einzigen Dokument angezeigt.

Diese Beschreibung mehrerer Ausgabedokumente ist in die folgenden Unterabschnitte gegliedert:

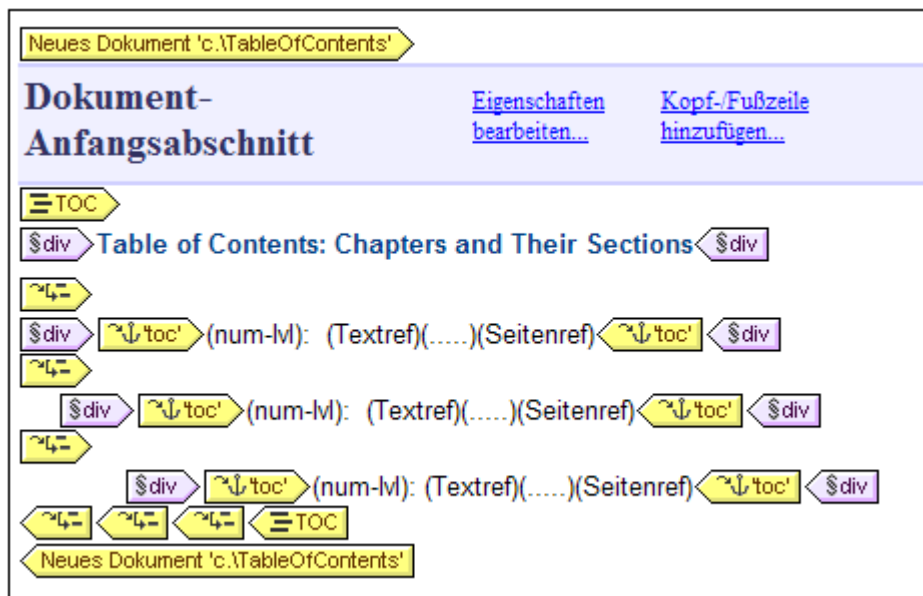
- [Einfügen einer Vorlage für ein neues Dokument](#)
- [Vorlagen für neue Dokumente und Design-Struktur](#)
- [URLs von Vorlagen für neue Dokumente](#)
- [Vorschau und Ausgabedokumentdateien](#)
- [Dokumenteigenschaften und Stile](#)

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](#).
2. Die Vorlage für das neue Dokument enthält einen Dokument-Anfangsabschnitt. Weitere zusätzliche Dokumentabschnitte können, wie im Abschnitt [Dokumentabschnitte](#) beschrieben, zum Dokument-Anfangsabschnitt hinzugefügt werden.

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](#) als auch die [Struktur des Eingabedokuments \(XML-Quelldokument\)](#) 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:

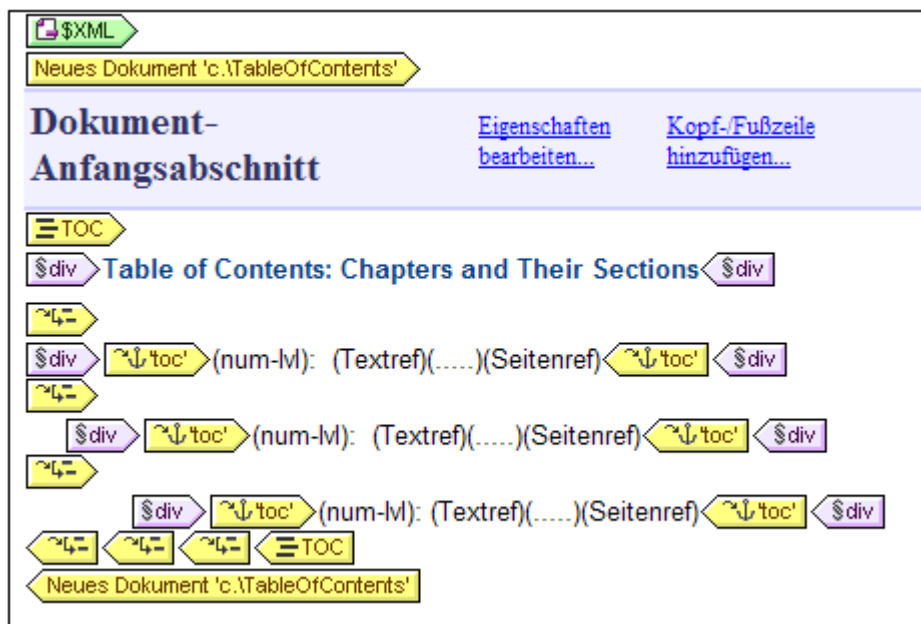
1. Der Node, in dem die Vorlage für das neue Dokument angelegt wird, wird nur einmal verarbeitet. In diesem Fall wird auch die Vorlage für das neue Dokument nur einmal verarbeitet. Als Name in der URL-Eigenschaft der neuen Dokumentvorlage kann daher ein statischer Name verwendet werden.
2. Der Node, 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.

URLs von Vorlagen für neue Dokumente

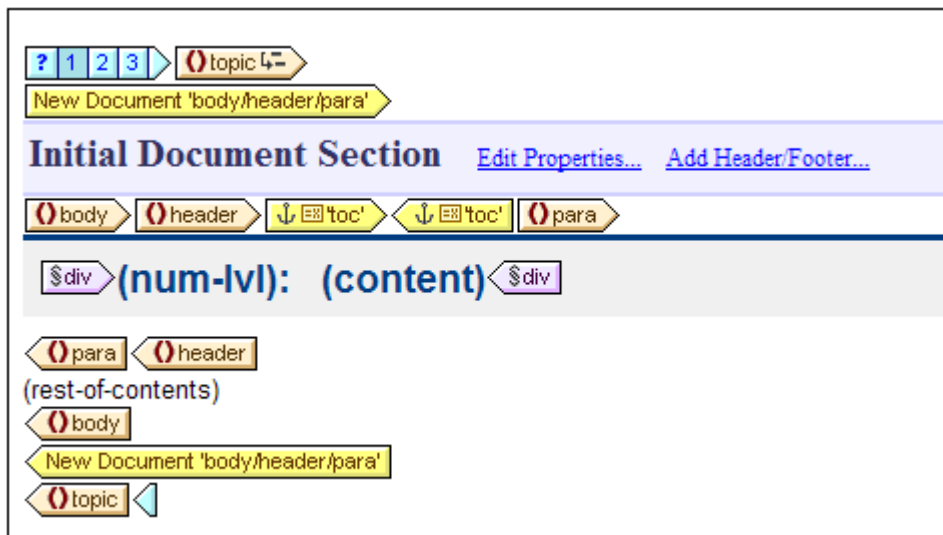
In diesem Abschnitt wird erläutert, wie die [URLs von Vorlagen für neue Dokumente mit der Design-Struktur im Zusammenhang stehen](#), wie [URLs bearbeitet werden](#) und [wie mehrere Ausgabedokumente durch Links miteinander verknüpft werden können](#).

URLs von Vorlagen für neue Dokumente

Wenn die Vorlage für das neue Dokument nur einmal verarbeitet wird (siehe [vorhergehender Abschnitt](#)), 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 Dokuments (\$XML) befindet, wird sie nur einmal verarbeitet. Für die URL wurde ein statischer `TableOfContents`-Wert definiert. Dieser Wert wird daher 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](#)). Wenn die URL alternativ dazu einen Pfad 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](#)), 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 die Seitenleiste "Eigenschaften" (Abbildung unten) aufgerufen, wo Sie das Wertfeld der Eigenschaft *URL ohne Dateierweiterung* bearbeiten können.



Wenn Sie eine statische URL eingeben möchten, bearbeiten Sie das Wertfeld, damit es den gewünschten URL-Text enthält. Wenn Sie eine dynamische URL eingeben möchten, klicken Sie in das Wertfeld, klicken Sie in der Symbolleiste der Seitenleiste "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:\MyOutput\filename` generiert. Die entsprechende Dateierweiterung wird automatisch für das jeweilige Ausgabeformat generiert.

Verknüpfen der Dokuments mittels Links

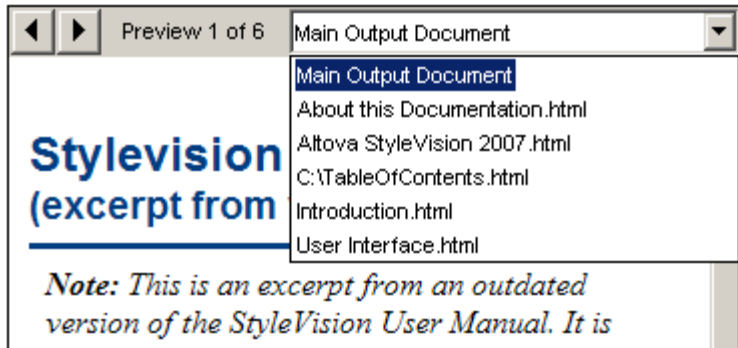
Mehrere Ausgabedokumente können mit Hilfe von [Lesezeichen und Hyperlinks](#) miteinander verknüpft werden. Ein [Lesezeichen](#) kann am Anfang der Vorlage für das neue Dokument oder an einer beliebigen Stelle in der Vorlage für das neue Dokument platziert werden. Anschließend

können in anderen Dokumenten [Hyperlinks](#) 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](#) 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.

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*).

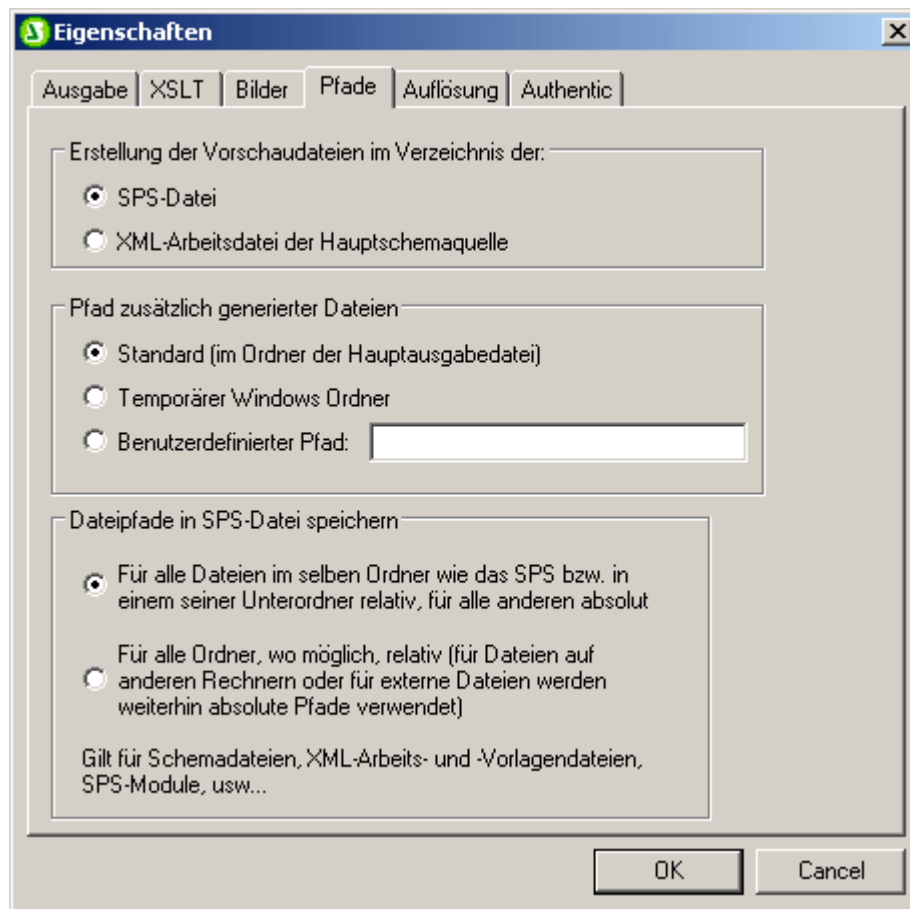


In der Abbildung oben sehen Sie die HTML-Vorschau eines SPS-Dokuments, das für die Generierung mehrerer Ausgabedokumente angelegt wurde. Die einzelnen Ausgabedokumente können auf folgende Weise im Vorschaufenster aufgerufen werden: (i) durch Navigation durch die verfügbaren Dokumente mit Hilfe der Pfeiltaste 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:

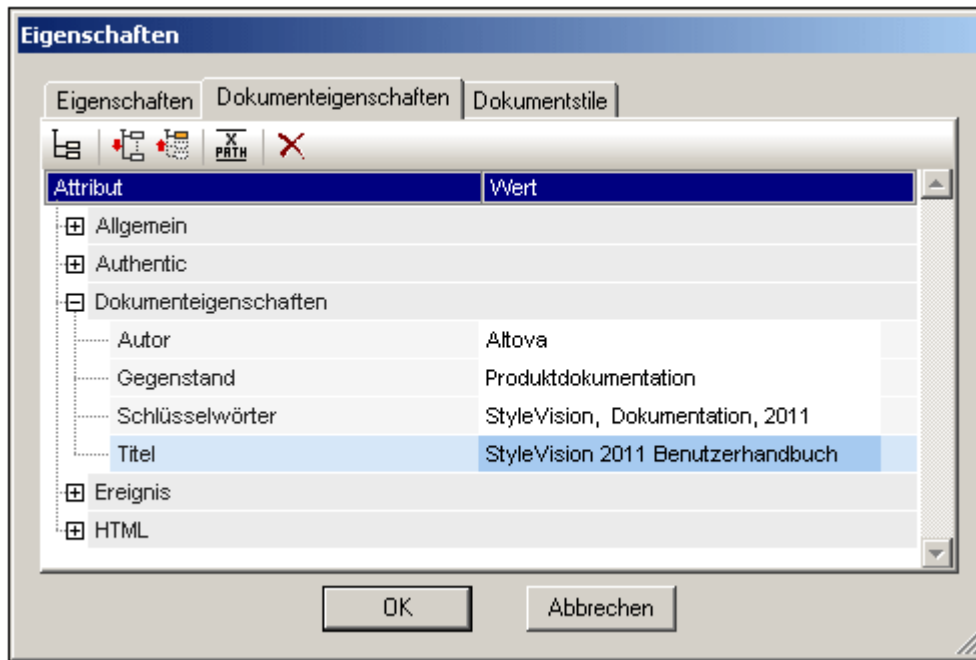
1. Platzieren Sie die Maus über den Menübefehl **Datei | Generierte Dateien speichern** und klicken Sie auf das gewünschte Ausgabeformat.
2. Navigieren Sie im Dialogfeld "Generierte Dateien speichern", das nun angezeigt wird, zu dem Ordner, in dem die generierte Datei gespeichert werden soll.
3. Geben Sie den Namen der Hauptdokumentdatei ein und klicken Sie auf **Speichern**.

Der Speicherort aller Ausgabedokumentdateien sowie aller anderen zusätzlich generierten Dateien wie Bilddateien und Diagrammbilddateien wird zu Ihrer Information in einem Pop-up-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.

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](#).

Kapitel 10

SPS-Datei: Komplexe Funktionen

10 SPS-Datei: Komplexe Funktionen

In den Abschnitten [Inhalt der SPS-Datei](#) und [Struktur der SPS-Datei](#) 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](#). 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](#). 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](#). Die Verarbeitung kann für eine Gruppe von Elementen, die mit einem XPath-Ausdruck ausgewählt werden, definiert werden.
- [Sortieren](#). Eine Gruppe von XML-Elementen kann nach mehreren Sortierschlüsseln sortiert werden.
- [Parameter und Variablen](#). 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](#). 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](#) (sich wiederholende Nodes im Dokument können automatisch nummeriert werden und die Nummern werden formatiert; (ii) [Textreferenzen](#) (Text im Dokument kann für die Referenzierung markiert werden und anschließend von einer anderen Stelle im Dokument referenziert werden); und (iii) [Lesezeichen und Hyperlinks](#) (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.

10.1 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:

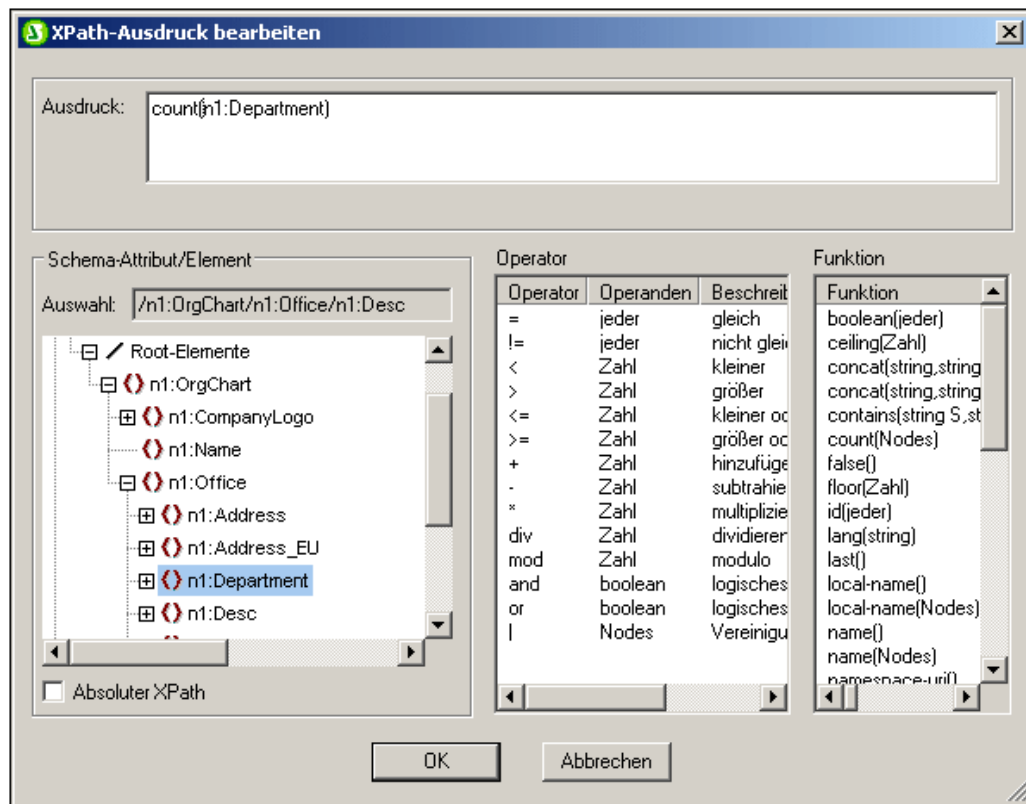
- um Berechnungen mit Operationen an dynamischen Datenwerten einzufügen. So können Sie z.B. die Anzahl der `Employee` Elemente in einem `Office` Element zählen (mit `count( Employee )`) oder die Werte aller `Price` Elemente in den einzelnen `Invoice` Elementen summieren (mit `sum( Price )`) oder die 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.
- zum Aktualisieren des Werts von Nodes im XML-Hauptdokument. So könnte z.B. der Node `Adresse` mit einem XPath-Ausdruck wie `concat( Title, ' ', FirstName, ' ', LastName )` aktualisiert werden.
- um den Inhalt eines Node an jeder beliebigen Stelle im Design anzuzeigen.

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 "XPath-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 Seitenleiste-bereichen des Dialogfelds zusammenstellen. Beachten Sie, in welchem Kontext-Node die Berechnung eingefügt wird; der Kontext-Node wird in der Schemaquellenstruktur markiert, wenn das Dialogfeld angezeigt wird. (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 ausgewählt haben, müssen Sie XPath 2.0-Ausdrücke verwenden.)
3. Wenn Sie optional dazu den Wert der automatischen Berechnung in einen Node im XML-Dokument kopieren möchten, können Sie diesen Node über einen XPath-Ausdruck auswählen. Eine Beschreibung zum Aktualisieren von Nodes mit dem Ergebnis der automatischen Berechnung finden Sie im Abschnitt [Aktualisieren von](#)

[Nodes mit automatischen Berechnungen.](#)

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 Authentic-Ansicht oder eine Ausgabeansicht.

Java- und .NET-Funktionen

Java- und .NET-Funktionen können in den XPath-Ausdrücken von automatischen Berechnungen verwendet werden. Hier sehen Sie Beispiele für XPath-Ausdrücke, die Java- und .NET-Funktionen enthalten::

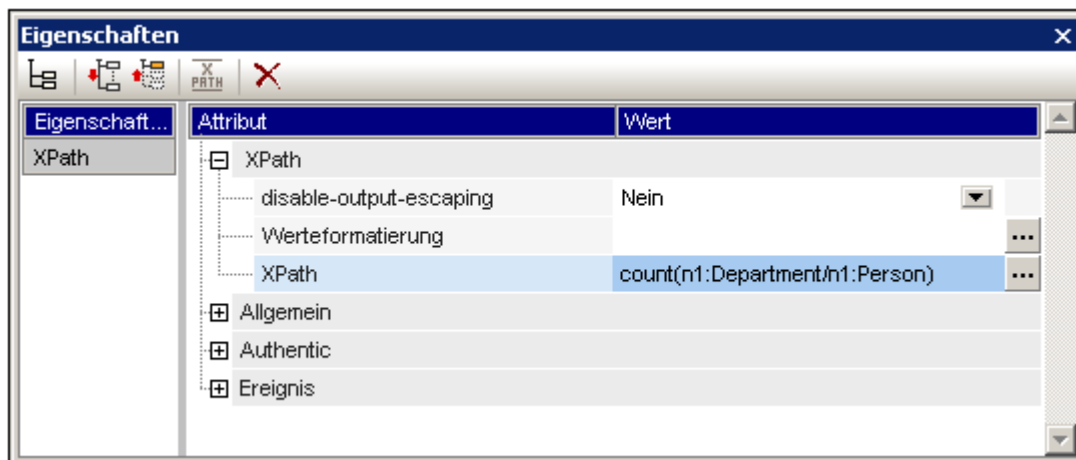
- **Java:** `java:java.util.Date.toString( java:java.util.Date.new() )`
- **.NET:** `clitype:System.Math.PI()`

Nähere Informationen zu Erweiterungen finden Sie unter [Anhänge | XSLT-Prozessorinformationen | Erweiterungen](#).

Anmerkung: Java- und .NET-Funktionen in automatischen Funktionen werden nur in der Enterprise Edition von Authentic Desktop und Authentic Browser unterstützt

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" (Abbildung unten) angezeigt, in dem Sie den XPath-Ausdruck bearbeiten können.



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](#) 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 Sonderabsatz typ

pre eingeschlossen wird, wird in der Ausgabe (mit Ausnahme der RTF-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

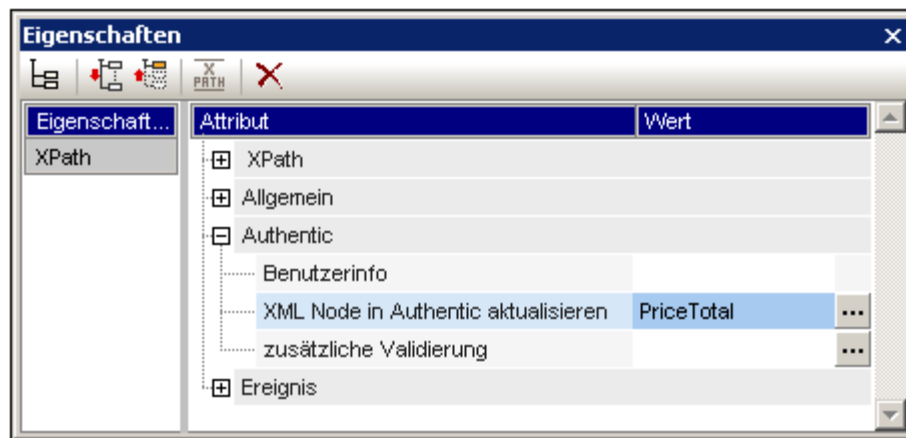
Beachten Sie die folgenden Punkte:

- 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.
- In der Authentic-Ansicht wird eine automatische Berechnung jedesmal, wenn ein für die Berechnung relevanter Wert geändert wird (d.h. jeder Node, der im XPath-Ausdruck inkludiert wird), neu ausgewertet.
- Das Ergebnis einer automatischen Berechnung kann in der Authentic-Ansicht oder einer anderen Ausgabeansicht nicht bearbeitet werden.
- Jeder Node im XML-Dokument kann mit dem Ergebnis der automatischen Berechnung aktualisiert werden.

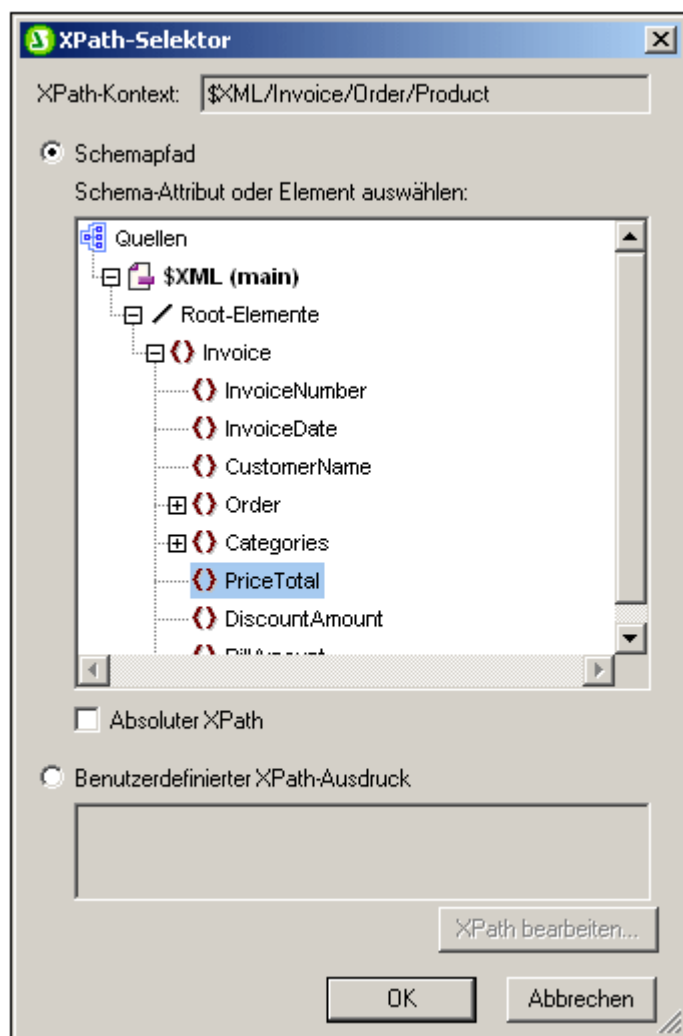
Aktualisieren von Nodes mit automatischen Berechnungen

Sie können den Wert (oder das Ergebnis) einer automatischen Berechnung in einen Node im XML-Hauptdokument (also dem Dokument, das in der Authentic-Ansicht bearbeitet wird) kopieren. Gehen Sie dazu folgendermaßen vor:

1. Erstellen Sie die automatische Berechnung wie im Abschnitt [Bearbeiten und Verschieben von automatischen Berechnungen](#) beschrieben.
2. Wählen Sie die automatische Berechnung aus und klicken Sie in der Seitenleiste "Eigenschaften" in der Eigenschaftsgruppe *Authentic* in der Eigenschaft `XML-Node in Authentic aktualisieren` auf die **Bearbeiten**-Schaltfläche (*Abbildung unten*).



3. Wählen Sie im Dialogfeld "XPath-Selektor" (*Abbildung unten*) den zu aktualisierenden Node aus und klicken Sie zum Fertigstellen auf **OK**. Wenn der XPath-Ausdruck absolut sein soll (d.h. bei der Dokument Root beginnen soll), aktivieren Sie das Kontrollkästchen "Absoluter XPath".



Alternativ dazu können Sie, wenn Sie das Kontrollkästchen "Benutzerdefinierter XPath-Ausdruck" auswählen, jeden beliebigen XPath-Ausdruck eingeben, um den Node auszuwählen, der aktualisiert werden soll.

Mit dem XPath-Ausdruck muss ein **einzelner Node** ausgewählt werden. Bei Auswahl mehrerer Nodes wird kein Node aktualisiert.

WICHTIG!

Damit ein Node im XML-Hauptdokument aktualisiert wird, müssen zwei Bedingungen erfüllt werden:

1. Im XPath-Ausdruck für die automatische Berechnung muss mindestens ein Wert enthalten sein, der mit einem XML Node in Zusammenhang steht, d.h. es muss sich um einen dynamischen Wert handeln. Z.B. `Price*1.2`. Dieser Ausdruck enthält das Element `Price` und ist daher dynamisch. Wenn der XPath-Ausdruck einen statischen Wert enthält (z.B. `string("Nanonull, Inc.")`), funktioniert die Funktion "XML-Node aktualisieren" nicht.
2. Alle im XPath-Ausdruck verwendeten Nodes müssen in der Authentic-Ansicht geändert werden. Wenn also der XPath-Ausdruck `Price*1.2` lautet und die automatische Berechnung erstellt wurde, um den Node `VATPrice` zu aktualisieren, dann muss das Element `Price` in der Authentic-Ansicht geändert werden, damit der Node `VATPrice`

aktualisiert wird.

Ändern des zu aktualisierenden Node und Abbruch der Aktualisierung

Um den zu aktualisierenden Node zu ändern, klicken Sie (im Fenster "Eigenschaften") unter "XPath" in der Eigenschaftsgruppe *Authentic* in der Eigenschaft `XML Node aktualisieren` auf die Schaltfläche **Bearbeiten** und wählen Sie anschließend im Dialogfeld "Schema-Selektor", das nun angezeigt wird, den gewünschten Node aus. Um die Eigenschaft `XML Node aktualisieren` zu löschen, klicken Sie in der Symbolleiste des Fensters "Eigenschaften" auf die Schaltfläche "Entfernen".

Soll für die Anzeige die automatische Berechnung oder der Inhalt des aktualisierten Node verwendet werden?

Wenn keine Bedingungsvorlagen verwendet wurden, können Sie für die Anzeige sowohl die automatische Berechnung als auch den Inhalt des aktualisierten Node verwenden. Es spielt keine Rolle, welche Option Sie auswählen, da der Node direkt nach der Auswertung der automatischen Berechnung aktualisiert wird und die Aktualisierung durch keine anderen Faktoren kompliziert wird. Beachten Sie allerdings, dass die Quelle des angezeigten Inhalts in diesen beiden Fällen eine unterschiedliche ist.

Ausblenden der automatischen Berechnung

In manchen Fällen möchten Sie vielleicht für eine Berechnung zur Aktualisierung eines Node eine automatische Berechnung verwenden. In diesem Fall kommt es zu einem der folgenden Szenarios:

- Sie möchten das Ergebnis nur einmal anzeigen. Sie können die automatische Berechnung nicht ausblenden, da sie sonst nicht ausgewertet würde. Wenn keine Bedingungsvorlage verwendet wird, sollte am besten die automatische Berechnung selbst und nicht der Inhalt des aktualisierten Node angezeigt werden.
- Sie möchten das Ergebnis nur anzeigen, um den Node zu aktualisieren. Formatieren Sie die automatische Berechnung in diesem Fall so, dass sie im Ausgabemedium unsichtbar ist (z.B. indem Sie eine automatische Berechnung vor einem weißen Hintergrund weiß formatieren).

[Beispiel: Eine Rechnung](#): Ausführliches Beispiel zur Verwendung von automatischen Berechnungen.

[Dialogfeld "XPath"](#): enthält eine Beschreibung des Dialogfelds "XPath-Ausdruck bearbeiten".

Automatische Berechnungen auf Basis von aktualisierten Nodes

Bei der Erstellung einer automatischen Berechnung (zweite automatische Berechnung), bei der ein Node verwendet wird, der von einer anderen automatischen Berechnung aktualisiert wird (erste automatische Berechnung), gibt es zwei mögliche Situationen:

- Die beiden automatischen Berechnungen befinden sich in derselben Vorlage.
- Die beiden automatischen Berechnungen befinden sich in unterschiedlichen Vorlagen.

Automatische Berechnungen in derselben Vorlage

Wenn sich die beiden automatischen Berechnungen in derselben Vorlage befinden, wird im SPS das folgende Verfahren angewendet:

1. Ein Node, der im XPath-Ausdruck der ersten automatischen Berechnung verwendet wird, wird geändert.
2. Alle Node-Werte im XML-Dokument werden gelesen und alle automatischen Berechnungen werden ausgeführt.
3. Nachdem die erste automatische Berechnung korrekt ausgeführt wurde, wird der angegebene XML Node (nennen wir ihn `Node-A`) aktualisiert. Die zweite automatische Berechnung, welche auf `Node-A` basiert, wird ausgeführt, verwendet aber den Wert von `Node-A`, bevor `Node-A` aktualisiert wurde. Dies liegt daran, dass der Wert von `Node-A` gelesen wurde, bevor der Node aktualisiert wurde, und seither nicht mehr gelesen wurde.
4. Wenn das Dokument nun auf irgendeine Weise bearbeitet wird oder wenn die Dokumentansichten geändert werden (Wechsel von und in die Authentic-Ansicht), so werden die Werte der Nodes erneut ausgelesen und die automatischen Berechnungen werden ausgeführt.
5. Nun wird die zweite automatische Berechnung ausgeführt. (Wenn mit dieser automatischen Berechnung ein Node aktualisiert werden soll, so muss ein Node, der in einem XPath-Ausdruck verwendet wird, wie dies bei Node-Aktualisierungen üblich ist, geändert werden, bevor die Aktualisierung durchgeführt wird.)

Die Zeitverzögerung zwischen der Aktualisierung von `Node-A` und der Auswertung der zweiten automatischen Berechnung mit dem aktualisierten Wert von `Node-A` könnte für den Benutzer der Authentic-Ansicht verwirrend sein. Um sicherzustellen, dass es nicht dazu kommt, sollte der XPath-Ausdruck der zweiten automatischen Berechnung am besten den XPath-Ausdruck der ersten automatischen Berechnung - und nicht den aktualisierten Node selbst - enthalten. Infolgedessen wird die zweite automatische Berechnung mit dem Input aus der ersten automatischen Berechnung ausgeführt. D.h. die erste automatische Berechnung wird im Rahmen der zweiten ausgeführt, sodass die zweite Berechnung unabhängig davon ausgeführt werden kann, was in `Node-A` enthalten ist.

Beispiel

In der ersten automatischen Berechnung wird die Mehrwertsteuer für ein Produkt berechnet: Verwendet werden dazu die Nodes für den (i) Nettopreis und den (ii) Mehrwertsteuersatz; der Node für den Mehrwertsteuerbetrag wird aktualisiert; in der zweiten automatischen Berechnung wird der Bruttopreis berechnet. Der Bruttopreis ist die Summe von Nettopreis und Mehrwertsteuerbetrag; der Node für den Bruttopreis wird aktualisiert.

- Die automatische Berechnung für den Mehrwertsteuerbetrag ist: $\text{NetPrice} * \text{VATRate} \div 100$. Wenn der Mehrwertsteuersatz (`VATRate`) für das Produkt eingegeben wird, wird die automatische Berechnung ausgeführt und der Node `VATAmount` wird aktualisiert.
- Wenn die automatische Berechnung zur Ermittlung des Bruttopreises $\text{NetPrice} + \text{VATAmount}$ lautet, wird die automatische Berechnung mit dem Wert von `VATAmount` ausgeführt, der eingelesen wurde, bevor `VATAmount` aktualisiert wurde.

- Wenn die automatische Berechnung zur Ermittlung des Bruttopreises jedoch $\text{NetPrice} + (\text{NetPrice} * \text{VATRate} \div 100)$ lautet, wird die automatische Berechnung mit dem Wert von `VATRate` ausgeführt und der Node `GrossPrice` wird aktualisiert. Der aktualisierte Node `VatAmount` wurde in der zweiten automatischen Berechnung ausgelassen.

Ein ausführliches Beispiel dazu finden Sie unter [Beispiel: Eine Rechnung](#).

Automatischen Berechnung in unterschiedlichen Vorlagen

Wenn sich zwei automatische Berechnungen in unterschiedlichen Vorlagen befinden, kann ein durch die erste automatische Berechnung aktualisierter Node von der zweiten automatischen Berechnung verwendet werden. Dies liegt daran, dass das Ergebnis automatischer Berechnungen berechnet wird und die Nodes für jede Vorlage separat aktualisiert werden. Ein Beispiel dazu finden Sie unter [Beispiel: Eine Rechnung](#).

[Beispiel: Eine Rechnung](#): Ausführliches Beispiel zur Verwendung von automatischen Berechnungen.

[Dialogfeld "XPath"](#): Hier finden Sie eine Beschreibung zum Dialogfeld "XPath-Ausdruck bearbeiten".

Beispiel: Eine Rechnung

Anhand des Beispiels `Invoice.sps` im [Ordner \(Eigene\) Dateien](#), `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/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
- Aktualisieren des Inhalts eines Node mit dem Ergebnis einer automatischen Berechnung
- Verwendung des Ergebnisses der automatischen Berechnung in einer anderen automatischen Berechnung

In der Beispieldatei wurden die automatischen Berechnungen mit einer gelben Hintergrundfarbe markiert (siehe Abbildung unten).

Zählen von Nodes

Im Beispiel "Rechnung" 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).

Product 1:	Learning XMLSpy
Net price:	€ 35.00
Category:	Book
VAT:	10%
Price including VAT:	€ 38.5

Product 2:	Scooby Doo's Greatest Hits

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 "Rechnung" 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 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. Um diese automatische Berechnung zu testen, ändern Sie die ausgewählte Produktart in der Auswahlliste einer beliebigen Produktkategorie; der Mehrwertsteuerwert ändert sich entsprechend (`Book=10%`; `CD=15%`; `DVD=15%`; `Electronics=20%`).

Aktualisieren eines Node-Inhalts mit dem Ergebnis einer automatischen Berechnung

Der durch die automatische Berechnung anhand der Nachschlagetabelle ermittelte Mehrwertsteuersatz ist dynamisch und wird temporär (für die Verwendung in der Authentic-Ansicht) im Arbeitsspeicher gespeichert. Das Ergebnis der automatischen Berechnung kann jedoch im `VAT` Node im XML-Dokument gespeichert werden. Dies hat zwei Vorteile: (i) Der Inhalt des `VAT` Node muss nicht vom Benutzer eingegeben werden, sondern wird automatisch mittels der automatischen Berechnung eingefügt; (ii) Jedes Mal, wenn die Nachschlagetabelle geändert wird, werden die Änderungen im `VAT` Node übernommen, wenn das Dokument `Invoice.xml` in Authentic geöffnet wird. Damit ein Node mittels einer automatischen Berechnung aktualisiert wird, wählen Sie die automatische Berechnung in der Design-Ansicht und klicken Sie im Fenster "Eigenschaften" (*Abbildung unten*) auf die Schaltfläche *Authentic | XML-Node in Authentic aktualisieren*. Wählen Sie im Dialogfeld, das daraufhin angezeigt wird, den `VAT` Node aus und klicken Sie auf **OK**.



Wenn der Benutzer in der Authentic-Ansicht in der Auswahlliste eine andere Produktkategorie auswählt, ermittelt die automatische Berechnung den Mehrwertsteuersatz durch Referenzierung der Nachschlagetabelle, zeigt den Mehrwertsteuersatz an und aktualisiert den `VAT` Node.

Verwendung eines durch eine automatische Berechnung aktualisierten Node in einer anderen 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:

$$\begin{aligned} \text{Bruttopreis} &= \text{Nettopreis} + \text{Mwst-Betrag} \\ \text{Da } \text{Mwst-Betrag} &= \text{Nettopreis} * \text{Mwstsatz} \text{ div } 100 \\ \text{Bruttopreis} &= \text{Nettopreis} + (\text{Nettopreis} * \text{Mwst-Betrag} \text{ div } 100) \end{aligned}$$

Der Nettopreis eines Produkts wird aus dem Node `PriceNet` abgerufen. Der

Mehrwertsteuersatz wird mittels der oben beschriebenen automatischen Berechnung ermittelt. Mit dieser automatischen Berechnung wird der VAT Node (Mwst-Node) aktualisiert. Der Inhalt des VAT Node kann nun in einer automatischen Berechnung verwendet werden, um den Bruttopreis zu berechnen. Der XPath-Ausdruck dafür würde lauten:

```
PriceNet + (PriceNet * (VAT div 100))
```

Der XPath-Ausdruck kann [im Fenster "Eigenschaften" angezeigt und bearbeitet werden](#). Sie können die automatische Berechnung für den Bruttopreis testen, indem Sie entweder den Preis oder die Produktkategorie eines beliebigen Produkts ändern. Beachten Sie, dass sich auch der Bruttopreis (Preis inkl. Mwst) des Produkts ändert.

Product 6:	A Short History of the American Century
Net price:	€ 20.00
Category:	DVD
VAT:	15%
Price including VAT:	€ 23
Price Total: € 358.9	

Im Invoice-SPS aktualisiert die Bruttopreisberechnung den Node `PriceGross` im XML-Dokument.

Die aktualisierten `PriceGross` Nodes können nun in einer automatischen Berechnung verwendet werden, in der die Preise aller erworbenen Produkte summiert werden. Der XPath-Ausdruck dafür würde lauten: `sum( Order/Product/PriceGross )`. Im Invoice-SPS aktualisiert diese automatische Berechnung den Node `PriceTotal`. Sie können diese automatische Berechnung testen, indem Sie die Preise der Einzelprodukte ändern und überprüfen, wie sich das auf den Gesamtpreis auswirkt.

Eine Übung zur automatischen Berechnung

Fügen Sie nun selbst zwei automatische Berechnungs-Komponenten zum SPS hinzu.

1. Erstellen Sie eine automatische Berechnung, mit der ein Mengenrabatt für die gesamte Rechnung berechnet wird. Wenn der Betrag der Bestellung (price total) 100 Euro, 300 Euro bzw. 600 Euro übersteigt, wird ein Rabatt von 5%, 10% bzw. 12% eingeräumt. Zeigen Sie den Rabattbetrag an (*siehe Abbildung unten*) und aktualisieren Sie den Node `DiscountAmount` im XML-Dokument.
2. Erstellen Sie eine automatische Berechnung, mit der der ermäßigte Rechnungsbetrag ermittelt wird. Der Betrag wäre der Gesamtpreis minus den (in der vorherigen automatischen Berechnung ermittelten) Rabattbetrag. Zeigen Sie den Rechnungsbetrag an (*siehe Abbildung unten*) und aktualisieren Sie den Node `BillAmount` im XML-Dokument.

Richten Sie diese Komponenten so ein, dass sie in der Authentic-Ansicht, wie in der Abbildung unten gezeigt werden.

Price Total:	€	358.90	<u>Volume Discounts</u>
Less Volume Discount:	€	35.89	Over € 100 = 05%
Bill Amount:	€	323.01	Over € 300 = 10%
			Over € 600 = 12%

Sie finden diese beiden zusätzlichen Berechnungen in der Datei `InvoiceWithDiscounts.sps`, die sich im [Ordner \(Eigene\) Dateien](#), `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/Auto-Calculations` befindet.

10.2 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 Bedingungs-vorlage. Wenn ein Node mit einer Bedingung verarbeitet wird, so wird die erste Verzweigung, bei dessen Auswertung das Ergebnis "true" ergibt, ausgeführt, d.h. die Bedingungs-vorlage 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](#): Hier wird beschrieben, wie eine Bedingung und ihre Verzweigungen erstellt werden.
- [Bearbeiten von Bedingungen](#): Hier erfahren Sie, wie XPath-Ausdrücke von Bedingungs-zweigen nach ihrer Erstellung bearbeitet werden.
- [Bedingungen für eine bestimmte Ausgabe](#): Hier wird beschrieben, wie Bedingungen zum Erzeugen unterschiedlicher Ausgaben für unterschiedliche Ausgabeformate verwendet werden.
- [Bedingungen und automatische Berechnungen](#): Hier wird erklärt, wie Bedingungen und automatische Berechnungen gemeinsam verwendet werden.

Definieren der Bedingungen

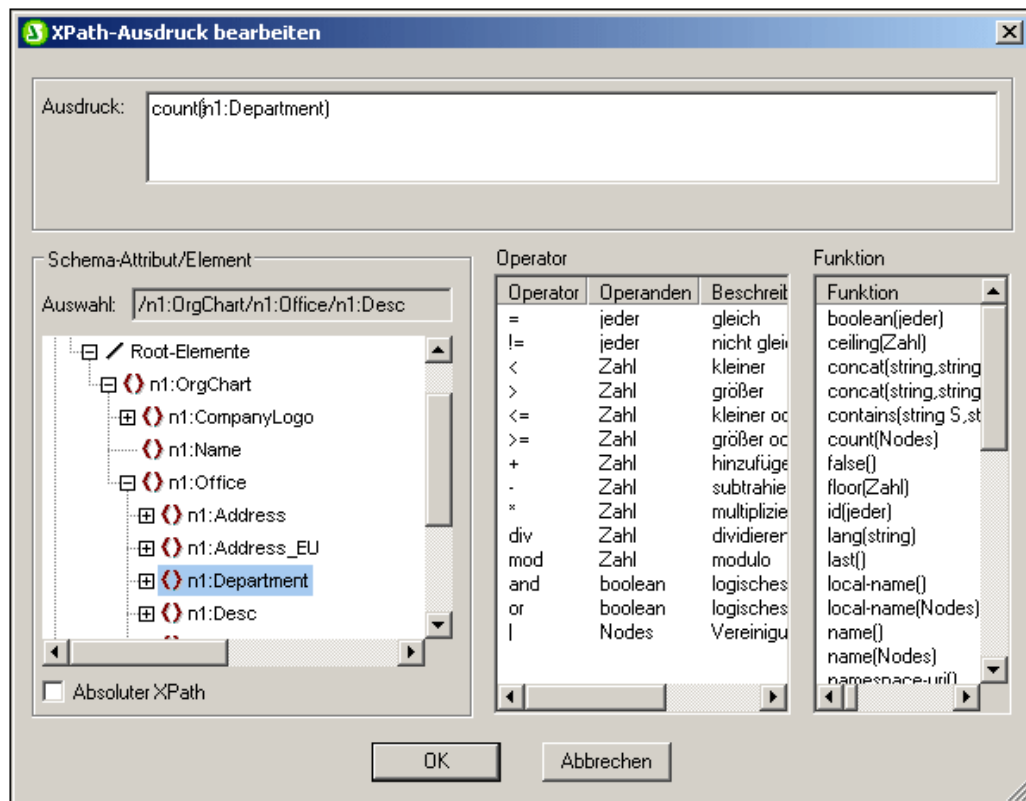
Das Definieren der Bedingung erfolgt folgendermaßen:

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 Bedingungsweisen.

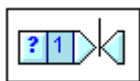
Erstellen der Bedingung mit ihrer ersten Verzweigung

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" angezeigt (*Abbildung unten*).



2. Geben Sie im Bereich "Ausdruck" über die Tastatur den XPath-Ausdruck für den Bedingungsweig ein. Alternativ dazu können Sie den Ausdruck auch durch Doppelklick auf Nodes, Operatoren und/oder Funktionen in den Seitenleiste 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 Texteingabepunkt 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.

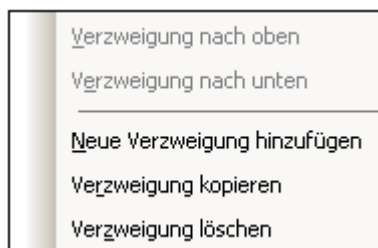


Um die gesamte Bedingung auszuwählen, klicken Sie auf die Zelle mit dem Fragezeichen. Um die erste Verzweigung auszuwählen, klicken Sie auf die Zelle mit der Nummer eins.

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**).



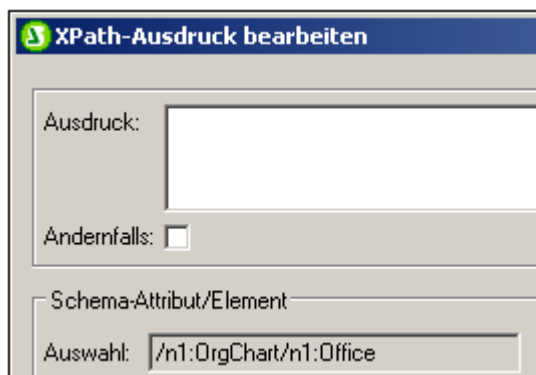
Um eine neue Verzweigung zu erstellen, rechtsklicken Sie auf eine Verzweigung der Bedingung und wählen Sie im Kontextmenü den Befehl **Neue Verzweigung hinzufügen**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" angezeigt. Nachdem Sie einen XPath-Ausdruck eingegeben haben und auf OK geklickt 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" 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 Bedingung 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" das Kontrollkästchen "Andernfalls" (*Abbildung unten*).

**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**.

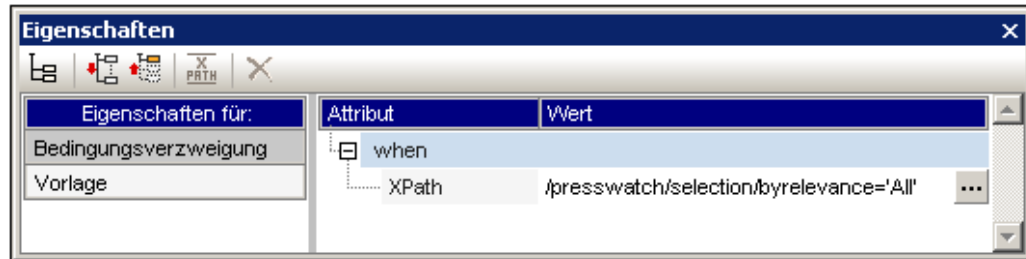
Löschen einer Verzweigung

Um eine Verzweigung zu löschen, wählen Sie sie aus, rechtsklicken Sie und wählen Sie **Verzweigung löschen**.

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" *Bedingungsverzweigung* aus (Abbildung unten).



3. Klicken Sie in der Eigenschaftsgruppe *when* auf die **Bearbeiten**-Schaltfläche  der Eigenschaft *XPath*. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" angezeigt. Hier können Sie den XPath-Ausdruck für diese Verzweigung bearbeiten.

Ausgabe-basierte Bedingungen

Einzelne Komponenten im Dokument-Design können für die unterschiedlichen Ausgabeformate von StyleVision (Authentic-Ansicht und HTML) unterschiedlich verarbeitet werden.

Angenommen Sie haben den Fall, dass Sie einen Link erstellen möchten, der in der Authentic-Ansicht auf eine Datei auf einem lokalen System, aber in der HTML-Ausgabe auf eine Webseite verweisen soll. In diesem Fall können Sie eine Bedingung zum Verarbeiten von Inhalt für die Ausgabe der Authentic-Ansicht und eine zweite Bedingung zur Verarbeitung von Inhalt für die HTML-Ausgabe erstellen. Oder angenommen, Sie möchten einen bestimmten Text in die Ausgabe der Authentic-Ansicht inkludieren, nicht aber in die HTML-Ausgabe. Es könnte eine Bedingung mit einer Verzweigung zur Verarbeitung der Ausgabe für die Authentic-Ansicht und einer Verzweigung für die HTML-Ausgabe erstellt werden.

Anmerkung: Bedingungen für eine bestimmte Ausgabe können rund um einzelne Teile oder Komponenten des Dokuments platziert werden, sodass die verschiedenen Ausgabedokumente unterschiedlich strukturiert werden können.

Erstellen von Bedingungen für eine bestimmte Ausgabe

Um Bedingungen für eine bestimmte Ausgabe zu erstellen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht die Komponente aus (oder markieren Sie den Teil des Dokuments), der in unterschiedlichen Ausgabeformaten unterschiedlich ausgegeben werden soll.
2. Klicken Sie auf die rechte Maustaste und wählen Sie im Kontextmenü, das daraufhin angezeigt wird, **Einschließen in | Ausgabe-basierte Bedingung einfügen**. Daraufhin wird die Ausgabebedingung mit drei Verzweigungen eingefügt, wobei jede Verzweigung denselben Inhalt (nämlich die ausgewählte Komponente) hat. Jede Verzweigung steht für eine einzelne Ausgabe (Authentic-Ansicht, RTF, oder HTML). Um festzustellen, welche Verzweigung für welche Ausgabe steht, positionieren Sie den Mauszeiger über den Verzweigungs-Tag oder überprüfen Sie den XPath-Ausdruck der ausgewählten Verzweigung (klicken Sie in der Seitenleiste "Eigenschaften" im Eintrag Bedingungsverzweigung auf die [Bearbeiten-Schaltfläche](#)).
3. Definieren Sie in jeder Verzweigung die gewünschte Verarbeitung. Wenn für ein bestimmtes Ausgabeformat keine Verarbeitung erfolgen soll, löschen Sie die Verzweigung für dieses Format (wählen Sie die Verzweigung und drücken Sie **Löschen** oder wählen Sie die Verzweigung und anschließend im Kontextmenü **Verzweigung löschen**).

Anmerkung: Es kann auch zuerst die Ausgabe-basierte Bedingung erstellt werden und (statischer und/oder dynamischer) Inhalt kann später eingefügt werden. Fügen Sie zuerst die Ausgabe-basierte Bedingung an einem Cursoreinfügekpunkt im Design ein. Fügen Sie anschließend den gewünschten statischen und/oder dynamischen Inhalt in die jeweilige Verzweigung ein.

Bearbeiten der Verzweigungen einer Ausgabe-basierten Bedingung

Der XPath-Ausdruck einer Verzweigung einer Ausgabe-basierten Bedingung ist

`$SV_OutputFormat = 'format'`, wobei `format` einer der folgenden Werte ist: `Authentic` oder `HTML`. Sie können den XPath-Ausdruck einer Bedingungsverzweigung bearbeiten (klicken Sie in der Seitenleiste "Eigenschaften" im Eintrag Bedingungsverzweigung auf die [Bearbeiten-Schaltfläche](#)). So könnten Sie z.B. die Authentic-Ansicht und das HTML-Ausgabeformat in einer Bedingungsverzweigung (unter Verwendung des XPath-Ausdrucks: `$SV_OutputFormat = 'Authentic' or $SV_OutputFormat = 'HTML'`) kombinieren.

Außerdem können Sie (i) eine oder mehrere Verzweigungen löschen; (b) eine `andernfalls` Verzweigung in einer Bedingung erstellen; und (c) die Verzweigungen relativ zueinander nach

oben oder nach unten verschieben und so die relative Priorität der Verzweigungen ändern. Nähere Informationen zum Ausführen dieser Aktionen finden Sie unter [Definieren der Bedingungen](#) und [Bearbeiten von Bedingungen](#).

Verwendung des Parameters `$SV_OutputFormat`

In der XSLT-Datei, die für die einzelnen Ausgabedateien erstellt wird, wird `$SV_OutputFormat` als globaler Parameter erstellt und es wird ihm der entsprechende Wert für dieses Ausgabeformat zugewiesen (d.h., `Authentic`, `RTF` oder `HTML`). Dieser Parameter kann außer Kraft gesetzt werden, indem zur Laufzeit ein anderer Wert dafür an den Prozessor übergeben wird. Dies könnte nützlich sein, wenn Sie z.B. zwei unterschiedliche HTML-Ausgabeoptionen erstellen möchten, von denen jede zur Laufzeit ausgewählt wird. Anschließend könnten Sie Bedingungsverzweigungen erstellen `$SV_OutputFormat = 'HTML-1'` and `$SV_OutputFormat = 'HTML-2'`. Zur Laufzeit könnten Sie den erforderlichen Parameterwert (`HTML-1` oder `HTML-2`) an den Prozessor übergeben. Nähere Informationen zur Verwendung von StyleVision über die Befehlszeile finden Sie unter [Befehlszeilenschnittstelle: StyleVisionBatch](#).

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, und dass der Node immer aktualisiert wird, wenn die automatische Berechnung ausgelöst wird. Befände sich die automatische Berechnung innerhalb einer Verzweigung, die nicht sichtbar ist, würde sie nicht ausgelöst und der Node würde nicht aktualisiert.
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.

10.3 Bedingte Präsenz

Die Funktion "Bedingte Präsenz" ermöglicht Ihnen, bestimmte Design-Elemente zu erstellen, wenn eine angegebene Bedingung erfüllt ist. Folgende Design-Elemente können bedingt erstellt werden:

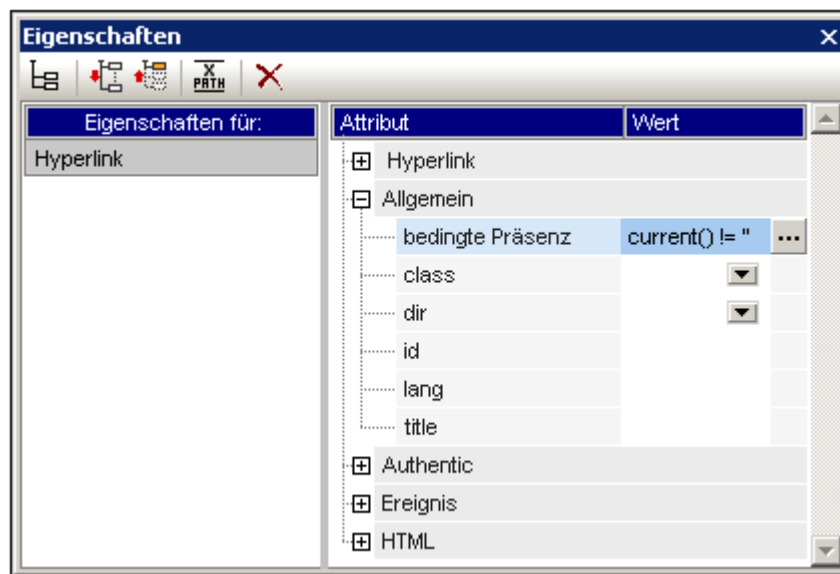
- [Hyperlinks](#)
- [Lesezeichen](#)
- [Neue Dokumente](#)
- [Benutzerdefinierte Elemente](#)
- [Ebenen von Inhaltsverzeichnissen](#)

Definieren von bedingter Präsenz

Um zu definieren, dass eine dieser Design-Komponenten bedingt erstellt werden soll und um die benötigte Bedingung zu formulieren, befolgen Sie folgendes Verfahren:

1. Markieren Sie die Design-Komponente.
2. Klicken Sie in der Seitenleiste "Eigenschaften" dieser Design-Komponente in der

Eigenschaftsgruppe *Allgemein* (Abbildung unten) auf das Symbol XPath bearbeiten 



3. Im Dialogfeld "XPath bearbeiten", das erscheint, geben Sie den XPath-Ausdruck ein, der als Bedingung erfüllt werden muss, damit die Design-Komponente in der Ausgabe eingesetzt wird.
4. Klicken Sie zum Fertigstellen auf **OK**.

Anmerkung: Wenn die Bedingung erfüllt ist, wird die Design-Komponente eingesetzt. Im anderen Fall wird sie das nicht, sämtlicher Inhalt der Design-Komponente wird jedoch – ohne dass diese Design-Komponente vorhanden ist – ausgegeben. In der obigen Abbildung wird zum Beispiel ein Hyperlink mit bedingter Präsenz erstellt. Die Bedingung überprüft, ob der aktuelle Node nicht leer ist. Wenn der Node nicht leer ist, wird die Bedingung erfüllt und der Hyperlink wird erstellt. Der Text des Hyperlinks wird aus dem Inhalt der Design-Komponente für den Hyperlink übernommen. (Die URL des Hyperlinks wird an anderer Stelle, in der Eigenschaftsgruppe *Hyperlink*, angegeben. Wird die Bedingung nicht erfüllt, dann wird der Text des Hyperlinks (übernommen aus der Design-Komponente "Hyperlink") zwar ausgegeben, jedoch als normaler Text und

nicht als Hyperlink.

Auf die gleiche Weise gilt für die anderen Design-Komponenten, die bedingt erstellt werden können, dass es die Design-Komponente selbst ist, die bedingt erstellt bzw. nicht erstellt wird. Der Inhalt der Design-Komponente wird in beiden Fällen erstellt (gleich ob die Bedingung erfüllt wird oder nicht).

10.4 Gruppieren

Die Gruppierungsfunktion steht in **XSLT 2.0** SPSs und für HTML- und RTF-Ausgabedokumente zur Verfügung. **Für die Ausgabe in der Authentic-Ansicht wird die Gruppierungsfunktion nicht unterstützt.**

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.

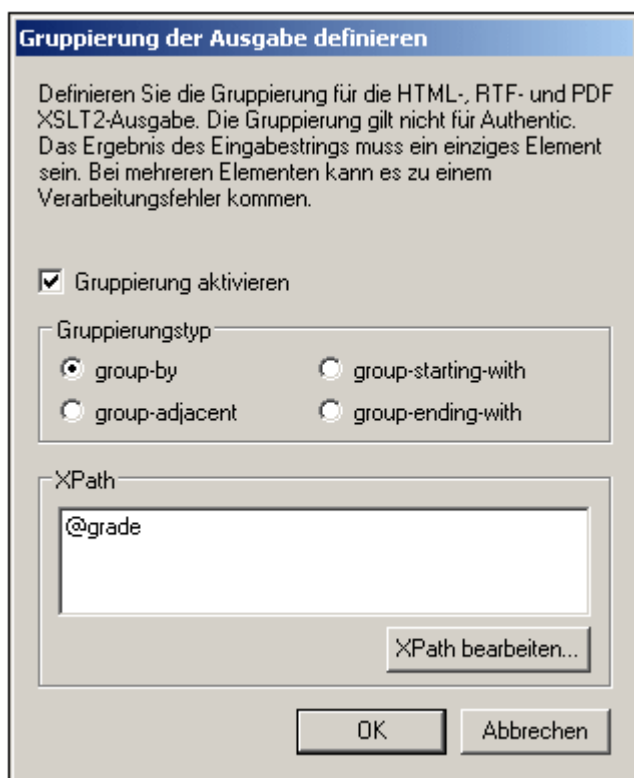
Gruppierungskriterien

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:

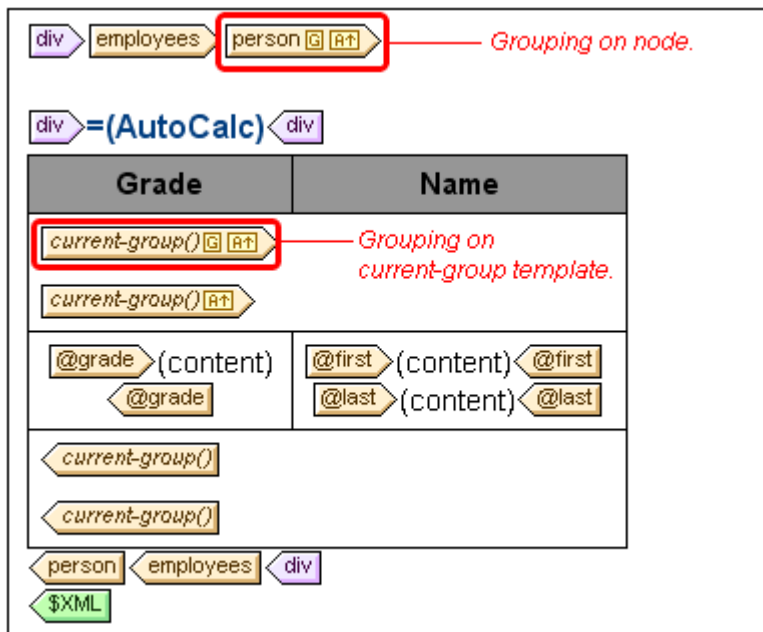
- **group-by:** Gruppiert die Elemente auf Basis eines XPath-definierten Schlüssels. So können z.B. `car` Elemente nach ihren `brand` Attributen gruppiert werden. Die Gruppierung wird nach dem Element `car` vorgenommen. Ein XPath-Ausdruck wählt das `brand` Attribut aus.
- **group-adjacent** verwendet 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.



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 Gruppeneinstellung 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](#). Eine Beschreibung zur Verwendung dieses Dialogfelds finden Sie im Abschnitt [Sortieren](#). 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](#), 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.

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](#), C: / Documents and Settings / <username> / My Documents / Altova / StyleVision2012 / 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.

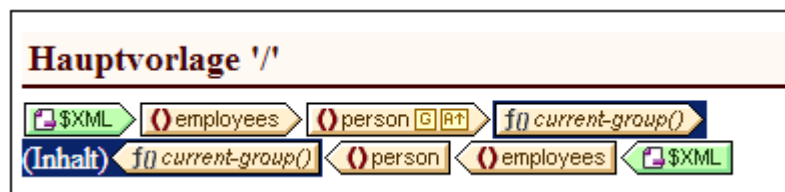
Strategie

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

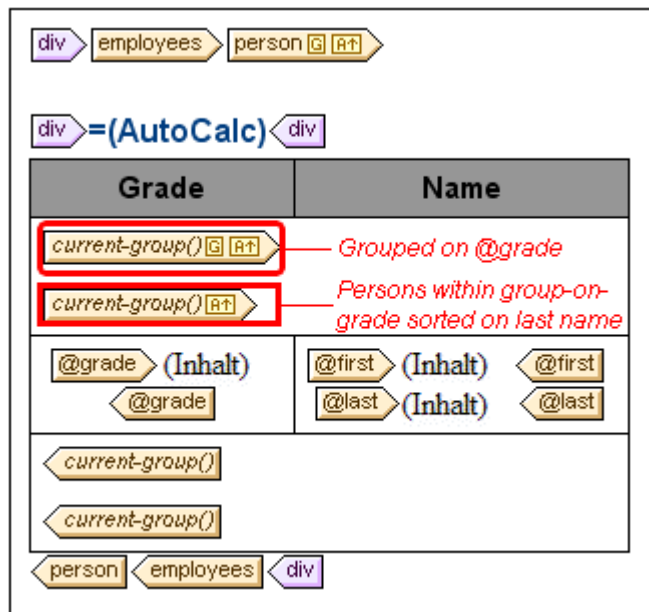
Das Design wurde folgendermaßen erstellt:

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 **OK**. (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.



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.



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-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.

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](#), `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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).

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

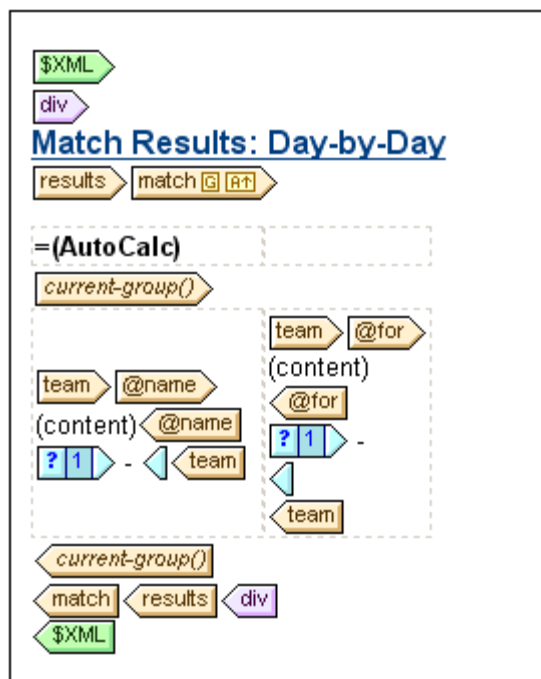
Für die beiden Abschnitte, die die Match-Ergebnisse enthalten, gruppieren wir Matches nach Datum und Turniergruppe. Für die Mitglieder der einzelnen Gruppen (Datum und Turniergruppe) erstellen wir randlose Tabellen (um den Inhalt ausrichten zu können). Auf diese Weise befinden sich Matches, die an einem 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 **OK**. (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`.

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**.

10.5 Sortieren

Die Sortierfunktion steht in XSLT 1.0 und XSLT 2.0 SPSs und für die HTML- und RTF-Ausgabe zur Verfügung. **Für die Ausgabe in der Authentic-Ansicht steht diese Funktion nicht 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- und RTF-Ausgabe unterstützt.

Benutzerdefinierte Vorlagen

[Benutzerdefinierte Vorlagen](#) sind Vorlagen, die auf Elemente angewendet werden, die mit Hilfe eines von Ihnen definierten XPath-Ausdrucks ausgewählt werden. Die durch den XPath-Ausdruck einer benutzerdefinierten Vorlage ausgewählten Nodes können auch sortiert werden. In diesem Fall wird die Sortierung an der benutzerdefinierten Vorlage angewendet.

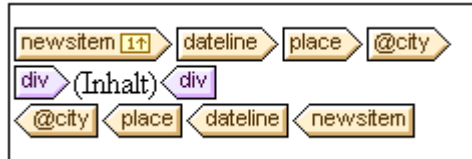
In diesem Abschnitt

- Beschreibung [des Sortiermechanismus](#).
- Ein [Beispiel](#) zur Beschreibung der Verwendung der Sortierfunktion.

Funktionsweise

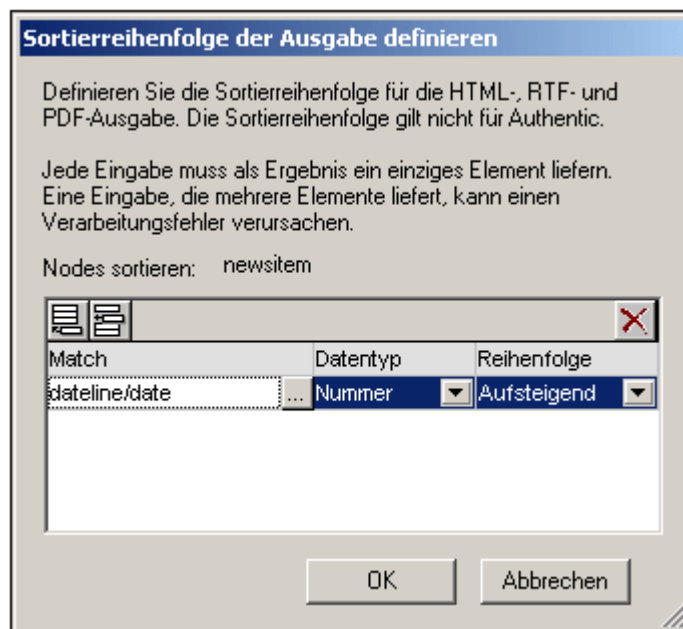
Die Konfiguration eines Schema-Element-Nodes für die Sortierung erfolgt in zwei Schritten:

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.



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](#).

Benutzerdefinierte Vorlagen

[Benutzerdefinierte Vorlagen](#) sind Vorlagen, die auf Elemente angewendet werden, die mit Hilfe eines von Ihnen definierten XPath-Ausdrucks ausgewählt werden. Die durch den XPath-Ausdruck einer benutzerdefinierten Vorlage ausgewählten Nodes können auch sortiert werden. In diesem Fall wird die Sortierung an der benutzerdefinierten Vorlage angewendet.

Anmerkung zu Sortierschlüsseln

Sowohl in XSLT 1.0 als auch XSLT 2.0 SPSs dürfen die für den Sortierschlüssel eingegebenen XPath-Ausdrücke für jede Elementinstanz nur einen **einzigsten Node** auswählen und kein Nodeset (XPath 1.0) und keine Elementsequenz (XPath 2.0); der Schlüssel für jedes Element sollte sich als String oder Zahlenwert auflösen.

Wenn in einem **XSLT 2.0** 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).

Beispiel: Sortieren nach mehreren Sortierschlüsseln

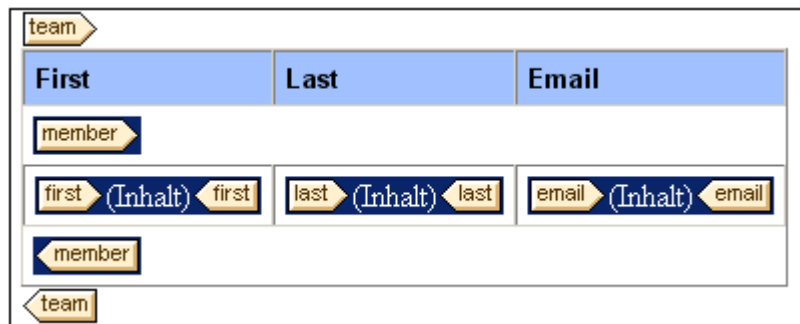
Schritt-für-Schritt-Beispiel

In dem einfachen Beispiel unten (zu finden im [Ordner \(Eigene\) Dateien](#), C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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*).

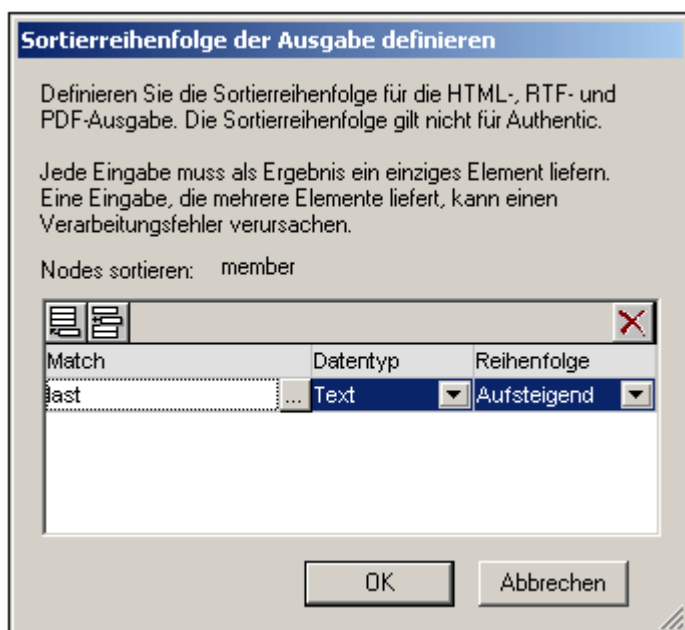
First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

Klicken Sie in der Design-Ansicht mit der rechten Maustaste auf das `member` Element (*erscheint in der Abbildung unten 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 `members` 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-

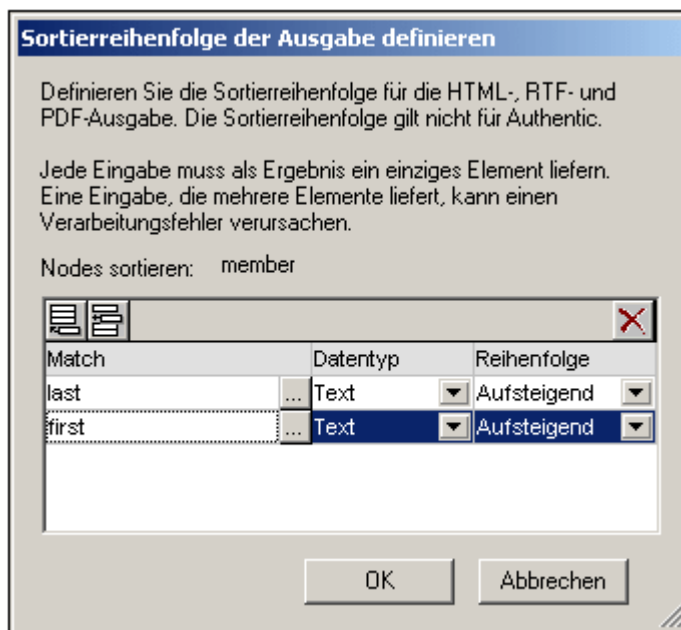
Schaltfläche  klicken, um den XPath-Ausdruck zu erzeugen. In der Spalte "Datentyp" können Sie auswählen, wie der Sortierschlüsselinhalt ausgewertet werden soll: als Text oder als Zahl. In der Spalte "Reihenfolge" wird die Sortierreihenfolge aufgelistet: Aufsteigend oder absteigend. Wählen Sie `Text` und `Aufsteigend`. Klicken Sie zum Fertigstellen auf **OK**.



In der Design-Ansicht enthält der `member` Tag ein Symbol, das anzeigt, dass er einen Sortierfilter enthält `member [a↑]`. 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.

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

10.6 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](#): Hier wird die Verwendungsweise von benutzerdefinierten Parametern in einem SPS erklärt.
- Unter [Parameter für Design-Fragmente](#) wird beschrieben, wie Parameter mit Design-Fragmenten verwendet werden können.
- [SPS Parameter für Quelldokumente](#): 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](#): 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.
- Mit Hilfe von editierbaren Variablen in Authentic kann der Benutzer Variablen in der Authentic-Ansicht bearbeiten und somit festlegen, welcher Inhalt in der Authentic-Ansicht angezeigt wird.

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. Der Standardwert des Parameters kann bei einzelnen XSLT-Transformationen außer Kraft gesetzt werden, indem über die [Befehlszeile](#) ein neuer globaler Wert an das XSLT Stylesheet übergeben wird.

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 die [Befehlszeile](#) an den Parameter übergeben.

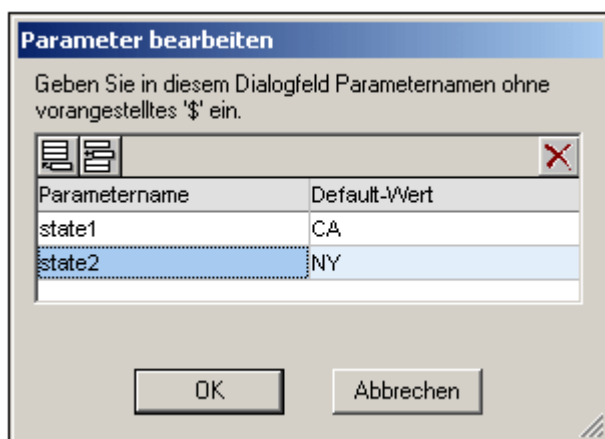
Funktionsweise

Die Verwendung von benutzerdeklarierten Parametern im SPS erfolgt in zwei Schritten:

1. [Deklariere den gewünschten Parameter](#)
2. [Referenzieren des deklarierten Parameters](#)

Deklariere Parameter

Alle benutzerdefinierten Parameter werden im Dialogfeld "Parameter bearbeiten" (*Abbildung unten*) deklariert und bearbeitet. Das Dialogfeld "Parameter bearbeiten" wird folgendermaßen aufgerufen: (i) über den Befehl [Bearbeiten | Stylesheet Parameter](#), (ii) über die Schaltfläche **Parameter** im Dialogfeld "Datenbankfilter bearbeiten" ([Bearbeiten | DB-Filter bearbeiten](#)) und (iii) in der [Seitenleiste "Design-Übersicht"](#) über die **Bearbeiten**-Schaltfläche  des Parametereintrags.



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. Der Standardwert wird bei jeder Referenzierung des Parameters verwendet und nur dann außer Kraft gesetzt, wenn über die [Befehlszeile](#) ein neuer Wert für diesen Parameter übergeben wird.

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 oder die Authentic-Ansicht bearbeiten.

Anmerkung:

- Das Dialogfeld "Parameter bearbeiten" enthält alle benutzerdefinierten Parameter in einem SPS.
- Parameter können auch in der [Seitenleiste Design-Übersicht](#) 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' )`). Wenn Ihr SPS auf einer Datenbank basiert, können Sie Parameter auch als Werte von DB-Filterkriterien verwenden. Die DB-Parameter werden jedoch im [Dialogfeld "Parameter bearbeiten" deklariert und bearbeitet](#).

Anmerkung: Die Referenzierung eines undeklarierten Parameters ergibt einen Fehler. Wird jedoch ein Parameter deklariert und nicht referenziert, ist dies kein Fehler.

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.ja` 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:

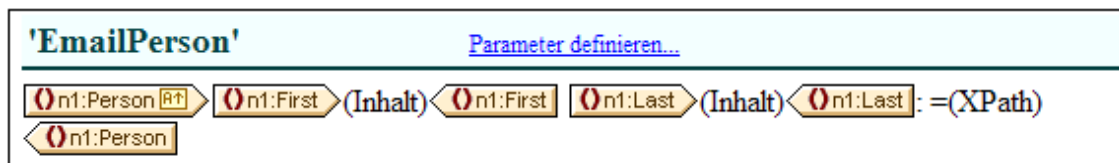
1. [Definition des Parameters](#) mit einem Standardwert für das Design-Fragment, für das er erstellt wird.
2. [Bearbeitung des Parameterwerts](#) an den Stellen, an denen das Design-Fragment verwendet wird.

Im Folgenden werden diese Schritte näher beschrieben.

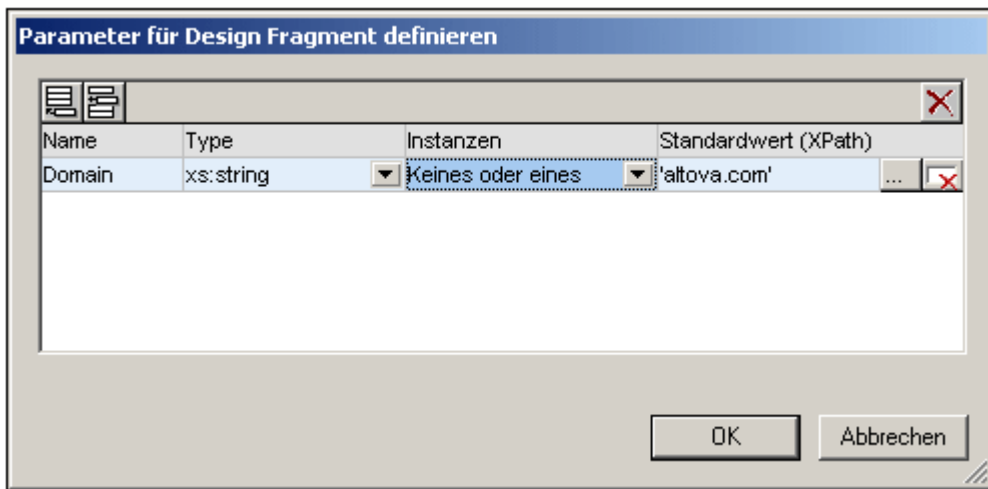
Anmerkung: Parameter für Design-Fragmente werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

Definition des Parameters

Jedem Design-Fragment kann eine ganze Reihe von Parametern zugewiesen werden. Klicken Sie dazu in der Titelleiste des Design-Fragments (*siehe Abbildung unten*) auf den Link "Parameter definieren".



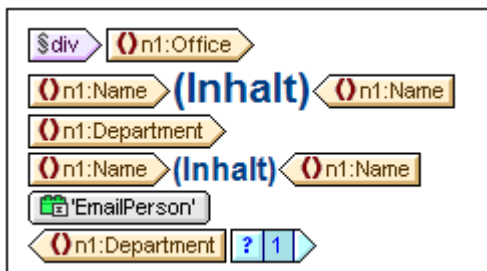
Daraufhin wird das Dialogfeld "Parameter für Design Fragment definieren" (*Abbildung unten*) aufgerufen. Klicken Sie auf das **Anhängen** oder das **Einfügen** Symbol links oben, um eine Parametereingabezeile hinzuzufügen. Geben Sie den Namen, den Datentyp, die Anzahl der Instanzen und den Standardwert 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 genau eins. Sie können beliebig viele Parameter hinzufügen.



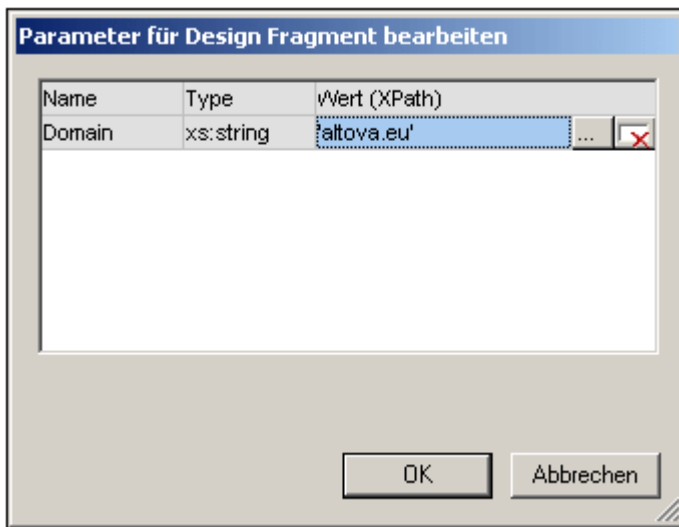
Es gibt zwei Arten von **Löschsymbolen**. Mit dem Löschsymbol rechts von den Parametereinträgen löschen Sie den Standardwert dieses Parameters. Mit dem **Löschsymbol** im rechten oberen Eck des Fensters löschen Sie den gerade markierten Parameter.

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 `EmailAddress`, 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 bearbeitet 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.

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 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.

Variablen

Die Verwendung von Variablen erfolgt in zwei Schritten: (d) [Deklarieren einer Variablen](#) und (ii) [Verwenden der Variablen](#).

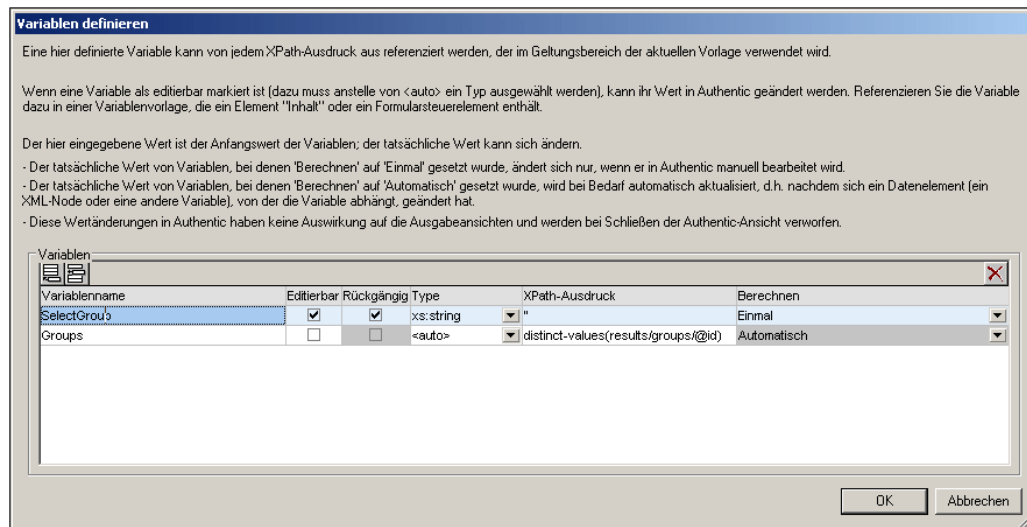
Anmerkung: Variablen werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

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 . 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

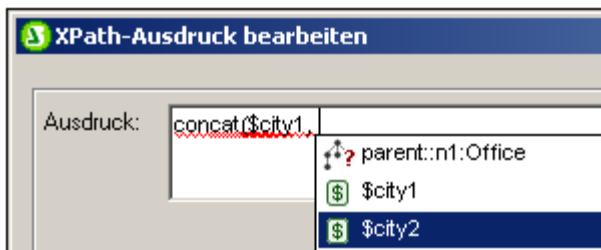
- editierbares Textfeld erstellt wurde und dieses vom Benutzer der Authentic-Ansicht bearbeitet wird), oder wenn ein im XPath-Ausdruck der Variablen verwendeter Node oder Wert durch den Benutzer der Authentic-Ansicht geändert wird.
4. Wenn die Variable auf "Editierbar" gesetzt wurde, so werden zwei weitere 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.

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" 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.

10.7 Inhaltsverzeichnis, Referenzierung, Lesezeichen

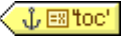
Mit den Inhaltsverzeichnis- und Referenzierungsfunktionen werden Markierungen an den gewünschten Punkten im Design gesetzt und diese werden vom Inhaltsverzeichnis, von Textreferenzen, automatischen Nummerierungssequenzen und Hyperlinks aus referenziert. Es werden zwei Mechanismen verwendet:

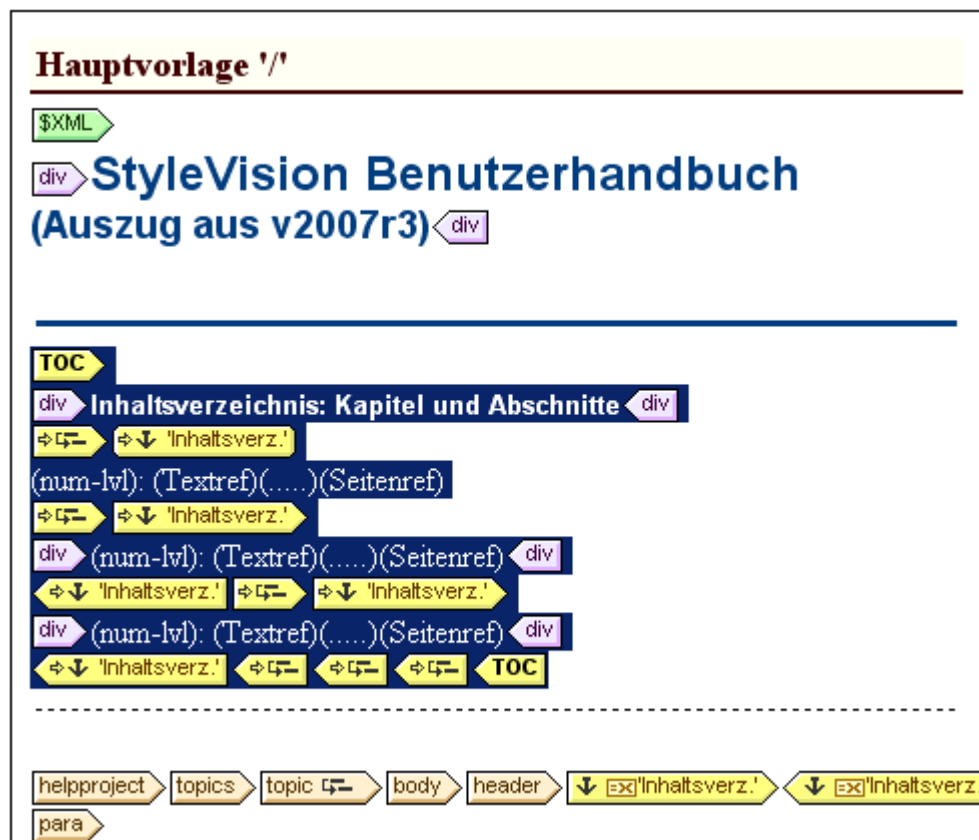
- An einer bestimmten Stelle im Design-Dokument wird ein einfacher Anker erstellt. Der Anker (oder die Markierung) erhält einen eindeutigen Namen. Dieser Name wird als Ziel von Links verwendet, die auf dieses Dokumentfragment verweisen. Dieser Mechanismus wird für die [Lesezeichen- und Hyperlinkfunktion](#) verwendet. Links können zusätzlich auch auf URLs außerhalb des Dokuments verweisen.
- Bei komplexeren Referenzierungen wie z.B. Inhaltsverzeichnissen und der automatischen Nummerierung von Dokumentabschnitten erfolgt die Erstellung des Ankers in zwei Schritten. Zuerst wird für das Design-Dokument die für das Inhaltsverzeichnis benötigte Hierarchie angelegt. Dazu werden verschiedenen Stellen in der Dokumentstruktur Ebenen zugewiesen. So kann einem Buchkapitel z.B. eine Ebene zugewiesen werden. Als nächstes muss der Text definiert werden, der in der referenzierenden Komponente angezeigt werden soll. Nach Definition der Ebenen und des Referenztextes kann die referenzierende Komponente erstellt werden. Weiter unten finden Sie unter [Die Funktionsweise beim Erstellen von Inhaltsverzeichnissen](#) einen Überblick über die Funktionsweise.

Die verschiedenen Referenzierungsfunktionen werden im restlichen Teil dieses Abschnitts näher beschrieben.

Die Funktionsweise beim Erstellen von Inhaltsverzeichnissen

Wenn Sie die Option [XSLT 2.0](#) (nicht XSLT 1.0) als XSLT-Version für Ihr SPS ausgewählt haben, können Sie an jeder beliebigen Stelle in Ihrem Design ein Inhaltsverzeichnis erstellen. Das Verfahren zum Erstellen eines Inhaltsverzeichnisses besteht aus zwei in den Unterabschnitten dieses Abschnitts beschriebenen Teilen:

- Die Elemente im Design, die im Inhaltsverzeichnis enthalten sein sollen, werden [im Design markiert](#). Dabei kann es sich um statischen oder dynamischen Inhalt handeln. In der unteren Hälfte der Abbildung unten, wird das Element `header` durch gelbe Inhaltsverzeichnis-Tags  `<toc>`  für das Einschließen in das Inhaltsverzeichnis markiert.
- [Für das Inhaltsverzeichnis wird eine Vorlage erstellt](#) (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.



Es spielt keine Rolle, welcher der beiden Teile zuerst erstellt wird, es können auch beide Teile gleichzeitig erstellt werden.

Das Inhaltsverzeichnis wird in der Authentic-Ansicht und in der HTML- und RTF-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

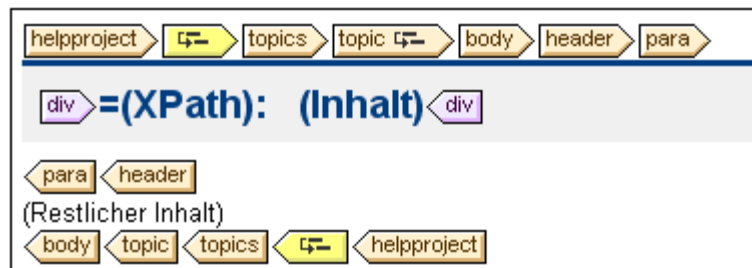
Im Folgenden finden Sie eine schrittweise Anleitung zum Erstellen eines Inhaltsverzeichnisses. Dabei werden zuerst Elemente markiert, die in das Inhaltsverzeichnis aufgenommen werden sollen, und anschließend wird die Inhaltsverzeichnis-Vorlage erstellt. (Alternativ dazu können Sie auch zuerst die Inhaltsverzeichnis-Vorlage 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](#) ausgewählt ist.
2. [Strukturieren Sie das Dokument in Ebenen](#). Wenn das Inhaltsverzeichnis mehrere Ebenen haben soll, strukturieren Sie das Design in einer Hierarchie verschachtelter Ebenen. Wenn das Inhaltsverzeichnis eine flache Struktur haben soll (also nur eine Ebene), erstellen Sie zumindest eine Ebene, die die Inhaltsverzeichnismarkierungen enthalten soll.
3. [Erstellen Sie ein oder mehrere Inhaltsverzeichnismarkierungen 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 Inhaltsverzeichnis-Vorlage sollte die erforderliche Anzahl von Inhaltsverzeichnis-Referenzebenen haben. Im Fall des Inhaltsverzeichnisses mit mehreren Ebenen sollten die Referenzebenen verschachtelt sein (*siehe Abbildung oben*).
5. [Erstellen Sie Inhaltsverzeichnisreferenzen](#). Definieren Sie in der Inhaltsverzeichnis-Vorlage für jede Ebene eine Inhaltsverzeichnisreferenz. Jede Inhaltsverzeichnisreferenz referenziert die entsprechenden Inhaltsverzeichnismarkierungen innerhalb dieser Dokumentebene anhand ihres Namens. Alternativ dazu kann die Inhaltsverzeichnisreferenz zusätzlich Inhaltsverzeichnismarkierungen auf anderen Ebenen referenzieren.
6. [Formatieren Sie die Einträge des Inhaltsverzeichnisses](#). Jeder Inhaltsverzeichniseintrag (in der Inhaltsverzeichnisvorlage) kann eine Nummerierung (auch eine hierarchische), den Text des Inhaltsverzeichniseintrags, eine punktierte Linie und bei paginierten Medien eine Seitennummer enthalten. Jeder Inhaltsverzeichniseintrag sowie seine Bestandteile können je nach Wunsch formatiert werden. Beachten Sie: Sie können die Nummerierung nicht nur in der Inhaltsverzeichnis-Vorlage, sondern auch innerhalb einer Inhaltsverzeichnismarkierung im Hauptteil des Dokuments inkludieren.

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 Ebenen](#). Eine 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 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 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](#) erläutert. [Eine-Inhaltsverzeichnis-Vorlage](#) 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.

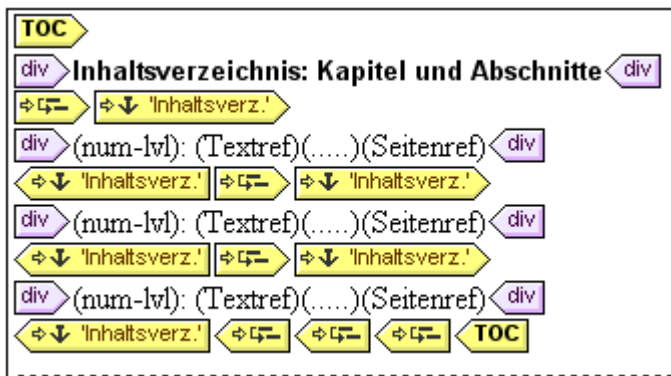
2. [Erstellen einer Inhaltsverzeichnismarkierung](#) im Design mit einem Namen und Text für den Inhaltsverzeichniseintrag. Die Inhaltsverzeichnismarkierung kann entweder eine Design-Komponente umschließen oder keine Design-Komponente umschließen; im zweiten Fall ist die Markierung leer. In der Abbildung 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](#).

Wie werden markierte Einträge in der Inhaltsverzeichnis-Vorlage referenziert?

Die [Inhaltsverzeichnis-Vorlage](#) 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 Ebene referenziert Inhaltsverzeichnismarkierungen anhand ihres Namens. Jede Inhaltsverzeichnismarkierung mit diesem Namen, die sich im XML-Dokument innerhalb der entsprechenden Ebene befindet, wird im Inhaltsverzeichnis auf dieser Ebene als Inhaltsverzeichniseintrag angelegt (wenn als Geltungsbereich der Inhaltsverzeichnis-Referenz die aktuelle Ebene definiert wurde). 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 Inhaltsverzeichnismarkierung wird als Text des Inhaltsverzeichniseintrags ausgegeben.



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 `Inhaltsverz.`  haben. Dadurch werden alle Inhaltsverzeichnismarkierungen auf der ersten Ebene (entsprechend der Struktur des Designs) mit dem Namen `Inhaltsverz.` für die Ausgabe im Inhaltsverzeichnis aufgerufen. Die Inhaltsverzeichnisreferenz auf der zweiten Ebene referenziert ebenfalls Inhaltsverzeichnismarkierungen mit dem Namen `Inhaltsverz.`. Dadurch werden alle Inhaltsverzeichnismarkierungen auf der zweiten Ebene (entsprechend der Struktur des Designs) mit dem Namen `Inhaltsverz.` für Elemente auf der zweiten Ebene des Inhaltsverzeichnisses verwendet. Die dritte Ebene funktioniert auf dieselbe Art: Inhaltsverzeichnismarkierungen mit dem Namen `Inhaltsverz.`, 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](#) und (ii) [Erstellung von Inhaltsverzeichnismarkierungen](#). Eine Beschreibung zum [Erstellen der Inhaltsverzeichnis-Vorlage](#) finden Sie im Abschnitt [Erstellen der Inhaltsverzeichnis-Vorlage](#).

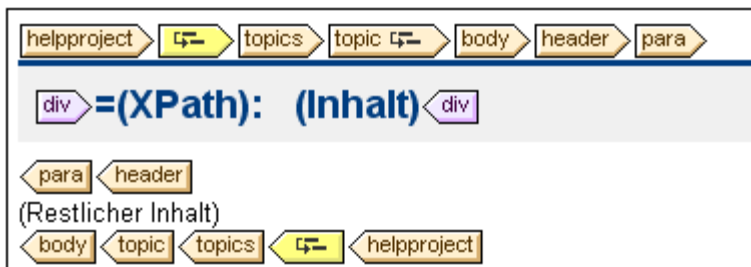
Strukturieren des Designs in Ebenen

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 im Dokument-Design erstellt werden.

Ebenen können in 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 hierarchische Struktur für die Ausgabe des SPS definieren müssen.

Erstellen von Ebenen

Jede Ebene wird separat erstellt. Im Design-Dokument können Ebenen in einer Vorlage oder rund um eine Komponente erstellt werden. In der Abbildung unten wurde eine Ebene in der Vorlage `topic` erstellt (gekennzeichnet durch ) und eine weitere Ebene rund um das Element `topics` (gekennzeichnet durch  ). Der entscheidende Unterschied zwischen diesen beiden Arten der Ebenenerstellung ist, dass mit der Option zum Einschließen innerhalb einer Ebene   Ebenen rund um Komponenten erstellt werden können, bei denen es sich nicht um Vorlagen handelt.

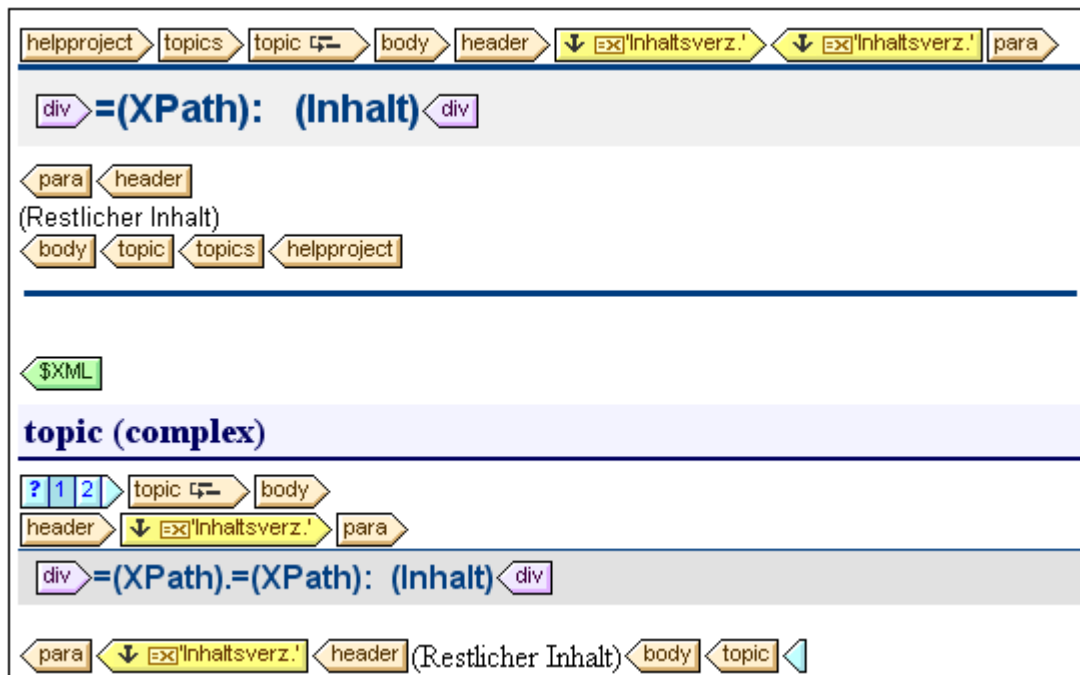


So erstellen Sie eine Ebene:

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 | Ebene**. Beide Optionen stehen auch im Menü **Einfügen | Inhaltsverzeichnis einfügen** zur Verfügung: **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 Hauptvorlage wird durch den Tag `$XML` gekennzeichnet) wurde eine Ebene für die erste 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 Inhaltsverzeichnis-Hierarchie, welche auf der zweiten Ebene von `topic` Elementen definiert ist, ist in die erste Ebene der Inhaltsverzeichnis-Hierarchie 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 Inhaltsverzeichnis-Hierarchie zusätzliche verschachtelte Ebenen 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.

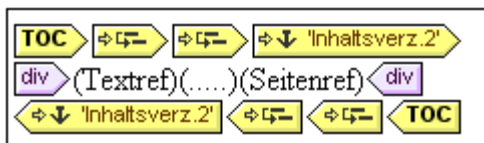
Ebenen in einer flachen Inhaltsverzeichnis-Hierarchie

In einer flachen Inhaltsverzeichnis-Hierarchie werden die Inhaltsverzeichniseinträge als eine einzige Ebene ausgegeben; Das Inhaltsverzeichnis wird in diesem Fall in Form einer einfachen Liste von Einträgen erstellt. In der Inhaltsverzeichnis-Vorlage werden die aufzulistenden Einträge wie üblich im Design-Dokument referenziert: nach Namen und nach der Ebene, in der sie vorkommen. Das Design-Dokument muss daher mindestens eine Ebene enthalten und diese Ebene muss alle erforderlichen Inhaltsverzeichnismarkierungen enthalten.

Wenn das Design mehrere Ebenen enthält und ein flaches Inhaltsverzeichnis z.B. für die zweite Ebene erstellt werden soll, könnte die Inhaltsverzeichnis-Vorlage zwei Ebenenreferenzen mit einer Inhaltsverzeichnisreferenz innerhalb der zweiten Ebene haben (siehe Abbildung unten).

Betrachten Sie z.B. das Design-Dokument in der Abbildung oben: Es hat eine Ebene in der

`topic` -Vorlage in der Hauptvorlage und Unterebenen in der `topic` -Vorlage in der globalen Vorlage. Die Inhaltsverzeichnis-Vorlage in der Abbildung unten erzeugt ein flaches Inhaltsverzeichnis bestehend aus den Themenüberschriften der zweiten Ebene (vorausgesetzt der Name des Inhaltsverzeichnis-Markierung ist Inhaltsverz.).



Dies kommt daher, weil die Inhaltsverzeichnisreferenz in der Inhaltsverzeichnis-Vorlage mit "toc" benannte Inhaltsverzeichnismarkierungen referenziert, die sich auf der zweiten Ebene befinden. Beachten Sie, dass das Element Inhaltsverzeichnis-Referenz in der Inhaltsverzeichnis-Vorlage innerhalb der zweiten Ebenenreferenz der Inhaltsverzeichnis-Vorlage erstellt wurde. Da nur eine Ebene ausgegeben wird (es gibt keine Ausgabe für die erste Ebene), handelt es sich beim ausgegebenen Inhaltsverzeichnis um ein flaches Inhaltsverzeichnis.

Anmerkung: 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.

Erstellen von Inhaltsverzeichnismarkierungen

Inhaltsverzeichnismarkierungen werden im Dokument-Design innerhalb einer [Inhaltsverzeichnisebene](#) erstellt. Sie können in der Hauptvorlage und/oder in globalen Vorlagen erstellt werden. Eine Inhaltsverzeichnismarkierung dient zwei Zwecken:

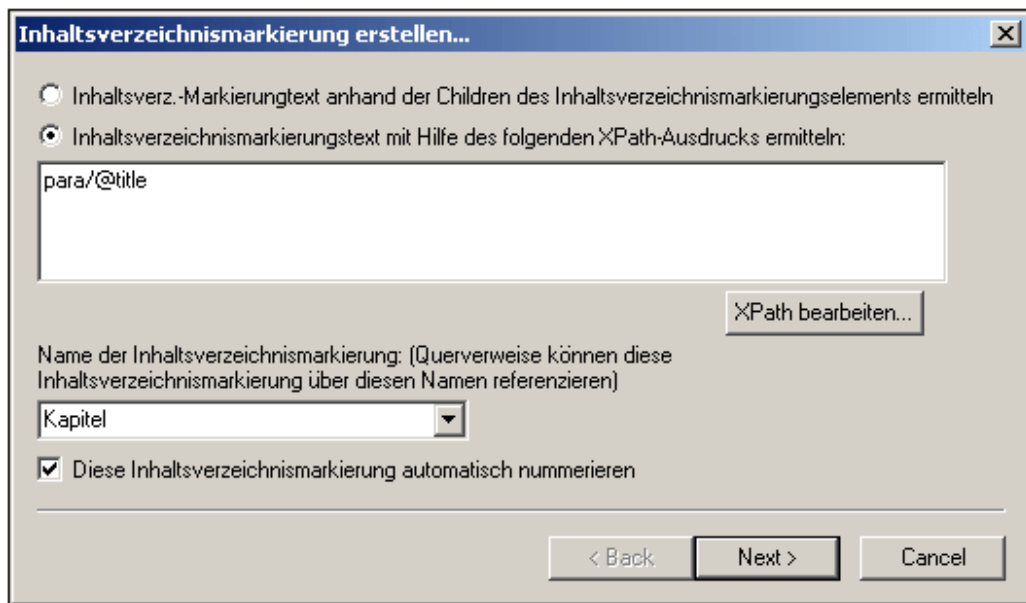
- Sie markiert eine (statische oder dynamische) Komponente im Design mit einem statischen Namen, den Sie ihr zuweisen. Die Markierung kann entweder eine Design-Komponenten umschließen oder nicht umschließen; im zweiten Fall sind die Tags leer. In der Ausgabe wird die Inhaltsverzeichnismarkierung als Anker, der durch seinen Namen identifiziert wird, instantiiert.
- Sie definiert 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 die Inhaltsverzeichnismarkierung auf zwei Arten erstellen: (i) mit Hilfe des Assistenten ["Inhaltsverzeichnismarkierung erstellen"](#), 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; und (ii) durch [Einfügen einer leeren Inhaltsverzeichnismarkierung](#), deren Eigenschaften Sie erst später definieren.

Erstellen von Inhaltsverzeichnismarkierungen mit Hilfe des Assistenten

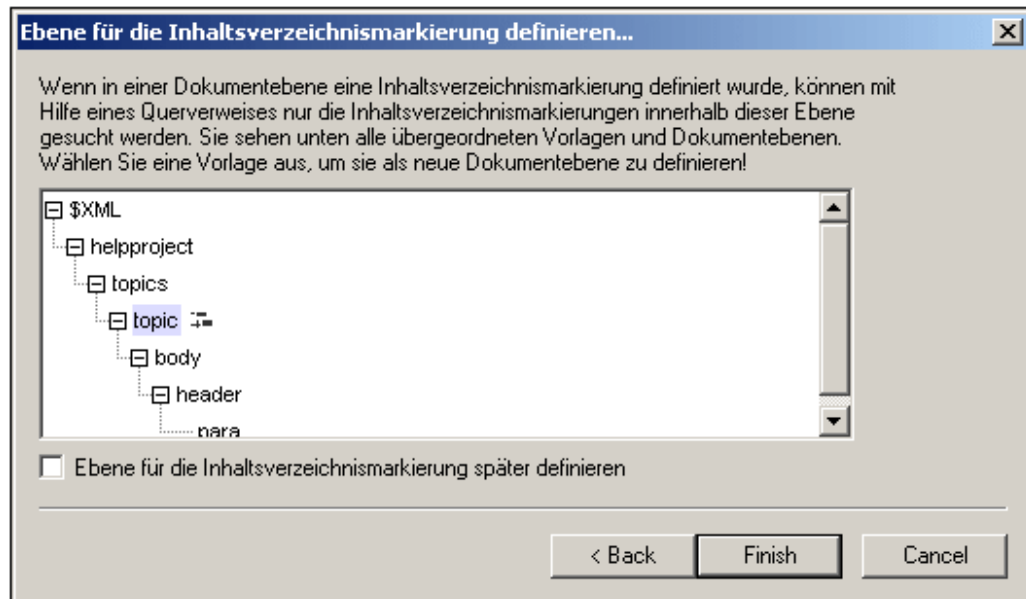
So erstellen Sie eine Inhaltsverzeichnismarkierung mit Hilfe des entsprechenden Assistenten:

1. Platzieren Sie den Cursor an die Stelle im Design, an der die Inhaltsverzeichnismarkierung eingefügt werden soll oder wählen Sie eine Design-Komponente, rund um die die Inhaltsverzeichnismarkierung eingefügt werden soll.
2. Wählen Sie im Menü **Einfügen** die Option **Inhaltsverzeichnis einfügen | Inhaltsverz.-Markierung (Assistent)**. Daraufhin wird der Assistent zum Erstellen der Inhaltsverzeichnismarkierung aufgerufen (*Abbildung unten*).



1. Im ersten Fenster des Assistenten (*Abbildung oben*) definieren Sie (i) den Texteintrag 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 das Ergebnis eines XPath-Ausdrucks verwendet werden soll. Beim Namen der Markierung können Sie Text direkt eingeben oder aus einer Dropdown-Liste mit den Namen der bereits definierten Markierungsnamen auswählen. Klicken Sie anschließend auf **Weiter**.
2. Im zweiten Fenster des Assistenten (*Abbildung unten*) können Sie die Ebene auswählen, innerhalb der die Inhaltsverzeichnismarkierung eingefügt werden soll.



Übergeordnete Vorlagen, in denen Ebenen zugewiesen wurden, werden durch ein Ebenensymbol gekennzeichnet (in der Abbildung oben hat die Vorlage `topic` eine Ebene). Wählen Sie eine Vorlagenebene aus, innerhalb der die Inhaltsverzeichnismarkierung erstellt werden soll. Wenn für diese Vorlage bereits eine Ebene existiert, wird die Inhaltsverzeichnismarkierung innerhalb dieser Ebene erstellt, andernfalls wird in der ausgewählten Vorlage eine neue Ebene erstellt. Alternativ dazu können Sie die Ebene auch später definieren, indem Sie das Kontrollkästchen "Ebene später definieren" aktivieren. Klicken Sie anschließend auf **Fertig stellen**.

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 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 Befehle der [Bearbeitungsbefehle](#) definieren.

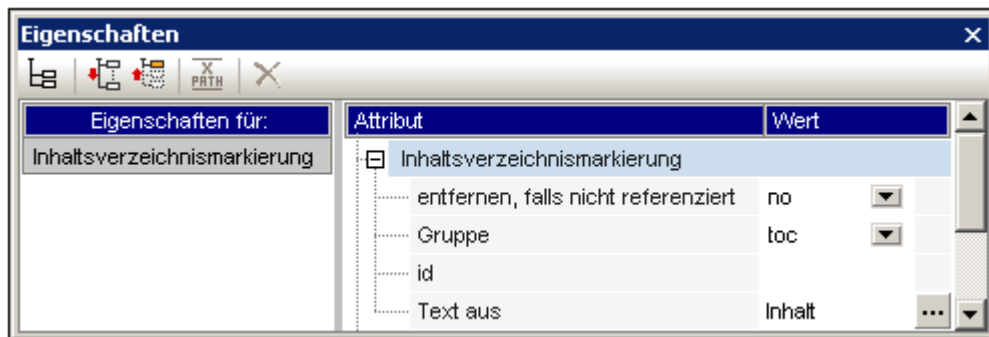
Einfügen einer hierarchischen oder sequenziellen Nummerierung für eine Komponente

Sie können innerhalb der Tags einer Inhaltsverzeichnismarkierung eine hierarchische oder sequenzielle Nummerierung einfügen. Rechtsklicken Sie an die Stelle, an der Sie die Nummerierung einfügen möchten, und wählen Sie **Inhaltsverzeichnis einfügen | Hierarchische / Sequenzielle Nummerierung**. Da eine Nummerierung nur an Stellen innerhalb einer Inhaltsverzeichnismarkierung eingefügt werden kann, empfiehlt es sich, die Inhaltsverzeichnismarkierung rund um eine Komponente zu erstellen, anstelle eine leere anzulegen. Dadurch haben Sie bei der Platzierung der Nummerierung mehr Flexibilität beim

Layout.

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"](#) oder rechtsklicken Sie auf die Inhaltsverzeichnismarkierung und wählen Sie die Eigenschaft aus, die Sie bearbeiten möchten.



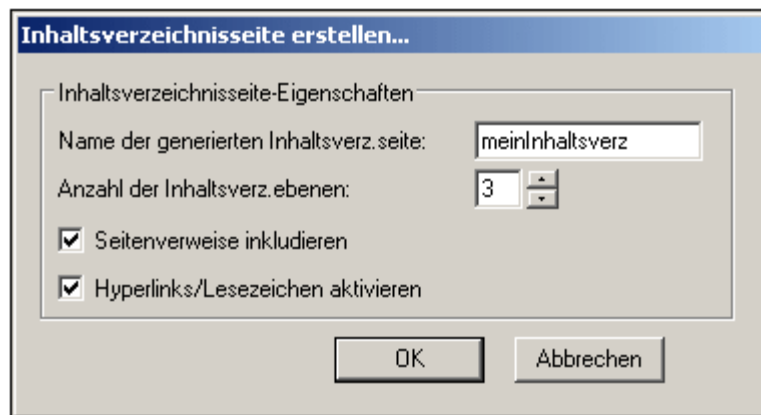
Die Inhaltsverzeichnismarkierung hat die folgenden drei Eigenschaften: (i) eine Option (*Text aus*) zur Definition des Texteintrags, der entweder aus dem Inhalt der Inhaltsverzeichnismarkierung oder aus einem XPath-Ausdruck stammen kann; (ii) den Namen der Gruppe der Inhaltsverzeichnismarkierung (*Gruppe*); und (iii) eine Option zum Entfernen der Markierung, wenn sie nicht referenziert wird.

Erstellen der Inhaltsverzeichnis-Vorlage

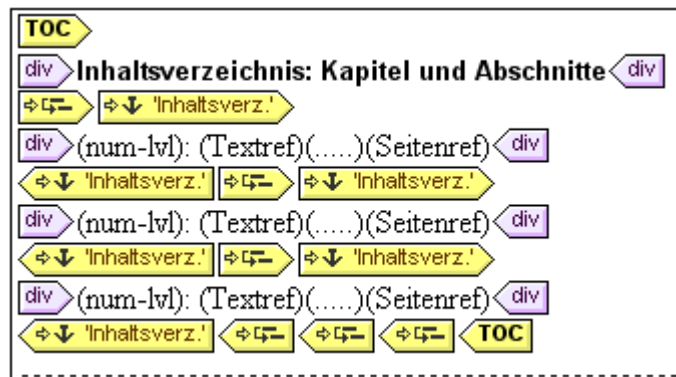
Die Inhaltsverzeichnis-Vorlage ist die Vorlage, anhand der in der Ausgabe das Inhaltsverzeichnis erstellt wird. Sie kann an jeder beliebigen Stelle im SPS-Design angelegt werden. Es können in einem einzigen SPS-Design auch mehrere Inhaltsverzeichnis-Vorlagen erstellt werden.

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](#) im Design-Dokument zu referenzieren. Wenn Sie mehrere Ebenen für das Inhaltsverzeichnis auswählen (nächste Option), wird derselbe Inhaltsverzref-Name auf allen Ebenen verwendet (obwohl einzelne Inhaltsverzref-Namen [später bearbeitet werden können](#)). (ii) Die Anzahl der [Ebenenreferenzen des Inhaltsverzeichnisses](#) 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 Inhaltsverzref mit einem Namen, der Inhaltsverzeichnismarkierungen identifiziert, die für diese Ebenenreferenz des Inhaltsverzeichnisses als Inhaltsverzeichniseinträge verwendet werden sollen. Innerhalb jeder Inhaltsverzref befindet sich eine Standardvorlage für den Inhaltsverzeichniseintrag, die Sie [jederzeit bearbeiten können](#).

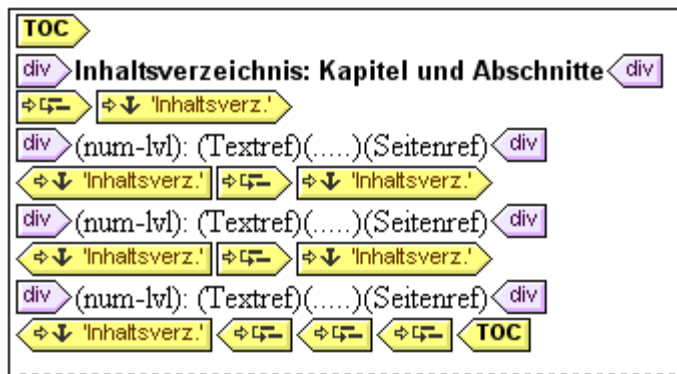
Bearbeiten der Inhaltsverzeichnis-Vorlage

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](#) können zur Struktur der Inhaltsverzeichnis-Vorlage hinzugefügt oder daraus gelöscht werden.
- Die [Eigenschaften einzelner Inhaltsverzeichnisreferenzen](#) (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.
- [Inhaltsverzref](#) können zu jeder Ebenenreferenz in der Inhaltsverzeichnis-Vorlage hinzugefügt bzw. daraus gelöscht werden.
- Der [Inhaltsverzeichniseintrag](#) innerhalb einer Inhaltsverzref kann unter Verwendung der Standard [StyleVision Verfahren](#) 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.

Ebenenreferenzen in der Inhaltsverzeichnis-Vorlage

Die [Inhaltsverzeichnis-Vorlage](#) 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"](#) 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](#) innerhalb eines [festgelegten Geltungsbereiches](#) im SPS Design.

Einfügen und Entfernen von Ebenenreferenzen

Ebenenreferenzen können in die Inhaltsverzeichnis-Vorlage eingefügt oder daraus gelöscht werden, nachdem diese erstellt wurde.

Um eine Ebenenreferenz einzufügen, wählen Sie in der Inhaltsverzeichnis-Vorlage den Inhalt aus, um den herum eine Ebenenreferenz erstellt werden soll und wählen Sie anschließend den Befehl **Einfügen | Inhaltsverzeichnis einfügen | Ebenenreferenz**. Alternativ dazu können Sie auch im Kontextmenü den Befehl **Einschließen in | Ebenenreferenz**. Eine Ebenenreferenz kann auch an der Cursorposition in die Inhaltsverzeichnis-Vorlage eingefügt werden.

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.

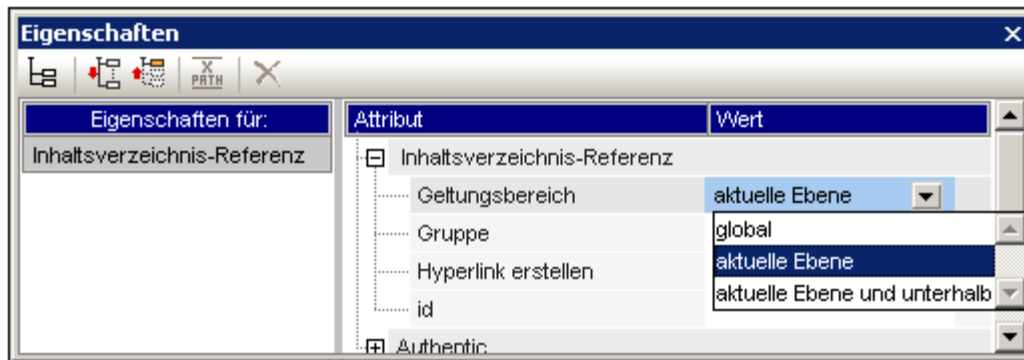
Inhaltsverzeichnisreferenz: Name, Geltungsbereich, Hyperlink

Inhaltsverzeichnisreferenzen (Inhaltsverzref) befinden sich innerhalb von Ebenenreferenzen (Ebenenref) und haben drei Eigenschaften:

- einen **Namen**, der auf Inhaltsverzeichnismarkierungen desselben Namens verweist, die sich im angegebenen Geltungsbereich befinden. Diese sollen als Einträge für diese Ebene in das Inhaltsverzeichnis übernommen werden.
- einen **Geltungsbereich**, mit dem festgelegt wird, für welche Ebenen im SPS Design die Inhaltsverzeichnisreferenz gilt. Es stehen drei Optionen zur Verfügung: global, aktuelle Ebene, aktuelle Ebene und die Ebenen unterhalb davon.
- 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.

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*, *Gruppe bearbeiten* oder *Geltungsbereich bearbeiten*) aus. Daraufhin wird das Fenster "Eigenschaften" angezeigt, in dem die jeweilige Eigenschaft zur Bearbeitung ausgewählt ist (*Abbildung unten*)

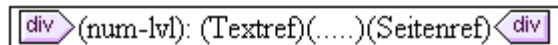


Alternativ dazu können Sie auch direkt zur gewünschten Eigenschaft im Fenster "Eigenschaften" gehen, wenn der Inhaltsverzref-Tag ausgewählt ist (Eigenschaftsgruppe *Inhaltsverz-Referenz*).

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, 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**. Um eine Komponente zu löschen, wählen Sie sie aus und drücken Sie die **Entf**-Taste.

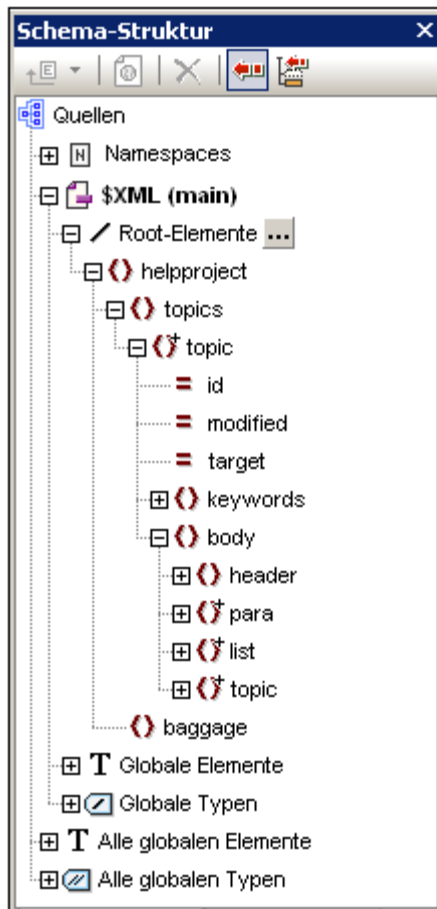
Zusätzlich dazu können Sie statischen Inhalt (z.B. Text) und dynamischen Inhalt (z.B. automatische Berechnungen) in den Inhaltsverzeichniseintrag einfügen.

Formatieren des Inhaltsverzeichniseintrags

Der Inhaltsverzeichniseintrag kann über die [Seitenleiste "Stile"](#) mit [CSS-Stilen](#) formatiert werden. Einzelne Inhaltsverzeichniseintrag-Komponenten können separat formatiert werden. Wählen Sie dazu die Komponente aus und weisen Sie ihr in der Seitenleiste "Stile" [Stileigenschaften](#) zu.

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](#), `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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](#) 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 zwei Schritten:

1. [Gliedern des Designs in Inhaltsverzeichnis-Ebenen](#): In das Design-Dokument werden eine oder mehrere Ebenen eingefügt, um das Ausgabedokument hierarchisch zu gliedern. Diese hierarchische Struktur entspricht der des Inhaltsverzeichnisses. In

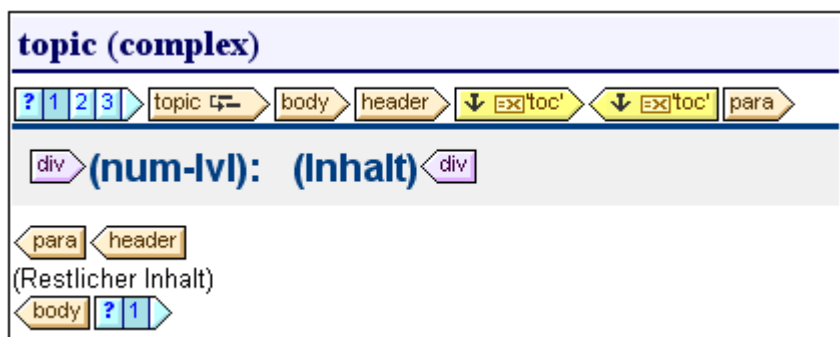
unserem aktuellen Beispiel wurde eine einzige Ebene erstellt - in der Vorlage `topic`. Infolgedessen gibt es im Design-Dokument nur eine Ebene und das Inhaltsverzeichnis kann in der Gliederung nur eine Ebene enthalten.

2. [Erstellen von Inhaltsverzeichnis-Markierungen](#): Innerhalb der in Schritt 1 (im Design-Dokument) erstellten Ebene wird eine Inhaltsverzeichnis-Markierung angelegt. Mit Hilfe dieser Markierung können Inhaltsverzeichnis-Referenzen in der Inhaltsverzeichnis-Vorlage auf diese Markierung verweisen. Die Inhaltsverzeichnis-Markierung enthält den Text, der im Inhaltsverzeichnis-Eintrag, der darauf verweist, angezeigt wird.
3. [Erstellen der Inhaltsverzeichnis-Vorlage](#): Dies ist die Vorlage, mit der das Inhaltsverzeichnis im Dokument erstellt wird. Sie ist in Ebenen-Referenzen (reflevels) gegliedert, die der Struktur der Ebenen im Design-Dokument entsprechen müssen. Wenn die Inhaltsverzeichnisvorlage z.B. drei verschachtelte reflevels enthält, so muss das Design-Dokument mindestens drei Ebenen haben. In diesem Beispiel haben wir ein einziges reflevel, das der einzigen Ebene im Design-Dokument entspricht. Es befindet sich innerhalb des reflevels, in das 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 Design, außerhalb der Hauptvorlage und dahinter besteht aus globalen Vorlagen.

Die Inhaltsverzeichnis-Definitionen (Inhaltsverzeichnis-Ebenen und 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 von Abschnitten und Unterabschnitten getrennt werden. Jede Art von `topic` Element (Kapitel, Abschnitt, Unterabschnitt) wird in eine andere Bedingungsverzweigung gesetzt, damit es anders verarbeitet wird als die anderen Arten von `topic` Elementen.



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 Ebene erstellt wurde. In den beiden anderen Bedingungsverzweigungen wurden in der `topic`-Vorlage keine Ebenen erstellt. Als Ergebnis davon wurde dem Dokument eine Ebene zugewiesen und zwar auf der Ebene des obersten `topic` Elements (vom Typ "Kapitel").

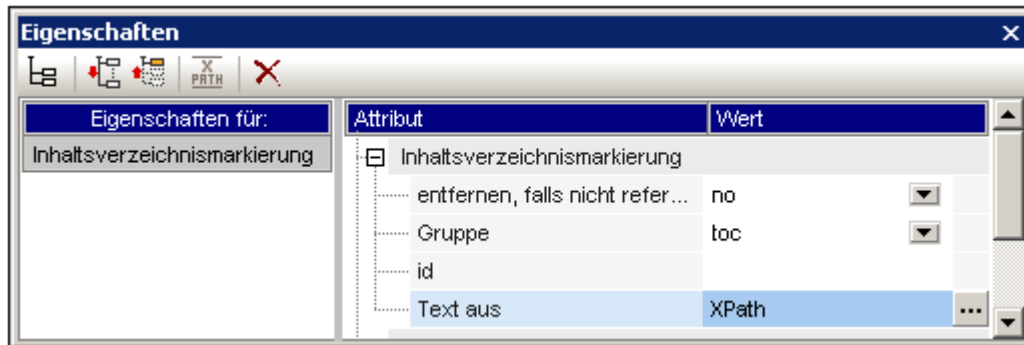
Erstellen der Inhaltsverzeichnis-Markierung

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 Gruppeneigenschaft wird die Gruppe Inhaltsverzeichnismarkierung definiert. In unserem Beispiel haben wir für diese Eigenschaft den Wert `toc` 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.

TOC-Vorlage

In die Inhaltsverzeichnis-Vorlage (Abbildung unten) wurde eine einzige Ebenenreferenz (reflevel) eingefügt. Dieses `reflevel` entspricht der Inhaltsverzeichnis-Ebene, die dem Kapitel-`topic` Element der ersten Ebene im Design zugewiesen wurde (siehe weiter oben 'SPS-Struktur und -Ebenen').

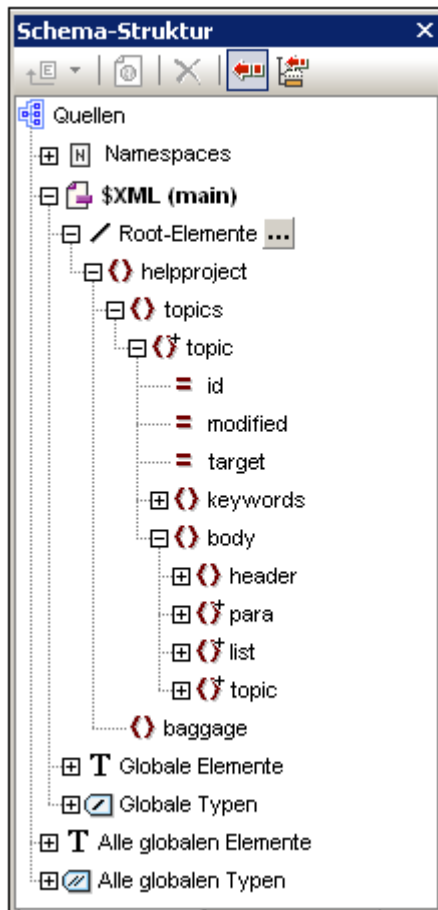


In dieses `reflevel` wurde eine Inhaltsverzeichnis-Referenz (TOCref) eingefügt. Diese TOCref dient zum Auswählen von Markierungen (i) die sich in der Markierungsgruppe `toc` (siehe 'Erstellen der Inhaltsverzeichnis-Markierung' weiter oben) befinden und (ii) sich ausschließlich im Geltungsbereich der 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 definiert. 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.

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](#), `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/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.



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:

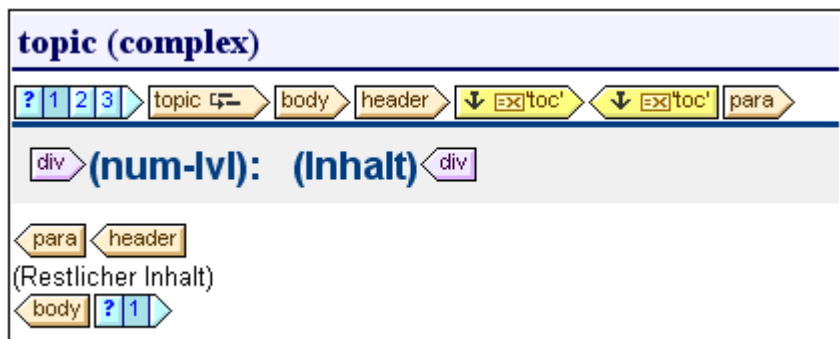
1. [Kapitelübersicht](#) (*Chapters at a Glance*): Listet die Namen der einzelnen Kapitel auf (die Kapitel auf der obersten Ebene).
2. [Kapitel und ihre Abschnitte](#) (*Chapters and Their Sections*): Listet die einzelnen Kapitel mit ihren untergeordneten Abschnitten auf (Kapitel auf oberster Ebene plus 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](#) (*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 Ebene erstellt und innerhalb des Child-Elements `header` (aber außerhalb des Elements `para`) wurde eine Markierung erstellt.



Da das Element `topic` rekursiv ist, sind auch die Ebenen und Markierungen rekursiv, d.h. es werden für jedes untergeordnete Kapitel eine neue hierarchisch untergeordnete Ebene und eine neue Markierung erstellt. Da die Formatierung der Überschrift (der Titel des Kapitels) für die einzelnen Ebenen unterschiedlich sein soll, haben wir die einzelnen Ebenen in eine separate Verzweigung einer Bedingung mit drei Verzweigungen eingeschlossen. Jede Verzweigung überprüft, auf welcher Ebene ein Kapitel vorkommt: der ersten, der zweiten oder der dritten.

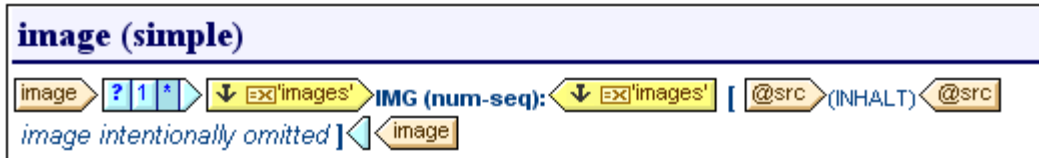
Beachten Sie, dass innerhalb der Ebenen eine hierarchische Nummerierung (`num-lvl`) eingefügt wurde. Dies erfolgt durch Rechtsklick an die gewünschte Stelle und Auswahl des Befehls **Inhaltsverzeichnis einfügen | Hierarchische Nummerierung**. Dadurch wird vor dem Titel jedes einzelnen Kapitels im **Textfluss des Dokuments** die korrekte hierarchische Nummerierung eingefügt, z.B. 3. 1 oder 4. 2. 3.

TOC-Beschreibungen

Im Folgenden finden Sie eine kurze Beschreibung der einzelnen Inhaltsverzeichnisse und die dabei zu beachtenden Punkte.

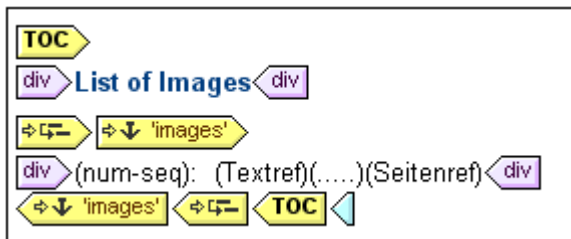
Kapitelübersicht: Wählen Sie die Inhaltsverzeichnis-Markierung in der globalen Vorlage für `topic` aus. Beachten Sie in der Seitenleiste "Eigenschaften" (*Abbildung unten*), dass der Eingabetext laut Definition mit Hilfe eines XPath-Ausdrucks erstellt wird. Wenn Sie im Wertefeld **Text aus** auf die 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 Markierung innerhalb des Elements `header` eingefügt wurde).

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 auf einer einzigen Referenzebene. 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 Referenzebene 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. Diese Nummern können auch formatiert werden.

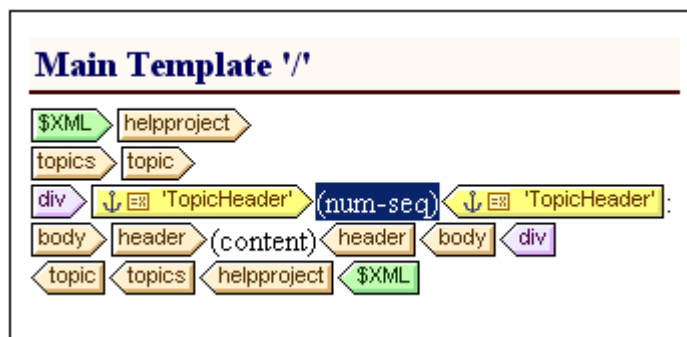
Automatische Nummerierung

Sich wiederholende Node-Instanzen können mit Hilfe der StyleVision-Funktion "Automatische Nummerierung" automatisch nummeriert werden. So kann z.B. in einem Element `Book`, das mehrere `Chapter`-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. Die automatische Nummerierung kann flach oder hierarchisch sein und es stehen die verschiedensten Formatierungsoptionen für die Nummerierung zur Verfügung.

Flach (sequenzielle Nummerierung)

Eine flache Nummerierung kann in eine [Inhaltsverzeichnisreferenz](#) oder eine [Inhaltsverzeichnismarkierung](#) eingefügt werden. Die flache Nummerierung in einer Inhaltsverzeichnisreferenz verweist zurück auf eine Markierung im Dokument; es würde sich hierbei um einen Inhaltsverzeichniseintrag handeln und wird im [Abschnitt "Inhaltsverzeichnis"](#) beschrieben. In diesem Abschnitt wird beschrieben, wie man innerhalb einer Inhaltsverzeichnisreferenz im Dokument eine flache Nummerierung erstellt. Eine derartige Nummerierung kann unabhängig von einem Inhaltsverzeichnis sein. Um eine flache Nummerierung in einem Dokument zu erstellen, gehen Sie 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.
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 flache (sequenzielle) Nummerierung (`num-seq`) eingefügt (in der Abbildung unten in der *Inhaltsverz.-Markierung 'TopicHeader' markiert*).



3. Klicken Sie mit der rechten Maustaste auf die Inhaltsverzeichnismarkierung und deaktivieren Sie den Befehl **Eingabetext mittels XPath erstellen**, da: (i) die Inhaltsverzeichnismarkierung nur für die flache Nummerierung und nicht für Inhaltsverzeichniseinträge verwendet wird; kein Texteintrag für die Inhaltsverzeichniseinträge erforderlich ist; und (ii) damit sichergestellt wird, dass kein fehlerhafter XPath-Ausdruck, der einen Transformationsfehler verursachen kann, verwendet wird.
4. Die Inhaltsverzeichnismarkierung benötigt keinen Namen (da sie von der Inhaltsverzeichnisvorlage aus nicht referenziert wird). Sie können ihr auch einen Namen geben, wenn Sie möchten (klicken Sie mit der rechten Maustaste auf die Inhaltsverzeichnismarkierung und wählen Sie den Befehl **Namen bearbeiten**).

In dem Beispiel in der Abbildung oben wurde die flache 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.

```

1: Altova StyleVision 2007
2: About this Documentation
3: Introduction
4: User Interface

```

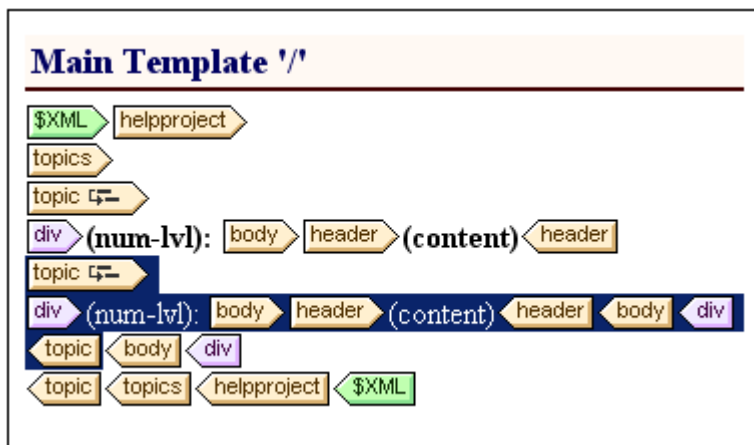
Anmerkung: Wenn die flache Nummerierung an einer anderen Gruppe von Nodes fortgesetzt werden soll, verwenden Sie eine Inhaltsverzeichnismarkierung, die denselben Namen hat, wie diejenige, an die angeschlossen werden soll.

Hierarchische Nummerierung

Eine hierarchische Nummerierung kann in eine [Referenzebene](#) oder eine [Ebene im Design](#) eingefügt werden. In einer Referenzebene verweist die hierarchische Nummerierung zurück auf eine Inhaltsverzeichnismarkierung im Dokument; es würde sich hierbei um einen Inhaltsverzeichniseintrag handeln und wird im [Abschnitt "Inhaltsverzeichnis"](#) behandelt. In diesem Abschnitt wird beschrieben, wie eine hierarchische Nummerierung innerhalb von Ebenen im Dokument erstellt wird. Eine solche hierarchische Nummerierung kann unabhängig vom Inhaltsverzeichnis sein.

Um eine hierarchische Nummerierung in einem Dokument zu erstellen, müssen Sie das Dokument zuerst in Ebenen strukturieren (siehe Abschnitt [Strukturieren des Designs in Ebenen](#)). 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*).



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 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 | Sequenzielle Nummerierung**).

Des Ergebnis des Designs in der Abbildung oben sieht folgendermaßen aus.

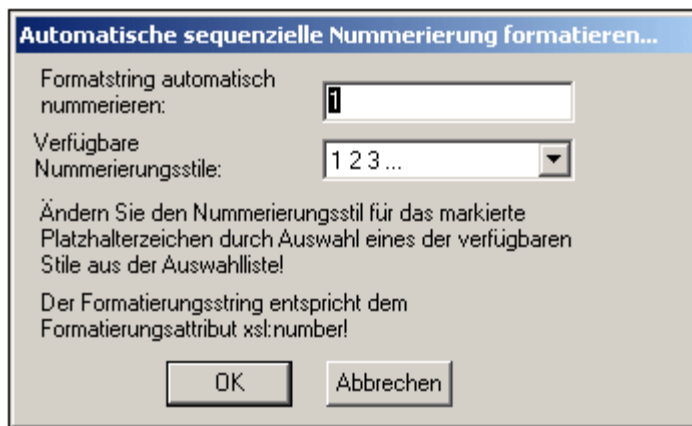
1: Altova StyleVision 2007
2: About this Documentation
3: Introduction
3.1: What Is an SPS?
3.2: Product Features
3.3: Setting up StyleVision
4: User Interface
4.1: Main Window
4.2: Design Entry Helpers

Die erste Ebene wird fett gedruckt, die zweite normal.

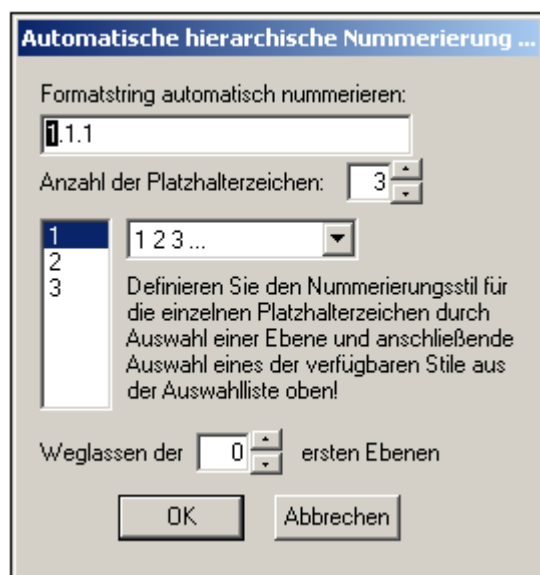
Formatierung

Zur Formatierung der automatischen Nummerierung klicken Sie mit der rechten Maustaste auf den Platzhalter für die automatische Nummerierung (`( num-seq )` oder `( num-lvl )`) und wählen Sie den Befehl "Format bearbeiten". Daraufhin wird das jeweilige Dialogfeld angezeigt (*Abbildungen unten*), in dem Sie die gewünschte Formatierung aus den verschiedensten Optionen auswählen können.

- Sequenzielle Nummerierung: Wählen Sie einen Nummerierungsstil aus. Ihre Auswahl wird im Feld "Formatstring..." angezeigt und kann gegebenenfalls geändert werden.



- Hierarchische Nummerierung: Wählen Sie zuerst den Nummerierungsstil aus und dann die Anzahl der Platzhalterzeichen. Das Ergebnis wird im Feld "Formatstring..." angezeigt. Ebenen können weggelassen werden, indem Sie in das Feld "Weglassen der...ersten Ebenen" eingeben, wieviele Ebenen weggelassen werden sollen.



Klicken Sie auf **OK**, wenn Sie damit fertig sind.

Textreferenzen

Auf Nodes in einem Dokument können Ankerpunkte erstellt werden und mit dynamischen Namen versehen werden. Diese Ankerpunkte können anschließend über ihre dynamischen Namen referenziert werden. Das bedeutet, dass Text für die Referenzierung markiert und anschließend von einer anderen Stelle im Dokument aus referenziert werden kann.

Auf der Benutzeroberfläche werden diese Ankerpunkte mit dynamischen Namen (also die Textreferenzen) mit Hilfe von Inhaltsverzeichnismarkierungen erstellt, die den zu referenzierenden Text mit Hilfe von XPath-Ausdrücken dynamisch auffinden können. Das Design kann dann Inhaltsverzeichnisreferenzen enthalten, die die gewünschten Inhaltsverzeichnismarkierungen anhand ihres Namens erkennen. Auf diese Art identifiziert die Inhaltsverzeichnisreferenz die Textreferenz und erstellt eine Verknüpfung zu dieser.

Inhaltsverzeichnisreferenzen können innerhalb von Ebenen und innerhalb von Referenzebenen gefunden werden. Zweitere Option wird in Inhaltsverzeichnissen verwendet und ist im Abschnitt [Inhaltsverzeichnis](#) beschrieben. In diesem Abschnitt beschreiben wir, wie Referenzen innerhalb von Ebenen im Design-Dokument erstellt werden, sodass diese Referenzen als Querverweise verwendet werden können.

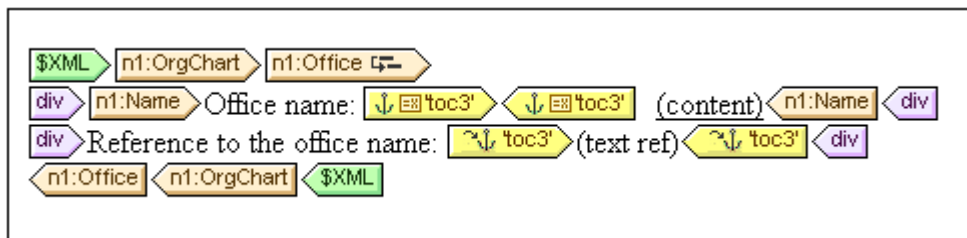
Schritt 1: Ebenen

Das Dokument ist, wie im Abschnitt [Strukturieren des Designs in Ebenen](#) beschrieben, in Ebenen gegliedert. Die Ebenen dienen bei der Referenzierung dazu, den Geltungsbereich der Referenzierung zu definieren. In der Abbildung unten wurde auf dem Element `n1:Office` eine Ebene erstellt.

Schritt 2: Erstellen von Inhaltsverzeichnismarkierungen

Innerhalb einer Ebene wird eine Inhaltsverzeichnismarkierung mit einem Namen und einem XPath-Ausdruck zum Identifizieren des Node im Dokument, bei dessen Inhalt es sich um den zu referenzierenden Text handelt, erstellt. In der Abbildung unten hat die

Inhaltsverzeichnismarkierung im Element `n1:Name` den Namen `toc3` und einen XPath-Ausdruck, mit dem der aktuelle Node gesucht wird. D.h. dass es sich bei der Textreferenz um den Inhalt des Node `n1:Name` handelt.



Bei der Verarbeitung des XML-Dokuments wird für jedes `n1:Office/n1:Name`-Element ein Anker mit einer Textreferenz erstellt, die der Wert des Elements `n1:Office/n1:Name` ist.

Eine Inhaltsverzeichnismarkierung wird durch Platzieren des Cursors an die gewünschte Stelle, Rechtsklick und Auswahl des Befehls **Inhaltsverzeichnis einfügen | Inhaltsverz.-Markierung** eingefügt.

Schritt 3: Erstellen von Inhaltsverzeichnisreferenzen

Eine Inhaltsverzeichnisreferenz wird eingefügt (Kontextmenü, **Inhaltsverzeichnis einfügen | Inhaltsverz.-Referenz**), um eine Verknüpfung zu Textreferenzen herzustellen, die mit einer Inhaltsverzeichnismarkierung generiert wurden. In der Abbildung oben befindet sich die

ausgewählte Stelle von  (text ref)  innerhalb derselben Ebene wie die, in der die Inhaltsverzeichnismarkierung erstellt wurde (in der Ebene *Office*). Beim Definieren der Inhaltsverzeichnisreferenz werden zwei Dinge definiert: Erstens der Name der Inhaltsverzeichnismarkierung, auf die verwiesen wird; in der Abbildung oben ist der Name *toc3*. Zweitens der Referenzierungsbereich; im Beispiel oben ist der Bereich die aktuelle Ebene. Das bedeutet, dass diese Referenz Inhaltsverzeichnismarkierungen innerhalb der aktuellen Ebene referenziert.

Die Ausgabe sieht in etwa folgendermaßen aus:

Office name: Nanonull, Inc.
Reference to the office name: Nanonull, Inc.
Office name: Nanonull Europe, AG
Reference to the office name: Nanonull Europe, AG

Der violette Text wird vom Platzhalter der Inhaltsverzeichnisreferenz (text ref) generiert. Der Inhalt der Textreferenz wird vom XPath-Ausdruck in der durch die Inhaltsverzeichnisreferenz referenzierten Inhaltsverzeichnismarkierung ermittelt.

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.

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-URLs für Hyperlinks.

In diesem Abschnitt wird beschrieben:

- wie [Lesezeichen](#) in das SPS eingefügt werden können.
- wie [Hyperlinks](#) 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.

Einfügen von Lesezeichen

Ein Lesezeichen (oder Anker) kann überall im SPS eingefügt werden - an einem Cursor-Einfügepunkt oder rund um eine SPS-Komponente.

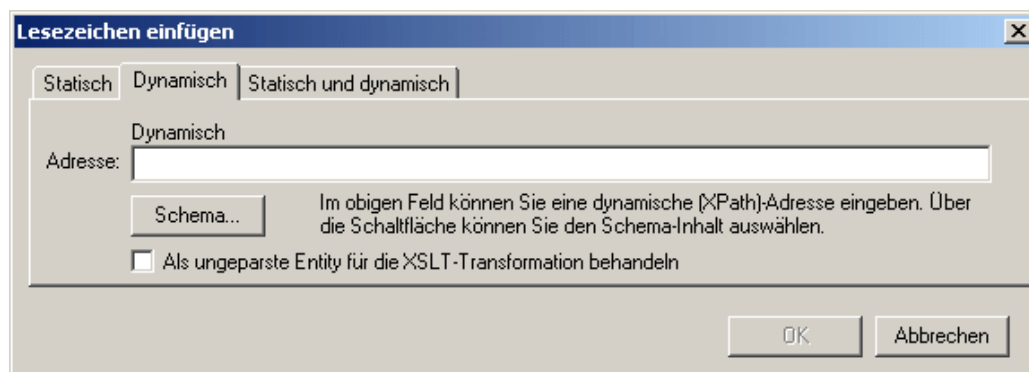
Lesezeichen im SPS werden über das Dialogfeld "Lesezeichen einfügen" (*Abbildung unten*) angelegt. In diesem Dialogfeld wird der Name des Lesezeichens definiert. Dabei kann es sich um einen statischen Namen oder einen dynamischen Namen, der (i) aus dem Inhalt des XML-Dokuments stammt oder (ii) mit Hilfe eines XPath-Ausdrucks erstellt wurde, handeln.

Lesezeichen werden an der im Design definierten Stelle erstellt. Wenn sich das Element, in dem sich das Lesezeichen befindet, wiederholt, so wird in jeder Instanz des sich wiederholenden Elements ein Lesezeichen erstellt. Wenn Sie dem Lesezeichen einen statischen Namen geben, so erhält jedes dieser Lesezeichen denselben Namen. Daher empfiehlt es sich, dem Lesezeichen in solchen Fällen (bei sich wiederholenden Elementen) einen dynamischen Namen zu geben. Dabei kann es sich z.B. um den Inhalt 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 unten*) sicherstellen, dass das Lesezeichen einen eindeutigen Namen hat. Zur Referenzierung des Lesezeichens kann dieselbe ID auch als `href`-Wert eines [Hyperlink](#) generiert werden.

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](#) oder klicken Sie auf die rechte Maustaste und wählen Sie **Einfügen | Lesezeichen**.
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](#)

Hinweis:

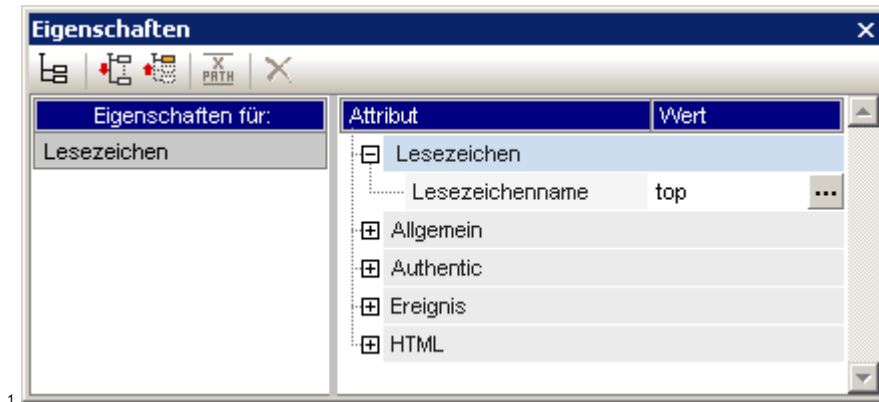
Ein Lesezeichen wird für eine einzelne fixe Stelle im Design angelegt. Wenn sich das

Element, in dem sich das Lesezeichen befindet, wiederholt, so wird in jeder Instanz des sich wiederholenden Elements ein Lesezeichen erstellt. Wenn Sie dem Lesezeichen einen statischen Namen geben, so erhält jedes dieser Lesezeichen denselben Namen. Daher empfiehlt es sich, dem Lesezeichen 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](#) generiert werden. Verwenden Sie in diesem Fall unbedingt den Fragment-Identifizier `#` 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 *Lesezeichennamen* (*Abbildung unten*). Daraufhin wird dasselbe Dialogfeld "Lesezeichen bearbeiten" (siehe *Abbildung oben*) angezeigt, das oben beschrieben wurde.



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.

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.

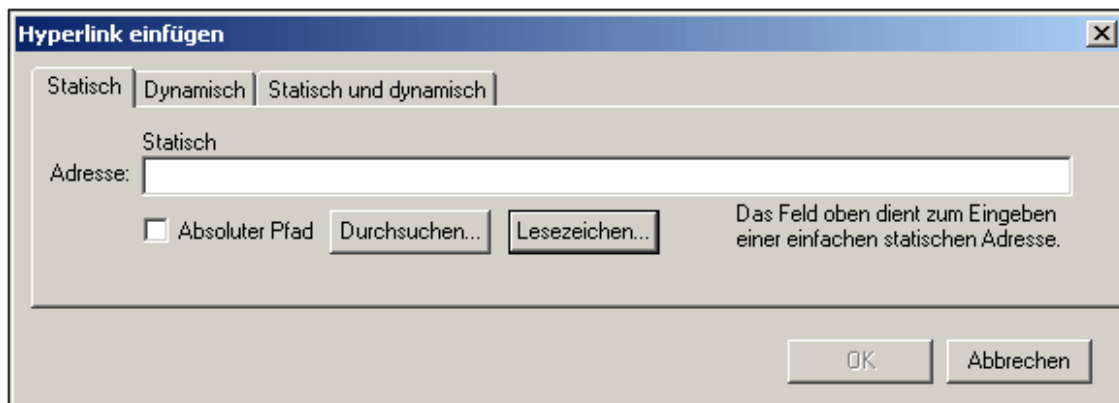
Erstellen von Hyperlinks

Ein Hyperlink kann auf folgende Arten erstellt werden:

- Rund um Text (statischen oder dynamischen), Nodes, Bilder, Bedingungsvarianten, 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 Bedingungsvariante 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](#) oder [Einfügen | Hyperlink](#) aufgerufen wird, definiert.



Ein Hyperlink kann

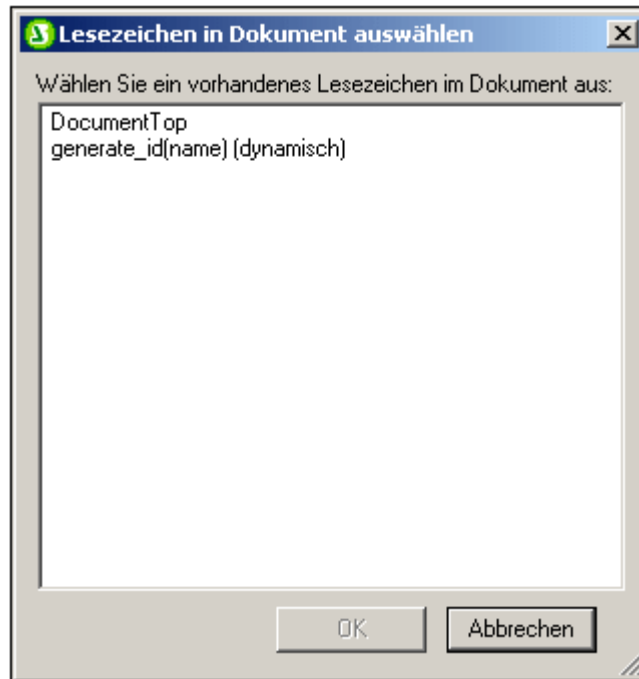
- auf ein [Lesezeichen](#) im selben SPS Design verweisen (in diesem Fall muss die Ziel-URI ein Fragment Identifier sein),
- [dynamisch generiert werden](#), um Lesezeichen-Markierungen zu finden (bei diesen URIs handelt es sich ebenfalls um Fragment Identifier),
- eine [externe Ressource](#) sein; die 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**.

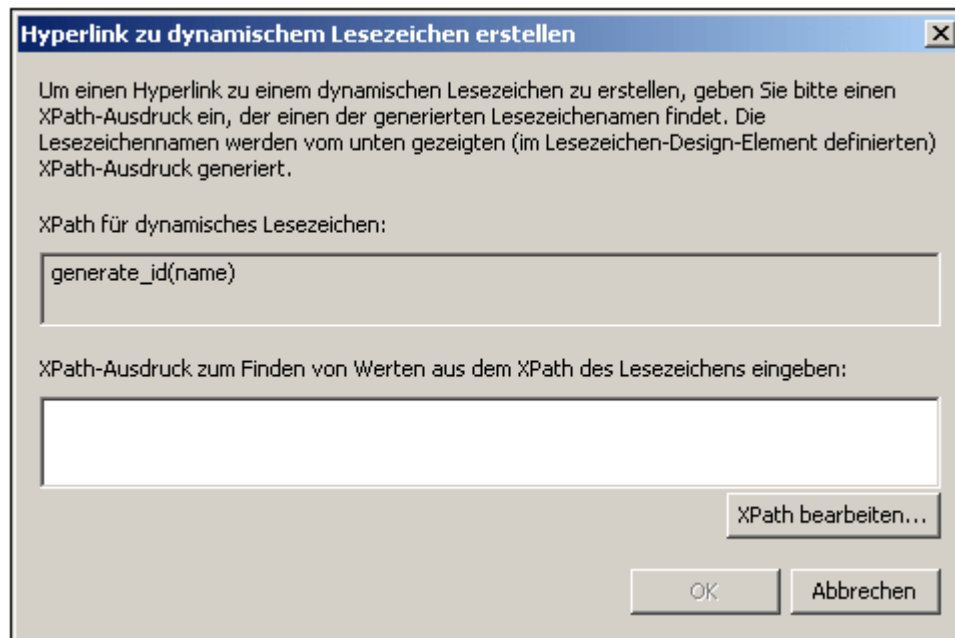
Anlegen eines Link zu einem Lesezeichen

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](#) 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 Lesezeichens angezeigt. Hier können Sie einen XPath-Ausdruck erstellen, der einen Namen generiert, der mit dem 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](#). 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](#) 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](#). 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:

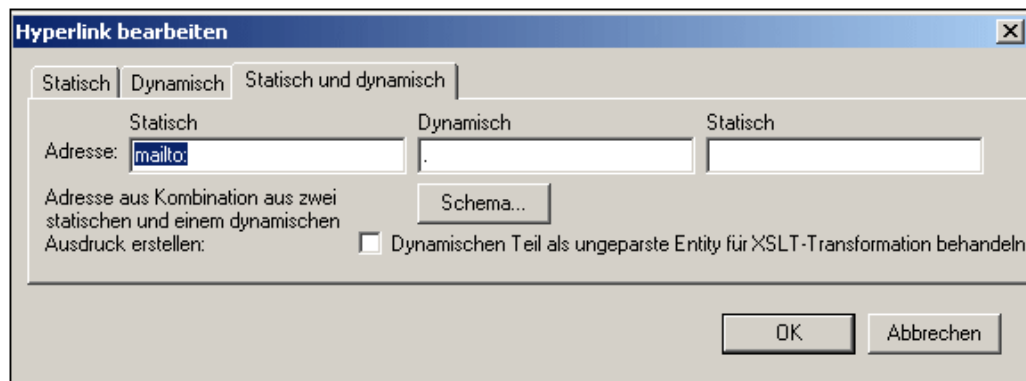
- Da die `generate-id()` Funktion als XPath-Ausdruck ausgewertet werden muss, verwenden Sie zum Definieren des Ziels des Hyperlink das Register "Dynamisch" des Dialogfelds "Hyperlink" (*siehe Abbildung unten*).
- Der ausgewertete Wert des href Attributs muss mit # (dem Fragment Identifiziert) beginnen. 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 dynamischen 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, 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*).



Eine Beschreibung zur Verwendung ungeparster Entities finden Sie im Abschnitt URIs von ungeparsten Entities.

Anmerkung: Die Verknüpfung mit externen Dokumenten wird in der FO-Spezifikation zwar unterstützt, doch kann es aufgrund von FOP-Einschränkungen vorkommen, dass externe Links in mit FOP erstellten PDF-Dokumenten nicht funktionieren.

Änderung des Inhalts eines Parent Node in der Authentic-Ansicht

Der Wert des Parent Node einer Schaltfläche kann allerdings vom Benutzer der Authentic-Ansicht ausgewählt werden und das SPS kann so konzipiert werden, dass die Darstellung sich ändert, je nachdem, welchen Wert der Benutzer der Authentic-Ansicht auswählt.

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 "Hyperlink-Eigenschaften". 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.

Beispiel: Mehrere Sprachen

Oft müssen Dokumente und Authentic-Formulare Inhalt in mehreren Sprachen enthalten oder der Benutzer muss darin die Sprache seiner Wahl auswählen. StyleVision bietet eine Reihe von Funktionen zu diesem Zweck. Unterhalb sehen Sie einige Möglichkeiten, wie dies umgesetzt werden kann. Alle Beispiele finden Sie im Projekt `Examples`, das mit StyleVision mitgeliefert wird. (Das Projekt `Examples` sollte beim ersten Start von StyleVision standardmäßig geladen werden. Sie können es auch laden, indem Sie den Menübefehl **Projekt | Öffnen** wählen und zur Datei `Examples.svp` im Ordner `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples` navigieren).

Verwenden von Variablen und Bedingungen

Die Sprache, die der Benutzer auswählt, wird in eine editierbare Variable eingegeben. Eine [Bedingung](#) mit mehreren Verzweigungen mappt die einzelnen Sprachen auf den richtigen Sprachinhalt. Anhand der vom Benutzer gewählten Sprache wird die entsprechende Bedingungsverzweigung gewählt.

The screenshot shows a form titled "Person" with a language selection section at the top. The language selection section includes a dropdown menu with "English" and "German" options, and a variable "\$Language" that stores the selected language. Below the language selection, there are several data fields for personal information, each with a corresponding variable and a condition. The fields are:

- First**: Variable `First`, Condition `(Inhalt)`
- Last**: Variable `Last`, Condition `(Inhalt)`
- Middle Initial**: Variable `Middle`, Condition `(Inhalt)`
- Maiden Name**: Variable `Maiden`, Condition `(Inhalt)`
- Date of Birth (month/day/year)**: Variable `DateOfBirth`, Condition `(Inhalt)`
- Social Security #**: Variable `SocialSecurity`, Condition `(Inhalt)`

At the bottom of the form, there are buttons for "Person", "I-9", and "\$XML".

Beachten Sie in der Abbildung oben, dass die Auswahl des Benutzers als Wert der editierbaren Variablen eingegeben wird. Die Bedingungen in der Tabelle haben zwei Verzweigungen für die beiden verfügbaren Sprachen und überprüfen den Wert der editierbaren Variablen. Die Ausgabe der Authentic-Ansicht, sieht aus wie in der Abbildung unten.

☐ English ☒ German

Vorname	Niki
Nachname	Devgood
2. Vorname	
Mädchenname	
Geburtstag (Tag.Monat.Jahr)	16.02.1980 1
Sozialversicherungsnummer	555-55-555

Die obige Strategie eignet sich gut für Formulare, in denen der Benutzer die gewünschte Sprache auswählen kann. Einzelheiten dazu finden Sie in der Datei `MultiLangByCondition.sps` im Projekt `Examples`.

Verwendung von Parametern und automatischen Berechnungen

Ein weiteres Szenario wäre jenes, in dem dieselben Daten in verschiedenen Sprachen ausgegeben werden. Eine mögliche Strategie dafür wäre die Verwendung eines [Parameters](#), dessen Wert die Auswahl der Ausgabe in der gewünschten Sprache auslöst. Die jeweilige Sprachausgabe kann z.B. mit Hilfe einer [automatischen Berechnung](#) ermittelt werden. Dabei könnte der jeweilige Inhalt je nach Parameterwert ausgegeben werden.

☒ I-9 ☐ Person

☒ XPath	☐ First (Inhalt) ☐ First
☒ XPath	☐ Last (Inhalt) ☐ Last
☒ XPath	☐ Middle (Inhalt) ☐ Middle
☒ XPath	☐ Maiden (Inhalt) ☐ Maiden
☒ XPath = XPath	☐ DateOfBirth ? 1 2 (Inhalt) ☐ DateOfBirth
☒ XPath	☐ SocialSecurity (Inhalt) ☐ SocialSecurity

☐ Person ☒ I-9 ☒ XML

In der Abbildung oben haben die automatischen Berechnungen XPath-Ausdrücke in der Form:

```
if ( $Language = 'E' ) then 'First' else
if ( $Language = 'G' ) then 'Vorname' else ''
```

Der Wert des globalen Parameters `$Language` kann im SPS-Design geändert werden oder zur Laufzeit über die Befehlszeile bereitgestellt werden. Zur Ausgabe derselben Daten in mehreren Sprachen können mehrere Transformationen durchgeführt werden.

Näheres dazu finden Sie in der Datei `MultiLangByAutoCalcs.sps`, die sich im Projekt `Examples` befindet.

Beispieldateien

Weitere Beispiele finden Sie in der `Examples-Projektdati Examples.svp`, die sich im Ordner `C: /Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples` befindet.

Kapitel 11

SPS-Datei: Darstellung

11 SPS-Datei: Darstellung

Im SPS Design ist für Komponenten eine einzige Gruppe von Stileigenschaften definiert. Diese Stile werden in den entsprechenden Ausgabedokumenten (*Authentic-Ansicht*, *HTML*, *RTF*, *PDF* und *Word 2007+ in der Enterprise Edition*; *Authentic-Ansicht*, *HTML* und *RTF* in der *Professional Edition*; *HTML* in der *Basic Edition*) in die entsprechenden Stil-Markup-Tags konvertiert. Einige Darstellungseffekte wie z.B. Effekte zur interaktiven Web-Darstellung (wie z. B. Auswahllisten und JavaScript Event Handler), sind aufgrund ihrer Natur in paginierten Ausgabedateien (RTF) nicht verfügbar. In diesen Fällen wird in diesen Medien eine passende Darstellung für den Effekt gewählt. StyleVision bietet jedoch für die Druckausgabe wichtige Optionen zur [Definition von Seiten](#). Diese Optionen wie z.B. Seitengröße, Seitenlayout, Kopf- und Fußzeilen werden zusätzlich zu den Stileigenschaften von Komponenten definiert und nur in der RTF-Ausgabe verwendet.

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](#) näher beschrieben.

Die Werte von Stileigenschaften können direkt in die Seitenleiste "Stil" oder die Seitenleiste "Eigenschaften" eingegeben werden oder sie können über [XPath-Ausdrücke](#) 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](#) 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](#). Beachten Sie, dass die in vordefinierten Formaten enthaltenen Stile in entsprechende Markup-Tags für die RTF-Ausgabe konvertiert werden.

Anmerkung: Beachten Sie beim Definieren von CSS-Stilen für eine SPS Komponente, dass einige Stile von ihrer Natur her, nicht auf die paginierte Medien (RTF) angewendet werden können. Auch HTML Selektoren (in externen Stylesheets und globalen Stilregeln) können nicht auf paginierte Medien (RTF) angewendet werden. Bei Verwendung solcher Selektoren wird neben dem Selektor ein Kommentar angezeigt, dass der Stil nicht auf die RTF-Ausgabe angewendet wird.

Design für die Ausgabe in paginierten Medien

Für die Unterstützung paginierter Medien (RTF und XSLT-Stylesheets für-RTF) stehen in StyleVision [Optionen zur Definition von Seiten und Layout](#) zur Verfügung. Diese Optionen stehen zusätzlich zu den Methoden zur Erstellung von Stilen für Komponenten zur Verfügung und werden in diesem Abschnitt im Unterabschnitt [Erstellen der Druckausgabe](#) beschrieben.

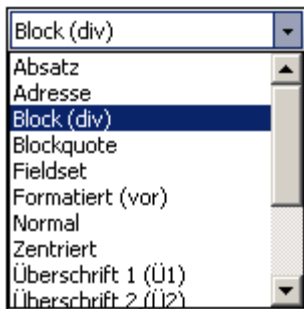
11.1 Vordefinierte Formate

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 werden](#).

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. 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](#).

Die Eingabetaste und vordefinierte Formate

Wenn Sie in der Authentic-Ansicht innerhalb des Inhalts eines Elements mit einem vordefinierten Format die **Eingabetaste** drücken, werden die aktuelle Elementinstanz und ihr Block beendet und es werden an dieser Stelle eine neue Elementinstanz und ein neuer Block eingefügt. Diese Eigenschaft ist z.B. dann nützlich, wenn der Benutzer der Authentic-Ansicht in der Lage sein soll, ein neues Element, z.B. ein Absatztypelement durch Drücken der Eingabetaste zu erstellen.

11.2 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 `disable-output-escaping` 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 `disable-output-escaping` Attribut für die verschiedenen Ausgaben einzeln oder für alle Ausgabe auf `yes`. Die verfügbaren Werte sind:

- Für HTML (um `disable-output-escaping` für die HTML-Ausgabe auf `yes` zu setzen);
- Für Authentic (um `disable-output-escaping` 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 RTF (um `disable-output-escaping` für die RTF-Ausgabe auf `yes` zu setzen).
- Für alle (um `disable-output-escaping` für alle Ausgaben `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.

Statischer Text	disable-output-escaping	Ausgabertext
& amp;	no	& amp;
& amp;	yes	&
&	no	&
&	yes	&
<	no	<
<	yes	<
A	no	A
A	yes	A
& amp; lt;	no	& amp; lt;
& amp; lt;	yes	<
& amp; amp; lt;	yes	<
& amp; <	yes	& <

Anmerkung: Disable-Output-Escaping wird in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

Verwendung von "disable output-escaping" 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. Vergewärtigen Sie sich dies, wenn Sie die Eigenschaft "Disable-Output-Escaping" verwenden.

11.3 Werteformatierung (Formatieren von numerischen Datentypen)

Mit Hilfe der Werteformatierung kann der Inhalt von numerischen Datentyp-Nodes nach dem XML-Schema (*siehe [Liste unten](#)*) 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 Zeitzonekomponente wie z.B. `+02:00`.) Die Werteformatierung wird in der Authentic-Ansicht angezeigt und kann, je nach Formatierungsdefinition auch für die Anzeige in der HTML- und RTF- 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](#)*), für die eine Werteformatierung verfügbar ist.

In den Unterabschnitten dieses Abschnitts wird Folgendes beschrieben:

- wie die [Werteformatierungsmechanismen funktionieren](#), und
- die [Syntax](#) 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 Authentic-Ansicht und optional 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`

Funktionsweise der Werteformatierung

Werteformatierung kann auf folgende Datentypen angewendet werden:

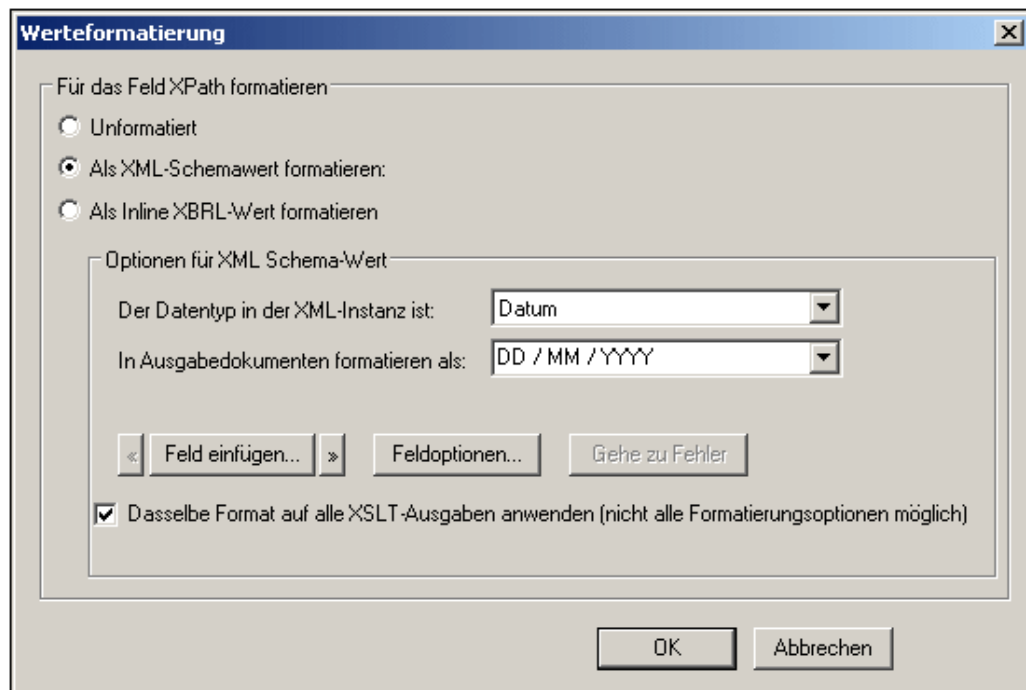
- einen [numerischen Datentyp-Node](#) wie z.B. `xs:decimal` oder `xs:date`, welcher im SPS **als Inhalt oder Eingabefeld vorhanden ist**.
- eine automatische Berechnung, deren Ergebnis ein Wert ist, der das lexikalische Format eines [numerischen Datentyps](#) hat.

Definieren der Werteformatierung

Um die Werteformatierung für einen Node oder eine automatische Berechnung im SPS zu definieren, gehen Sie folgendermaßen vor:

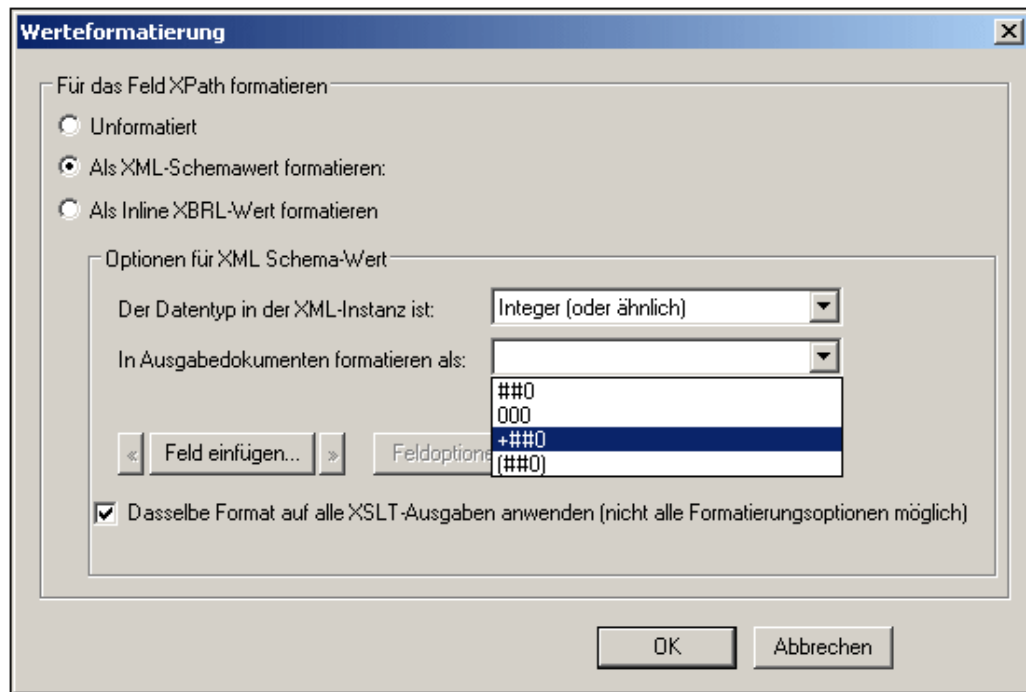
1. Wählen Sie den Platzhalter `Inhalt` oder das Eingabefeld des Node oder die automatische Berechnung aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag `Inhalt` und anschließend die Eigenschaftsgruppe `Inhalt` (oder `XPath`). Klicken Sie anschließend auf die

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.



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](#).

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).

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`.

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
,	Stellentrennzeichen
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...)

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 $\bar{y}$ oder $\bar{Y}$ wird die Länge der Ausgabe durch die Anzahl der $\bar{y}$ oder $\bar{Y}$ Zeichen in der Definition festgelegt. Wenn Sie z.B. $\bar{y}\bar{y}\bar{y}$ angeben, wäre die Ausgabe für einen Wert 2006 006; bei einer Definition $\bar{y}\bar{y}\bar{y}\bar{y}\bar{y}\bar{y}$ 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.

Eingabelänge in der Authentic-Ansicht: Wie der Inhalt eines Node in der Authentic-Ansicht angezeigt wird, hängt von der Werteformatierung für diesen Node ab. Aus diesem Grund ist der Benutzer der Authentic-Ansicht nicht in der Lage, mehr Zeichen als in der Werteformatierungsdefinition erlaubt, einzufügen. Auf diese Art können Sie die Eingabe in der Authentic-Ansicht einschränken. Beachten Sie jedoch, dass, wenn die Länge eines Werts, der bereits **vorher** im XML-Dokument vorkam, größer ist, als in der Formatierungsdefinition angegeben, dennoch der gesamte Wert angezeigt wird.

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 rechte 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
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 (<i>siehe Anmerkung unten</i>)
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.

Anmerkung zum Basisjahr

Bei Verwendung von Kurzformaten (wie `yy` und `YY`) definiert das Basisjahr, dass das Jahrhundert entfernt wird. So könnte z.B. die Feldoption Basisjahr in der Definition `DD-MM-YY{ base=1940}` verwendet werden. Wenn der Benutzer einen Wert eingibt, der gleich oder größer als die letzten beiden Stellen des Basisjahrs ist, welche als zweistellige Ganzzahl betrachtet werden, so ist das Jahrhundert dasselbe wie das des Basisjahrs. Wenn der vom Benutzer eingegebene Wert kleiner ist als der Ganzzahlwert der letzten beiden Stellen, ist das Jahrhundert das Jahrhundert des Basisjahres plus 1. Wenn Sie als Basisjahr z.B. `base=1940` einstellen, so wird bei Eingabe von 50 durch den Benutzer der Authentic-Ansicht, in das XML-Dokument der Wert 1950 eingegeben; wenn der Benutzer 23 eingibt, wird in das XML-Dokument der Wert 2023 eingegeben.

Beachten Sie die folgenden Punkte:

- Obwohl normalerweise als Kurzformat für das Jahr eine 2-stellige Zahl verwendet wird, können bei Angabe eines Basisjahres auch Jahresformate mit einer oder drei Stellen verwendet werden.
- Datentypen, bei denen Kurzformate verwendet werden können, sind: `xs: date`,

`xs:dateTime`, `xs:gYear`, **und** `xs:gYearMonth`.

- Stellen Sie bei Definition des Eingabeformats für eine automatische Berechnung sicher, dass im Dialogfeld "Werteformatierung" der richtige Datentyp ausgewählt ist. (Der ausgewählte Datentyp sollte der des Ergebnisses der automatischen Berechnung sein.)
- Bei Verwendung des Feldtyps `yy` ist das Standardbasisjahr 1930. Dieser Wert wird außer Kraft gesetzt, wenn Sie explizit ein Basisjahr definieren.
- Wenn der Feldtyp `yy` ohne Angabe eines Basisjahres verwendet wird, kann der Benutzer der Authentic-Ansicht nur die letzten beiden Stellen des vierstelligen Jahreswerts ändern; die ersten beiden Stellen bleiben im XML-Dokument unverändert.

11.4 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](#). Externe Stylesheets können über die Seitenleiste [Design-Übersicht](#) und die Seitenleiste [Globale Stile](#) hinzugefügt werden.
- In [globalen Stilen](#) 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 Stile](#) definiert.
- [Lokal](#) 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](#) 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](#) (lokale Stile) und in der Seitenleiste [Globale Stile](#) (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. **Spezifität.** Wenn zwei Regeln für dasselbe Element gelten, hat die Regel mit dem spezifischeren Selektor Vorrang.
3. **Reihenfolge.** Wenn zwei Regeln denselben Ursprung und dieselbe Spezifität 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 den Modulen, die in der Modulliste weiter unten gereiht sind, Vorrang. Nähere Informationen zu modularen SPSs finden Sie im Abschnitt [Modulare SPSs](#).

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](#), ob für das HTML- und das Authentic-Ansichts-Ausgabedokument die Stile entsprechend der [älteren oder neueren Interpretation von Internet Explorer](#) verwendet werden sollen. Definieren Sie anschließend die CSS-Stile entsprechend der gewählten Interpretation. Nähere Informationen finden Sie unter [Eigenschaften: CSS-Unterstützung](#).

Anmerkung: Nähere Informationen zur CSS-Spezifikation finden Sie unter <http://www.w3.org/TR/REC-CSS2/>.

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](#)
- [Anzeigen und Ändern der Struktur der externen CSS Stylesheets](#)
- [Ändern der Vorrangigkeit der externen CSS Stylesheets](#)
- [Wechseln zwischen der dem vollständigen CSS Stylesheet-Satz und einem einzigen CSS Stylesheet](#)

Externe CSS Stylesheets können über zwei Seitenleisten verwaltet werden: die [Seitenleiste "Globale Stile"](#) und die [Seitenleiste "Design-Übersicht"](#). 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 "Globale Stile" den Eintrag "Extern" aus (*Abbildung unten*).



2. Klicken Sie links oben in der Symbolleiste des Fensters Globale 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](#) 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 "Globale 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 "Globale 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 "Globale Stile" auf die Schaltfläche **Zurücksetzen**.

Ändern der Vorrangigkeit der externen CSS Stylesheets

Die im Fenster "Globale 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>
```

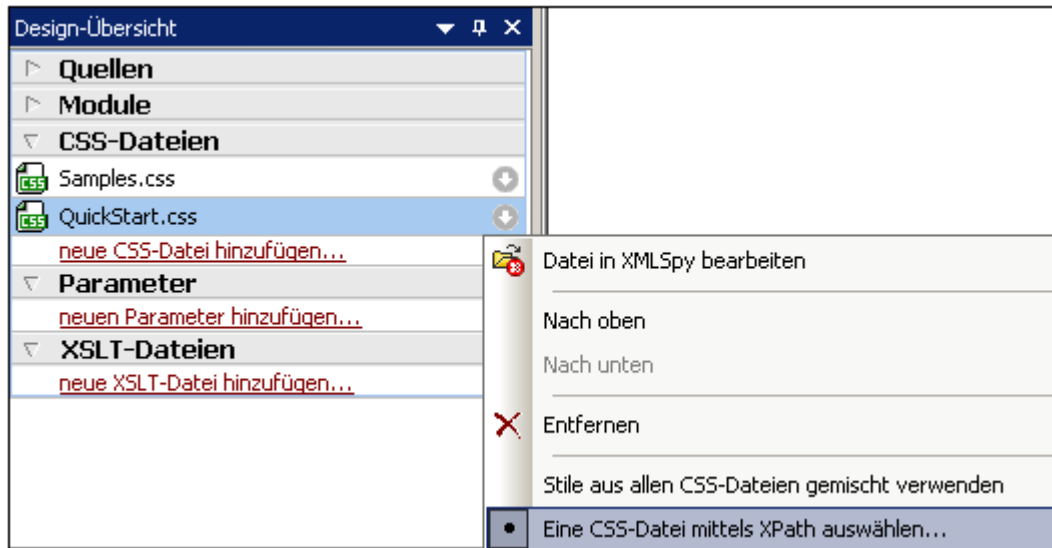
Die Reihenfolge, in der die Dateien in der HTML-Datei aufgelistet sind, entsprechen der Reihenfolge, in der sie im Fenster "Globale Stile" und in der Seitenleiste "Design-Übersicht" in der CSS-Dateistruktur unter "Extern" aufgelistet sind. Um die Reihenfolge der CSS Stylesheets im der Seitenleiste "Globale 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 "Globale 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.

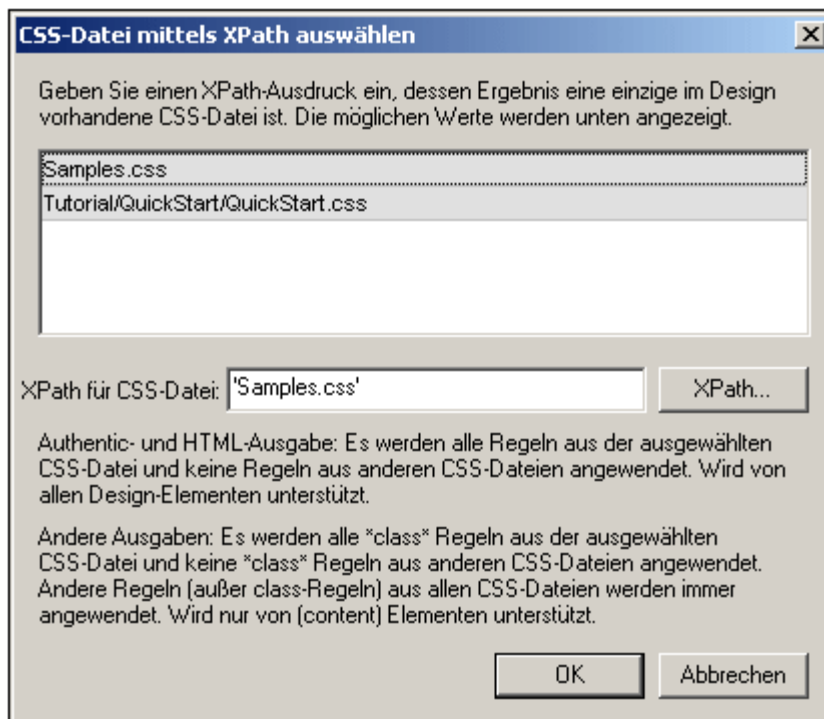
Wechseln zwischen allen CSS-Dateien und einer einzigen CSS-Datei

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.



In der Authentic- und der HTML-Ausgabe werden alle Regeln aus der ausgewählten CSS-Datei angewendet. Diese Regeln werden in allen Design-Komponenten unterstützt. In der RTF-Ausgabe hingegen werden nur die Klassen-Selektor-Regeln aus der ausgewählten CSS-Datei angewendet. Andere Regeln werden aus allen CSS-Dateien angewendet. Diese Regeln können auf die folgenden Design-Komponenten angewendet werden: [automatische Berechnungen](#), [den Platzhalter \(Inhalt\)](#), [Absatz \(Block\)-Komponenten](#) und [Tabellenzellen](#).

Globale Stile

Globale Stile werden im Fenster Globale Stile für das gesamte SPS Design definiert und im Fenster Globale Stile unter dem Eintrag "Global" aufgelistet. Sie werden an die Authentic-Ansicht und das HTML-Ausgabedokument als CSS-Regeln übergeben. Im HTML-Dokument werden diese CSS-Regeln in das Element `/html/head/style` geschrieben.

Im Fenster Globale 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

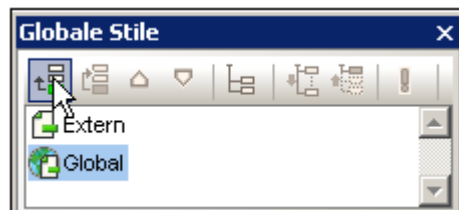
Die folgenden [Selektoren](#) werden unterstützt:

- *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`

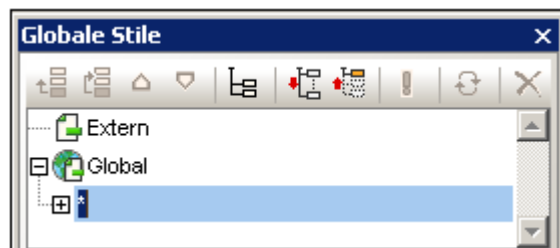
Hinzufügen eines globalen Stils

Um einen globalen Stil zum SPS Design hinzuzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht im Fenster "Globale Stile" den Eintrag "Global" aus (*Abbildung unten*).



2. Klicken Sie auf die Schaltfläche **Hinzufügen** links oben in der Symbolleiste des Fensters Globale 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, klicken Sie (i) entweder auf die rechte Maustaste und wählen Sie im Untermenü "Selektor hinzufügen" eine Option aus, oder (ii) klicken Sie auf den Selektor und bearbeiten Sie ihn.



4. Definieren Sie nun die CSS-Eigenschaftswerte für den Selektor. Eine Anleitung dazu

finden Sie im Abschnitt [Definieren von Stilwerten](#).

- 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 "Globale 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 im Fenster "Globale Stile" auf einen globalen Stil oder den Eintrag "Global" 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 | HTML-ElementName** hinzugefügt werden.

Bearbeiten und Löschen von globalen Stilen

Sowohl der Stilselektor als auch seine Eigenschaften können im Fenster "Globale 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](#). (Die Stileigenschaften können in drei möglichen Ansichten angezeigt werden. Diese Ansichten, bzw. wie man zwischen diesen wechselt, sind im Abschnitt [Ansichten von Eigenschaftsdefinitionen](#) 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](#). (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](#).

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 "Globale 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;
            }
        .main { color: green; }
```

```
-->
</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 Globale 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="main">`, so würden diesem Element zwei globale Stile entsprechen: der Stil mit dem Selektor `h1` und der Stil mit dem Selektor `.main`. Es wird die Farbeigenschaft des `.main` Selektors verwendet, weil dieser in der Stilliste nach dem `h1` Selektor aufgelistet ist. 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.

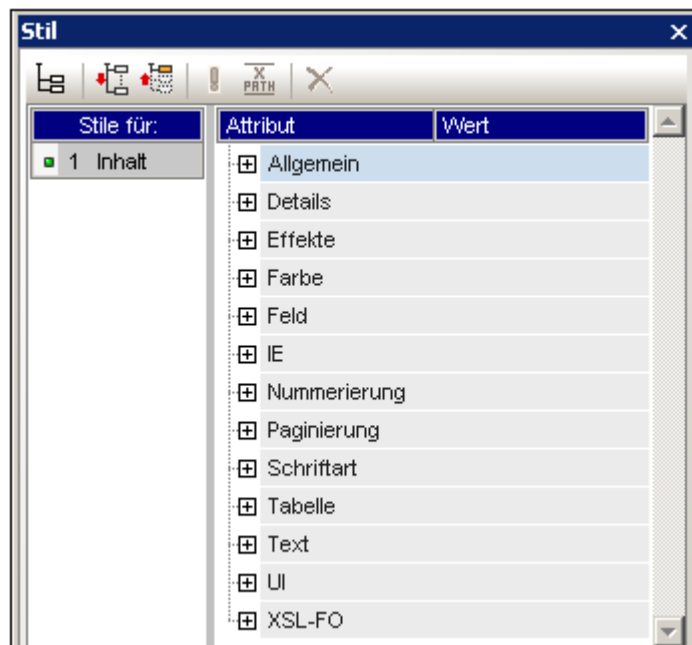
Um die Priorität eines globalen Stils zu ändern, wählen Sie den Stil aus und verschieben Sie den globalen Stil in der Struktur der globalen Stile mit Hilfe der Schaltflächen **Nach oben** und **Nach unten**. Wenn Sie z.B. den globalen Stil `.main` an eine Stelle vor den globalen Stil `h1` verschieben, hätte die Farbeigenschaft des Stils `h1` Vorrang vor der des Stils `.main`.

Lokale Stile

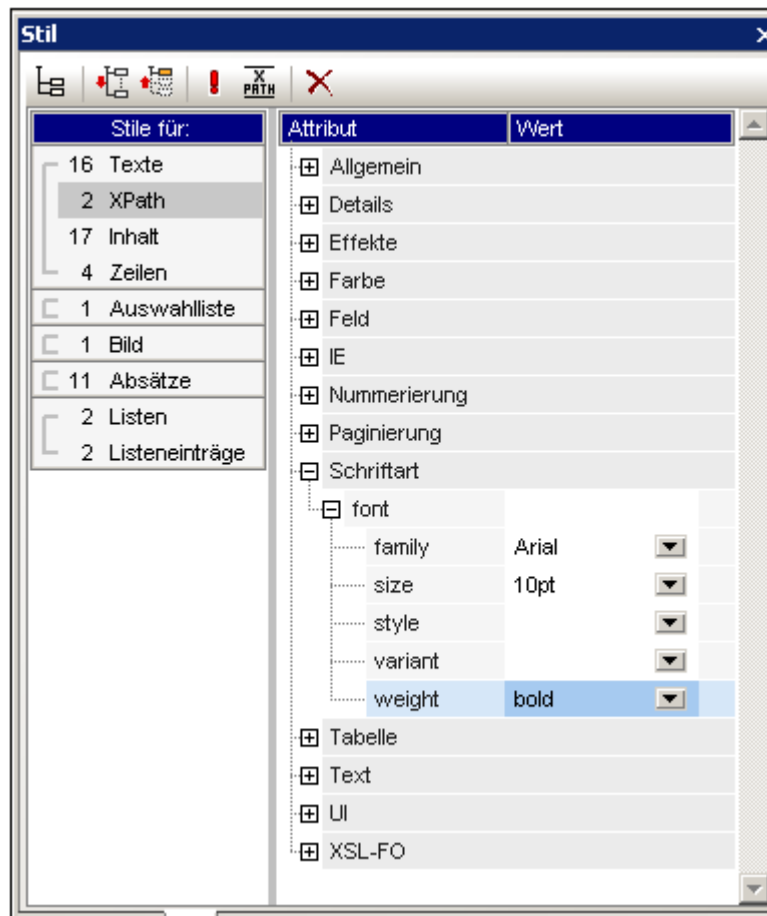
Wenn Stile lokal definiert werden, werden die Stilregeln direkt in der Komponente definiert. Diese lokalen Stilregeln haben Vorrang vor globalen Stilregeln und Stilregeln in externen CSS Stylesheets, die diese Komponente auswählen. Lokal definierte Stile sind CSS-Stile und werden in der Seitenleiste [Stil](#) definiert (im Gegensatz zu globalen Stilen, die in der Seitenleiste [Globale Stile](#) definiert werden).

Die lokale Definition eines Stils erfolgt in zwei Schritten:

1. Die Komponente, für die Stile definiert werden sollen, wurde 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..



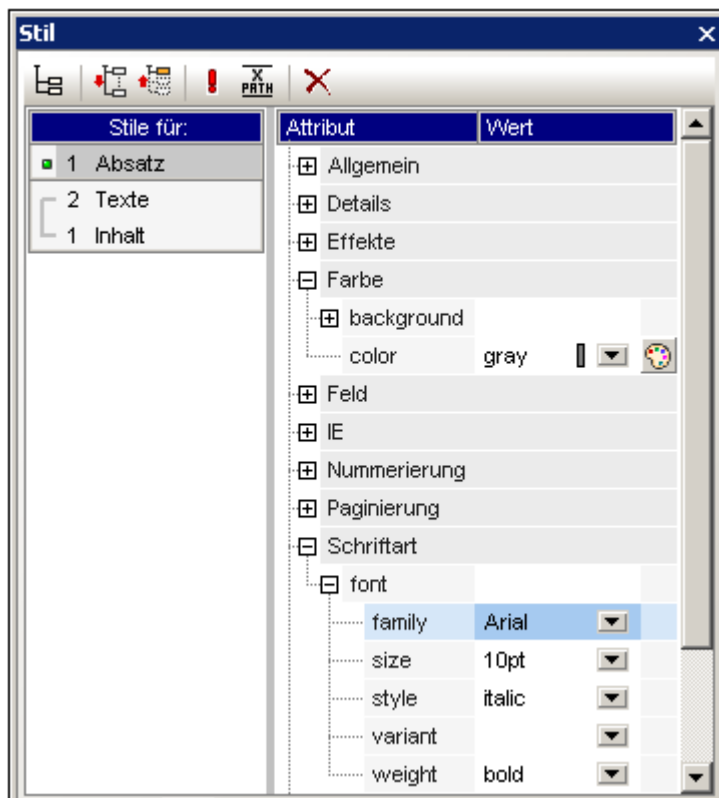
Oft enthält die in der Design-Ansicht ausgewählte Komponente andere Komponenten. In diesem Fall werden alle Komponenten in der Auswahl geordnet nach Komponententyp in der Seitenleiste "Stil" in der Spalte *Stile für* angezeigt. In der Abbildung unten sehen Sie die verschiedenen Komponententypen, die 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 in der Spalte *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, in der Spalte *Stile für* ausgewählt haben, werden die Stile für diese Auswahl in der [Spalte für die Stildefinitionen](#) definiert. Eine Anleitung dazu finden Sie im Abschnitt [Definieren von Stilwerten](#).

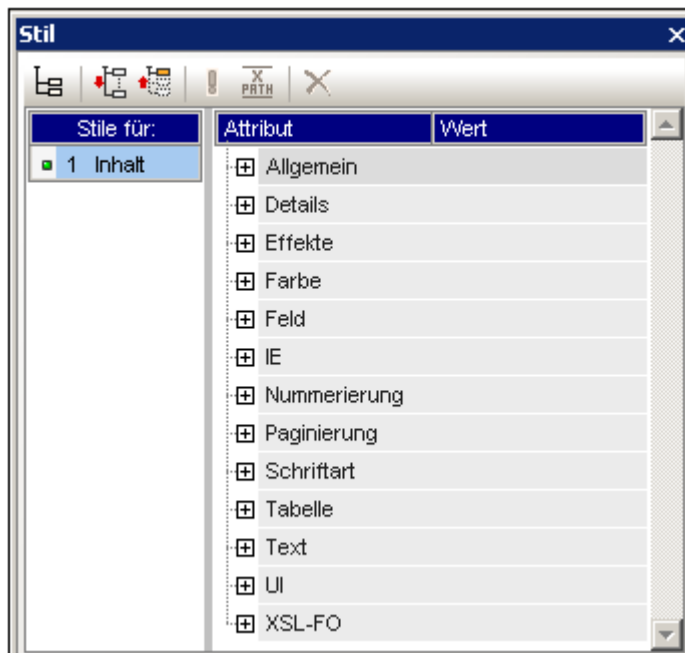
Definieren von Stilwerten

Die Stileigenschaften für den in der Spalte *Stile für* ausgewählten Komponententyp werden in der [Seitenleiste "Stil" in der Spalte für die Stildefinitionen](#) (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 in der Spalte *Stildefinitionen* definierte Stilwert auf alle Instanzen der ausgewählten Komponententypen angewendet.



Stileigenschaftsgruppen

Die in der Spalte *Stildefinitionen* verfügbaren Eigenschaften sind in Gruppen gegliedert (siehe Abbildung unten).

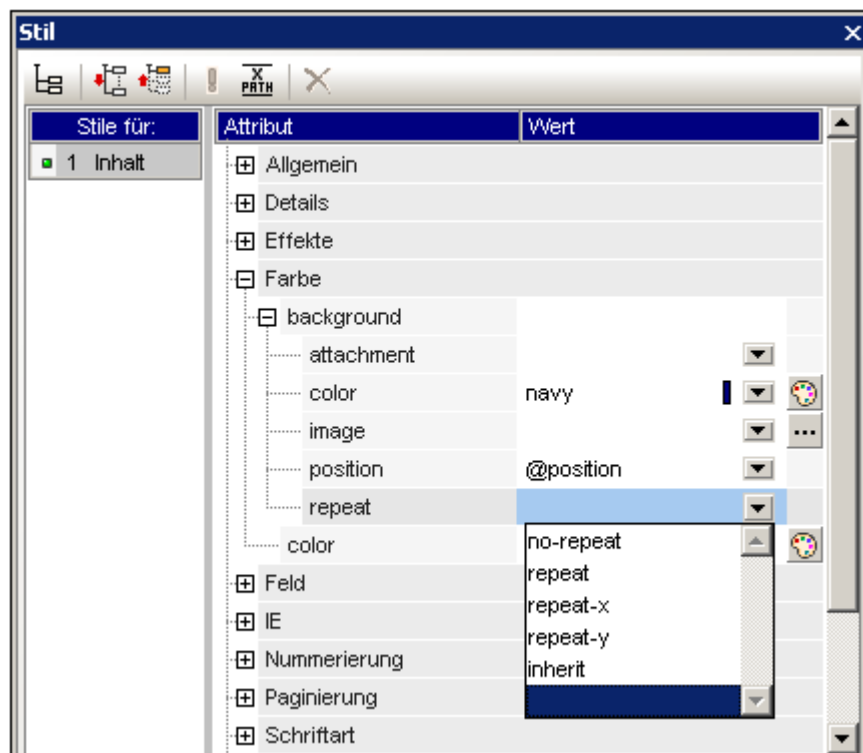


Die Anzeige der Eigenschaften kann mit Hilfe der Symbolleisten-Schaltflächen [Nicht leere auflisten](#), [Alle erweitern](#) und [Alle reduzieren](#) konfiguriert werden. Die einzelnen Eigenschaftsgruppen können erweitert werden, um Zugriff auf die Untergruppen und Stileigenschaften zu erhalten (siehe Abbildung unten).

Eingabe von Eigenschaftswerten

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`).



- Werte für Stile können auch mit Hilfe eines [XPath-Ausdrucks](#) 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](#) 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.

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 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 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"

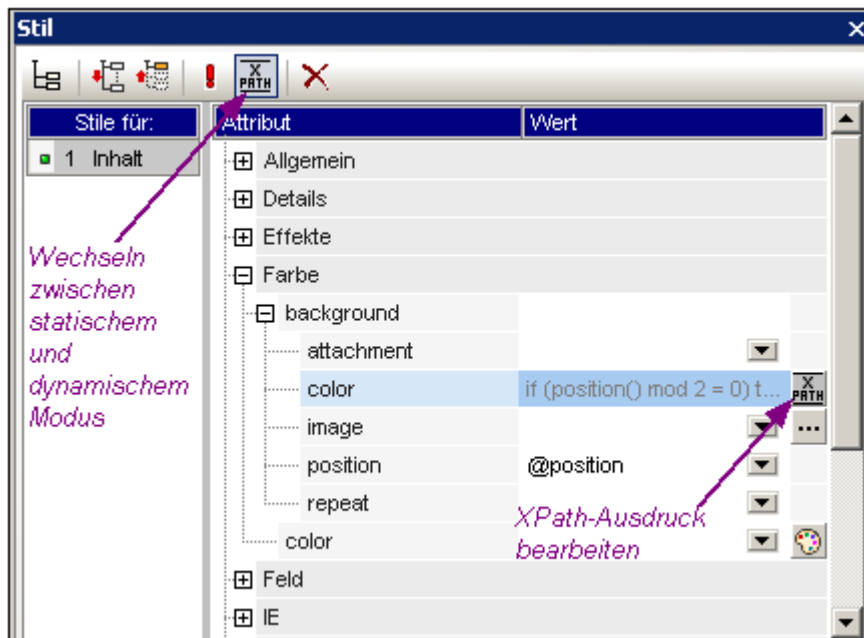
Statischer Modus und dynamischer (XPath)-Modus bei Eigenschaftswerten

Bei den Eigenschaften, bei denen [XPath-Ausdrücke aktiviert sind](#), 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"](#) wird angezeigt. In diesem Dialogfeld können Sie den XPath-Ausdruck für die Eigenschaft eingeben. Klicken Sie anschließend auf **OK**.

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](#).) 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"](#) 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"](#) 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"](#) 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()` Function 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 ausgegeben 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 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.

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](#)
- [den Platzhalter \(Inhalt\)](#)
- [Absatz \(Block\)-Design-Elemente](#)
- [Tabellenzellen](#)

Vorteile von Kombistilen

Kombistile bieten die folgenden Vorteile:

- Die Stileigenschaften befinden sich in den XML-Daten und können daher vom Benutzer geändert werden. Die RichEdit-Funktion in der Authentic-Ansicht ermöglicht die grafische Bearbeitung von Kombistilen mit Hilfe einer Symbolleiste. Nähere Informationen zum Einrichten von RichEdit für Kombistile finden Sie unter [Kombistile in Authentic](#).
- 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">
```

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

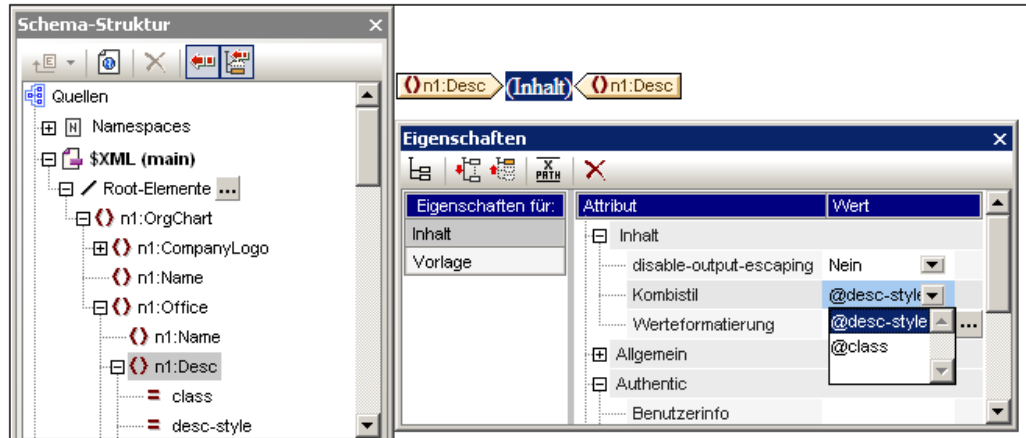
Definieren eines Attributs als Wert für einen Kombistil

Wenn Sie den Kombistil einer Design-Komponente als Attribut definieren, kann der Benutzer der Authentic-Ansicht diesen Kombistil bearbeiten. Der Benutzer der Authentic-Ansicht kann den Cursor an eine beliebige Stelle in der Textausgabe des Design-Dokuments platzieren und den Kombistil dieser Design-Komponente über die RichEdit-Symbolleiste der Authentic-Ansicht

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. (Beachten Sie, dass die Auswahlliste auch einen leeren Eintrag hat, falls Sie keinen Kombistil anwenden möchten. In diesem Fall ist die RichEdit-Funktion der Authentic-Ansicht in der Ausgabe dieses Design-Komponente nicht aktiviert.)

Der Benutzer kann den Kombistil dieser Design-Komponente nun in der Authentic-Ansicht über die RichEdit-Symbolleiste bearbeiten.

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 des Dialogfelds "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-Ausgabe 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 geschweiften Klammern und ermöglichen eine dynamische Zuweisung des Attributwerts.

11.5 Flexibles Zuweisen von Textstilen in Authentic

Ein SPS kann so konfiguriert werden, dass auch der Benutzer der Authentic-Ansicht einen Text mit Stilen versehen kann. Dazu wird ein Textfragment in der Authentic-Ansicht ausgewählt und es werden Stileigenschaften für den ausgewählten Text definiert. Diese Stileigenschaften können von Ihnen (dem Ersteller des SPS) vordefiniert werden oder sie können vom Benutzer der Authentic-Ansicht definiert werden. In beiden Fällen erhält der Benutzer der Authentic-Ansicht die Möglichkeit Text in der Authentic-Ansicht mit Stilen zu versehen.

Die folgenden Optionen zum Zuweisen von Textstilen stehen zur Verfügung:

- **Kombistile:** Im Attribut eines Elements im XML-Quelldokument wird eine Gruppe von CSS-Stileigenschaften (der Kombistil) definiert. Indem man dieses Attribut als Kombistil-Eigenschaft einer Design-Komponente definiert, steht der Kombistil über die [RichEdit](#)-Funktion der Authentic-Ansicht zur Bearbeitung zur Verfügung. Eine Beschreibung dazu finden Sie im Abschnitt [Kombistile](#).
- **RichEdit:** Wenn ein Element im SPS-Design als RichEdit-Komponente erstellt wurde, so kann der Benutzer der Authentic-Ansicht Textfragmente in diesem Element auswählen und diesen über die RichEdit-Stileigenschaften der Authentic-Ansicht Stile zuweisen. Mit Hilfe von RichEdit kann der Benutzer der Authentic-Ansicht Attribute wie die Schriftart (font), die Schriftbreite (font-style), den Schriftschnitt (font-style), die Schriftartverzierung (font-decoration), die Farbe (color), die Hintergrundfarbe und die Textausrichtung definieren. Nähere Informationen dazu finden Sie im Abschnitt [RichEdit](#).
- **Textstatus-Symbole:** Sie können als Autor eines SPS eine Symbolleisten-Schaltfläche für die Authentic-Ansicht erstellen und diese Schaltfläche mit einem Elementnamen verknüpfen. Solche Symbolleisten-Schaltflächen werden als Textstatus-Symbole bezeichnet. Wenn der Benutzer der Authentic-Ansicht ein Textfragment in der Authentic-Ansicht auswählt und auf ein Textstatus-Symbol klickt, wird der markierte Text in das mit dem Textstatus-Symbol verknüpfte Element eingeschlossen (Ein Textstatus-Symbol ist nur dann aktiv, wenn das Element entsprechend dem Schema an dieser Stelle zulässig ist). Jede für dieses Element (in einer globalen Vorlage für dieses Element) definierte Textformatierung wird auf das ausgewählte Textfragment angewendet.

Sie können in Ihrem SPS eine Kombination aus allen diesen drei Textstiloptionen verwenden.

Beachten Sie die folgenden Punkte:

- Bei allen drei Textstiloptionen müssen Stilelemente und/oder -attribute im XML-Dokument vorhanden sein. Folglich müssen die erforderlichen Elemente und/oder Attribute gemäß dem Schema, auf dem das SPS basiert, auf den hierarchischen Ebenen, auf denen diese eingefügt werden sollen, zulässig sein.
- Textstile, die vom Benutzer der Authentic-Ansicht angewendet werden, werden sowohl in der Authentic-Ansicht als auch in den Ausgabeformaten angezeigt.

Anmerkung: Die oben angeführten Stiloptionen stammen aus Stileigenschaften, die in der XML-Datei gespeichert sind, und stehen zusätzlich zu Stileigenschaften, die im SPS zugewiesen werden, zur Verfügung.

Kombistile

Mit der Kombistilfunktion hat der Benutzer der Authentic-Ansicht die Möglichkeit, der gesamten Ausgabe der folgenden Designkomponenten Stile zuzuweisen:

- [automatische Berechnungen](#)
- [Platzhalter \(Inhalt\)](#)
- [Absatz \(Block\)-Design-Elemente](#)
- [Tabellenzellen](#)

Eine Anleitung dazu finden Sie weiter unten.

Anmerkung: Eine weitere allgemeinere Beschreibung zu Kombistilen finden Sie im Abschnitt [Kombistile](#). In diesem Abschnitt wird beschrieben, wie Kombistile so konfiguriert werden können, dass sie vom Benutzer der Authentic-Ansicht bearbeitet werden können.

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">
```

Als Ersteller des SPS erlauben Sie dem Benutzer der Authentic-Ansicht nun, diese Stileigenschaften zu bearbeiten. Dadurch kann der Benutzer der Authentic-Ansicht der Textausgabe der Design-Komponente Stile zuweisen. Sie geben dem Benutzer die Möglichkeit, einen Kombistil zu bearbeiten, indem Sie den Kombistil der Design-Komponente als Attribut definieren, das den Standard-Kombistil enthält (*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:

<code>font-family</code>	<code>font-size</code>	<code>font-weight</code>	<code>font-style</code>
<code>color</code>	<code>background-color</code>	<code>text-align</code>	<code>text-decoration</code>

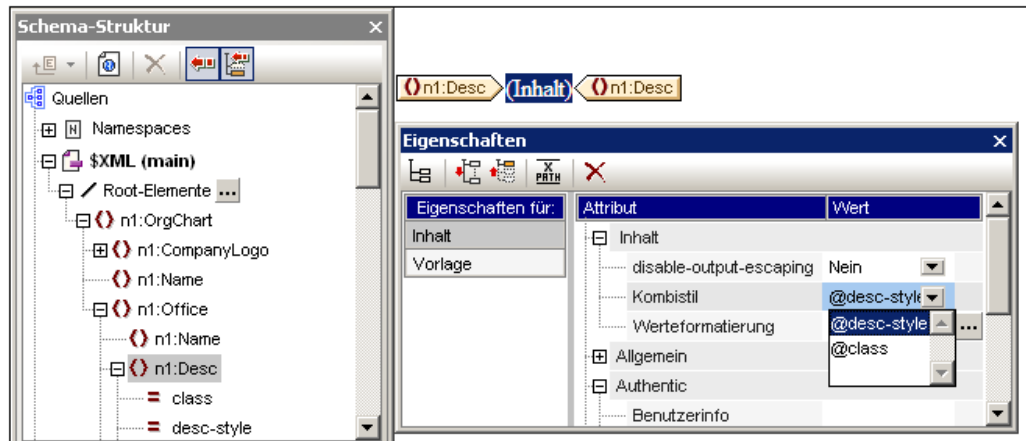
Definieren eines Attributs als Wert für einen Kombistil

Wenn Sie den Kombistil einer Design-Komponente als Attribut definieren, kann der Benutzer der Authentic-Ansicht diesen Kombistil bearbeiten. Der Benutzer der Authentic-Ansicht kann den Cursor an eine beliebige Stelle in der Textausgabe des Design-Dokuments platzieren und den Kombistil dieser Design-Komponente über die RichEdit-Symbolleiste der Authentic-Ansicht 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. (Beachten Sie, dass die Auswahlliste auch einen leeren Eintrag hat, falls Sie keinen Kombistil anwenden möchten. In diesem Fall ist die RichEdit-Funktion der Authentic-Ansicht in der Ausgabe dieses Design-Komponente nicht aktiviert.)

Der Benutzer kann den Kombistil dieser Design-Komponente nun in der Authentic-Ansicht über die RichEdit-Symbolleiste bearbeiten.

RichEdit

Wenn ein Element im SPS-Design als RichEdit-Komponente erstellt wurde, kann der Benutzer der Authentic-Ansicht Textfragmente innerhalb dieses Elements markieren und ihnen über die RichEdit-Stileigenschaften der Authentic-Ansicht Stile zuweisen. Mit Hilfe von RichEdit kann der Benutzer der Authentic-Ansicht die Schriftart (font), die Schriftbreite (font-weight), den Schriftstil (font-style), die Schriftartverzierung (font-decoration), die Farbe (color), die Hintergrundfarbe und die Textausrichtung definieren.

Diese Beschreibung der RichEdit-Funktion ist folgendermaßen gegliedert:

- [RichEdit Funktionsweise](#)
- [Erstellen eines Elements im SPS-Design als RichEdit-Komponente](#)
- [Verwenden von RichEdit in der Authentic-Ansicht](#)

RichEdit Funktionsweise

Wenn ein Benutzer der Authentic-Ansicht ein Textfragment in der Authentic-Ansicht auswählt und RichEdit-Stile darauf anwendet, wird das entsprechende Textfragment in das RichEdit-Stilelement und Attribut gesetzt, das die Stilinformationen enthält. Die vom Benutzer der Authentic-Ansicht ausgewählten RichEdit-Stileigenschaften werden als Wert dieses Stilattributs eingefügt.

Wenn z.B. bei folgendem Textfragment im XML-Quelldokument

```
<p> ... Altova StyleVision 2012 features a unique graphical
design interface ... </p>
```

einem Teil dieses Textfragments eine RichEdit-Stileigenschaft "bold" zugewiesen wird, so wird das Textfragment im XML-Quelldokument folgendermaßen dargestellt:

```
<p> ... <span style="font-weight: bold">Altova StyleVision 2012</
span> features a unique graphical design interface ... </p>
```

Das RichEdit Stilelement im Beispiel oben ist `span` und sein Attribut, das die RichEdit-Stileigenschaften enthalten soll, ist `style`. Das Element und sein Attribut könnten jeden beliebigen Namen haben. Anstatt das Stilelement `span` zu nennen, könnten Sie es auch `Style` nennen und anstatt das Stilattribut `style` zu nennen, könnten Sie ihm den Namen `css` geben. In diesem Fall hätte das Textfragment den folgenden Markup-Code:

```
<p> ... <Style css="font-weight: bold">Altova StyleVision 2012</
Style> features a unique graphical design interface ... </p>
```

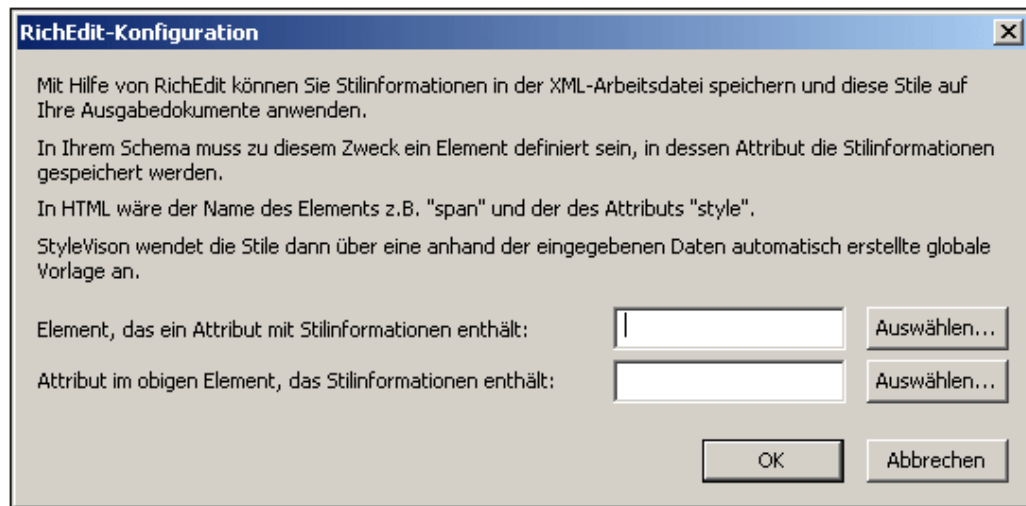
Wichtig ist nur, dass - egal welchen Namen Sie für das Stilelement und sein Attribut wählen - **dieses Stilelement und sein Attribut im Schema definiert** und in dem Element, das den zu formatierenden Text enthält, zulässig sein müssen.

Bei Verarbeitung des Texts mit einem XSLT Stylesheet werden die Stileigenschaften als Markup-Code für das jeweilige Ausgabeformat an die Ausgabe übergeben.

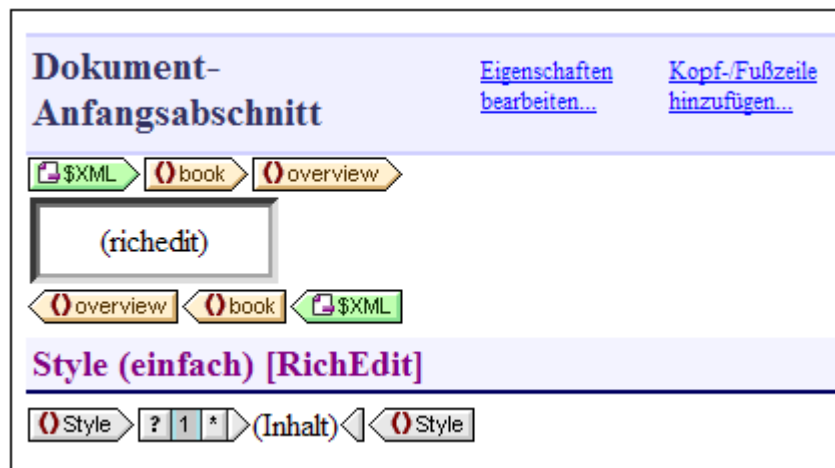
Erstellen eines Elements als RichEdit-Komponente

Um ein Element als RichEdit-Komponente zu erstellen, gehen Sie folgendermaßen vor:

1. Ziehen Sie das Element aus der Schema-Struktur an die gewünschte Stelle im Design.
2. Wählen Sie im Menü, das daraufhin angezeigt wird, den Befehl **RichEdit erstellen**. Daraufhin wird das Dialogfeld "RichEdit-Konfiguration" (*Abbildung unten*) angezeigt.



3. Geben Sie im Dialogfeld "RichEdit-Konfiguration" den Namen des Stilelements und seines Attributs, das die RichEdit-Stileigenschaften enthalten soll, ein. Sie können das gewünschte Element und Attribut auch aus der Schema-Struktur auswählen. Klicken Sie auf die entsprechenden **Auswählen**-Schaltflächen, um die Schema-Struktur zu öffnen.
4. Klicken Sie, wenn Sie fertig sind, auf **OK**. Das Element wird als RichEdit-Komponente (siehe Abbildung unten) erstellt und im Design wird eine **nicht bearbeitbare** globale RichEdit-Vorlage mit den Namen des Stilelements (*in der Abbildung unten* `Style`) erstellt.



Beachten Sie Folgendes:

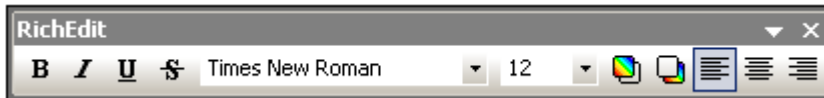
- Die nicht bearbeitbare globale RichEdit-Vorlage wird erzeugt, wenn die erste RichEdit-Komponente im SPS erstellt wird. In der Folge werden keine weiteren zusätzlichen globalen RichEdit-Vorlagen angelegt.
- Wenn später zusätzlich zur ersten RichEdit-Komponente weitere RichEdit-Komponenten erstellt werden, wird das Dialogfeld "RichEdit-Konfiguration" nicht mehr angezeigt. Die RichEdit-Komponente wird automatisch so konfiguriert, dass sie mit der bereits erstellten globalen RichEdit-Vorlage für das Stilelement übereinstimmt.
- Die globale RichEdit-Vorlage kann insofern neu konfiguriert werden, als das Stilelement und -attribut geändert werden können. Klicken Sie dazu mit der rechten Maustaste auf eine RichEdit-Komponente, wählen Sie den Befehl **RichEdit-Element/Attribut konfigurieren** und geben Sie den gewünschten Element- und Attributnamen ein. Der

neue Elementname wird in der Titelleiste der globalen RichEdit-Vorlage angezeigt.

Allen Schema-Elemente, die als RichEdit-Komponenten erstellt wurden, können nun in der Authentic-Ansicht RichEdit-Eigenschaften zugewiesen werden.

Verwenden von RichEdit in der Authentic-Ansicht

Wenn der Cursor in der Authentic-Ansicht in ein Element gesetzt wird, das als RichEdit-Komponente erstellt wurde, werden die Schaltflächen in der RichEdit-Symbolleiste (*Abbildung unten*) aktiv. Andernfalls sind sie ausgegraut.



Wählen Sie den gewünschten Text aus und definieren Sie den gewünschten Stil über die Schaltflächen der RichEdit-Symbolleiste. Der formatierte Text wird in die Tags des Stilelements eingeschlossen

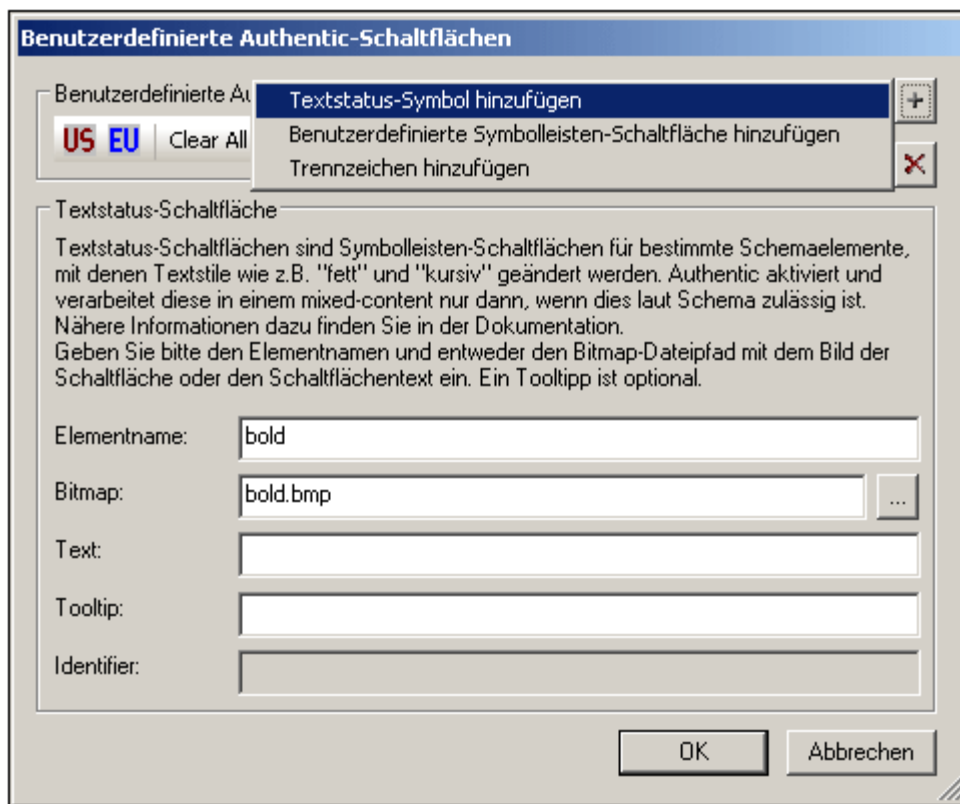
Textstatus-Symbole

Ein Textstatus-Symbol ist eine Authentic-Ansicht Symbolleiste-Schaltfläche, die mit einem Elementnamen verknüpft ist. Wenn Sie in der Authentic-Ansicht auf ein Textstatus-Symbol klicken, wird das ausgewählte Textfragment in das mit dem Textstatus-Symbol verknüpfte Element eingeschlossen. Jede beliebige für dieses Element (in einer globalen Vorlage für dieses Element) definierte Textformatierung wird daher auf das ausgewählte Textfragment angewendet. Auf diese Art kann der Benutzer der Authentic-Ansicht mittels Textstatus-Symbolen Textstile anwenden.

Als Ersteller des SPS werden Sie Textstatus-Symbole erstellen und die Stile der mit den Textstatus-Symbolen verknüpften Elemente definieren.

Erstellen eines Textstatus-Symbols

Um ein Textstatus-Symbol für die Authentic-Symbolleiste zu erstellen, wählen Sie zuerst den Befehl **Authentic | Benutzerdefinierte Symbolleisten-Schaltflächen**, klicken Sie dann im rechten oberen Bereich des Dialogfelds "Authentic Symbolleisten-Schaltflächen" (siehe *Abbildung unten*) auf die Schaltfläche **Hinzufügen** und wählen Sie den Eintrag **Textstatus-Symbol hinzufügen**.



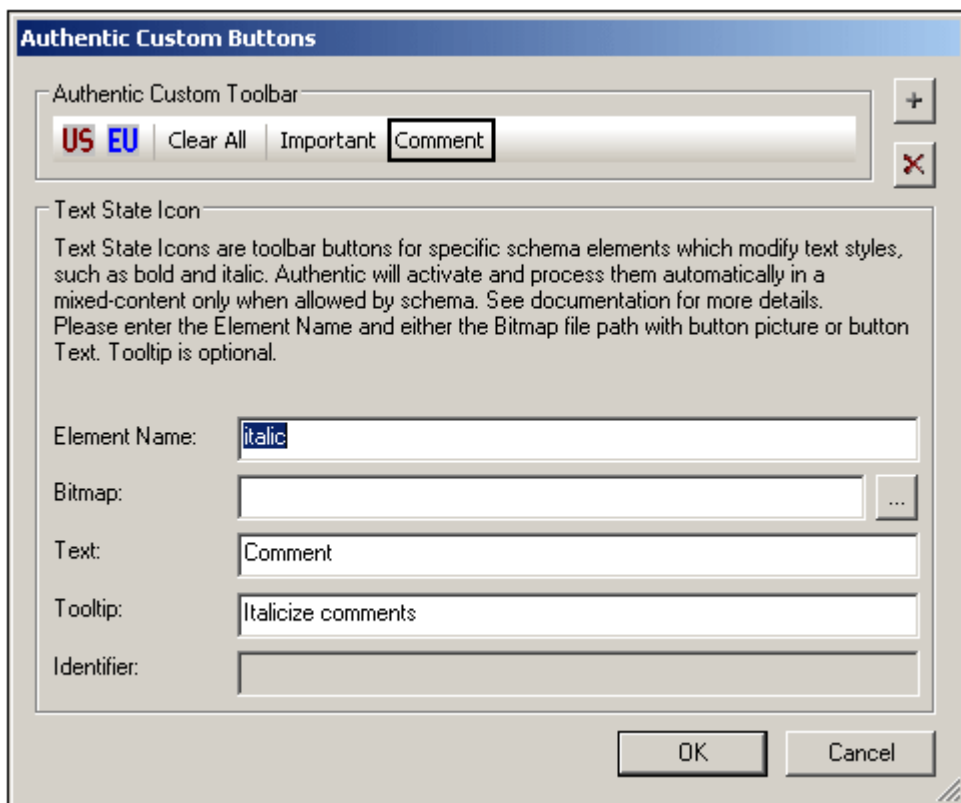
Textstatus-Symbole haben die folgenden Parameter:

- **Elementname:** Dies ist das Element, in das das ausgewählte Textfragment eingeschlossen wird, wenn Sie in der Authentic-Ansicht auf das Textstatus-Symbol klicken.
- **Bitmap:** Der Pfad zu einem Bild für das Textstatus-Symbol. Der Pfad wird relativ zum SPS angegeben.
- **Text:** Wenn kein Bitmap verfügbar ist, wird der in dieses Feld eingegebene Text als

Beschriftung des Textstatus-Symbols angezeigt.

- *Tooltip*: Dies ist ein optionaler Hilfstext, der dem Benutzer der Authentic-Ansicht angezeigt wird, wenn er den Mauszeiger über das Symbol platziert.

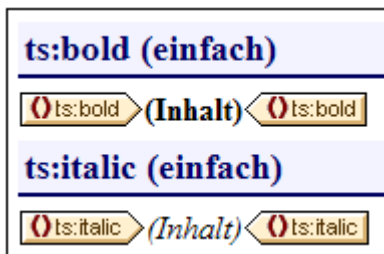
Wir haben im Beispiel oben bekannte Bitmaps als Textstatus-Symbole für die Formatierung von Text mit den Elementen `bold` und `italic` verwendet. Je nach Wunsch können Sie jede beliebige Bitmap-Grafik oder jeden Schaltflächentext verwenden, der Ihrer Meinung nach für den Benutzer der Authentic-Ansicht besser verständlich ist. So können z.B. wichtige Textfragmente mit dem Element `bold` verknüpft werden, während Kommentare mit dem Element `italic` verknüpft werden können. Die Textstatus-Symbole könnten dann, wie in der Abbildung unten gezeigt, erstellt werden.



Beachten Sie in der Abbildung oben, dass das Textstatus-Symbol **Comment** mit dem Element `italic` verknüpft wurde. Wenn daher Text in der Authentic-Ansicht ausgewählt wird und dann auf das Textstatus-Symbol **Comment** geklickt wird, wird dieses Textfragment in das Element `italic` eingeschlossen und auf die für das Element `italic` definierte Art verarbeitet (Beschreibung siehe nächster Abschnitt).

Definieren einer Stilregel für das mit einem Textstatus-Symbol verknüpfte Element

Die Stileigenschaften eines mit einem Textstatus-Symbol verknüpften Elements können in einer globalen Vorlage (siehe Abbildung unten) definiert werden.



In dieser Abbildung sehen Sie die globalen Vorlagen zweier Elemente (*bold* und *italic*), von denen jedes mit einem Textstatus-Symbol verknüpft wurde. Durch Auswahl der Komponente (*Inhalt*) und Definieren des gewünschten Schriftstils in der Seitenleiste "Stil" wurde die Stilzuweisung vorgenommen.

Immer, wenn der Benutzer der Authentic-Ansicht nun diese Elemente rund um Textfragmente erstellt, wird die entsprechende globale Vorlage (mit ihren Textstilen) auf dieses Textfragment angewendet.

11.6 Erstellen der Druckausgabe

Eigenschaften für die Ausgabe auf paginierten Medien (PDF, RTF und Word 2007+ in der *Enterprise Edition*, RTF in der *Professional Edition*) können in der Eigenschaftsgruppe *Seitenlayout* der [Seitenleiste "Eigenschaften"](#) definiert werden. Sie haben beim Design für die Druckausgabe die folgenden Möglichkeiten:

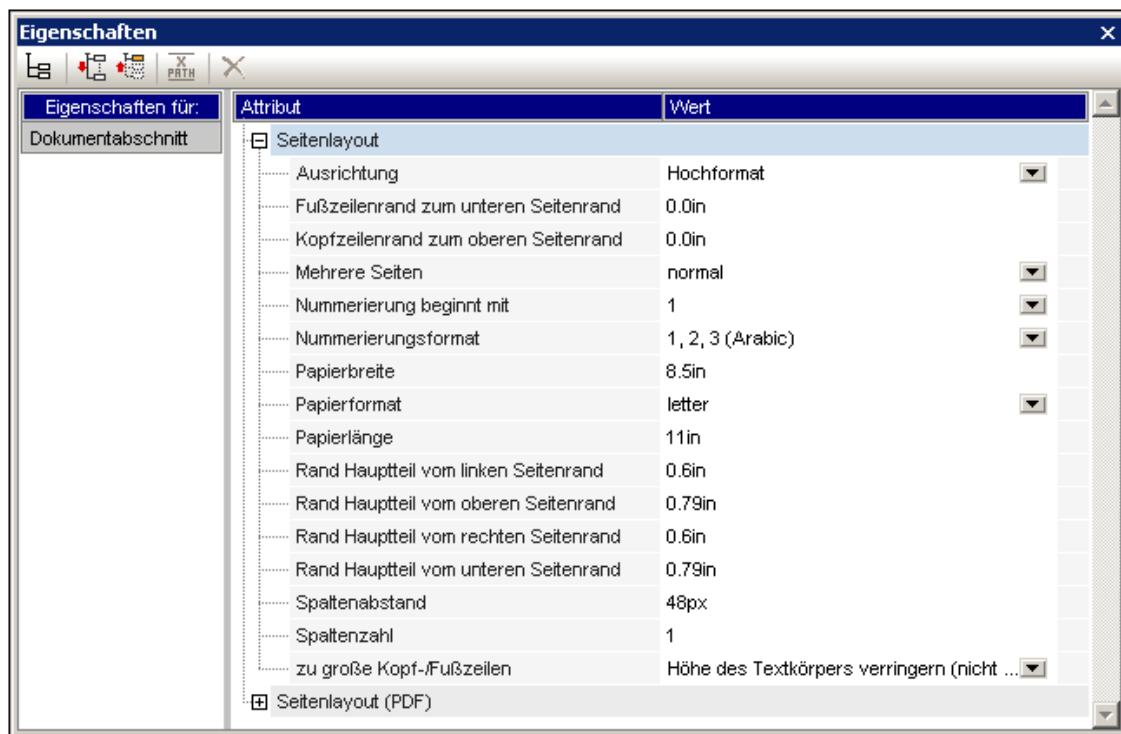
- Das Dokument kann in Abschnitte unterteilt werden, wobei jeder Abschnitt eigene Seitendefinitionen haben kann. Die Eigenschaften, die definiert werden können, sind unten aufgelistet.
- Sie können die Seitenmaße (Höhe und Breite) sowie die Seitenausrichtung (Hoch- oder Querformat) definieren.
- Sie können die Ränder für den Hauptteil der Seite und den Abstand zum oberen und unteren Seitenrand für Kopf- und Fußzeilen definieren. Außerdem können Sie für mehrere Seiten definieren, ob sie als gegenüberliegende Seiten (mit gespiegelten Rändern) dargestellt werden sollen oder ob der rechte und linke Seitenrand bei jeder Seite der gleiche sein soll.
- Sie können für jeden Abschnitt Kopf- und Fußzeilen definieren.
- Sie können festlegen, dass jeder Abschnitt einen eigenen Nummerierungsstil hat und neu durchnummeriert wird oder dass die Seitennummerierung im nächsten Abschnitt fortgesetzt werden soll.
- Sie können für jeden Abschnitt die Anzahl und Breiten der Spalten auf einer Seite definieren.

Seitenleiste "Eigenschaften"

Die Seiteneigenschaften können in der Seitenleiste "Eigenschaften" in der Eigenschaftsgruppe "Seitenlayout" (*Abbildung unten*) für jeden Dokumentabschnitt einzeln vorgenommen werden. Um diese Eigenschaften für einen bestimmten Abschnitt aufzurufen, klicken Sie im Anfangsabschnitt oder in den Abschnittswchselelementen auf den Link "Abschnittseigenschaften bearbeiten" (*Abbildung unten*).

Dokumentabschnitt	Eigenschaften bearbeiten...	Kopf-/Fußzeile hinzufügen...	Kopf-/Fußzeilen ausblenden

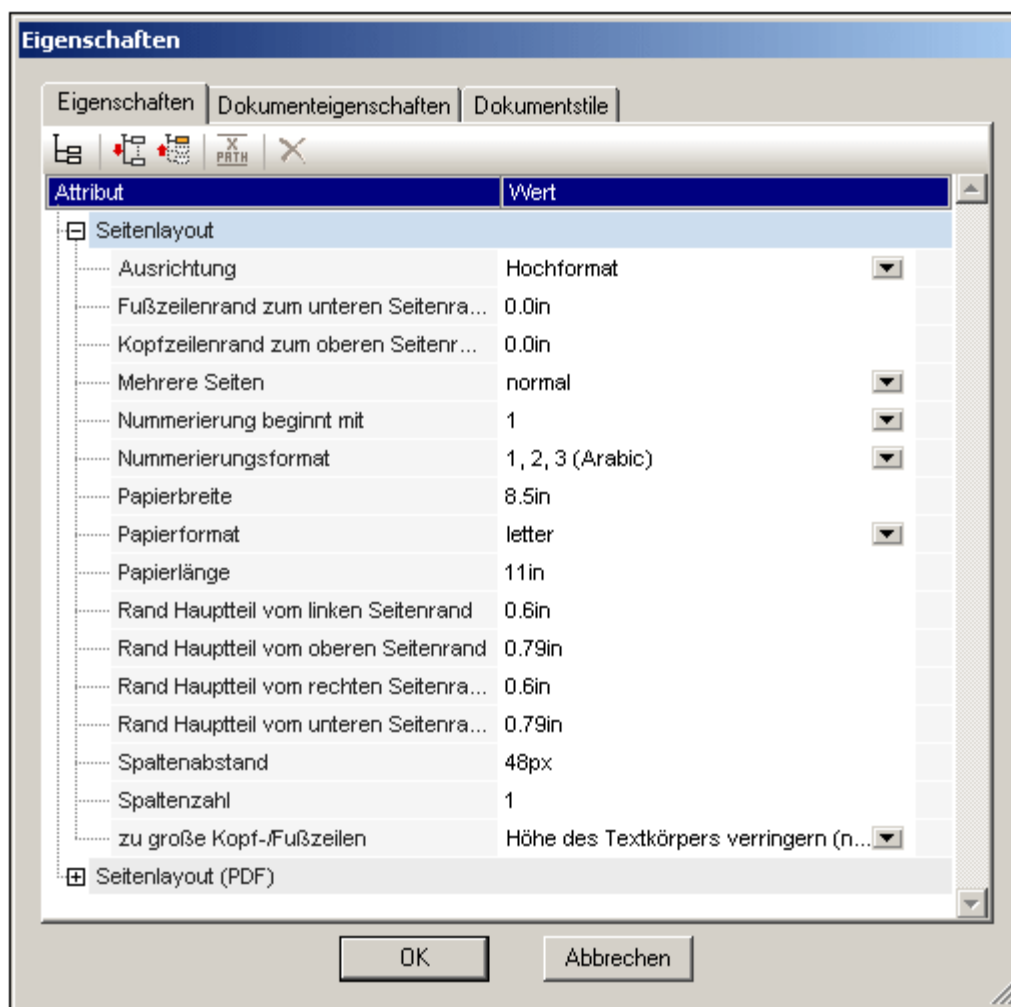
Wenn Sie im Dokumentabschnitt auf den Link "Eigenschaften bearbeiten" klicken, erscheint das Fenster "Eigenschaften", wobei die Seitenlayout-Eigenschaften darin aktiv sind (*Abbildung unten*).



Eine Anleitung zum Definieren der Werte für diese Eigenschaften finden Sie im Abschnitt Seiteneigenschaften.

Dokumentabschnitte

Ein SPS kann in mehreren Dokumentabschnitten angelegt werden, wobei jeder Dokumentabschnitt seine eigenen Seitendefinitionseinstellungen hat. Nehmen wir als Beispiel einen Bericht, der Tabellen mit Daten und beschreibenden Text zu den Daten enthält. Dieser Bericht kann in zwei Dokumentabschnitte unterteilt werden: einen Dokumentabschnitt, der den erklärenden Text enthält und im Hochformat ausgerichtet ist, und einen Dokumentabschnitt, der die Datentabellen enthält, der im Querformat ausgerichtet werden kann. Zu jedem Dokumentabschnitt kann die gesamte Bandbreite der Seiteneigenschaften definiert werden (siehe *Abbildung unten*). Außerdem kann jeder Dokumentabschnitt andere [Kopf- und Fußzeilen](#) haben.



Wenn ein SPS erstellt wird, wird es mit einem Dokumentabschnitt, dem Dokumentanfangsabschnitt angelegt. Dieser Dokumentabschnitt ist der erste Dokumentabschnitt eines Dokuments (mit einem oder mehreren Dokumentabschnitten) und kann nicht gelöscht werden. Die Eigenschaften des Dokumentanfangsabschnitts enthalten Eigenschaften und Stile für das gesamte Dokument, die im Unterabschnitt [Dokumentanfangsabschnitt](#) näher beschrieben sind.

Einfügen von Dokumentabschnitten

Um einen neuen Dokumentabschnitt hinzuzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle im Dokument, an der der neue Dokumentabschnitt beginnen soll.
2. Wählen Sie im Kontextmenü (Aufruf durch Rechtsklick) den Befehl **Seite/Spalte/ Dokumentabschnitt einfügen | Neuer Dokumentabschnitt**. Alternativ dazu können Sie diesen Befehl auch im Menü **Einfügen** auswählen. Daraufhin wird ein neuer Dokumentabschnitt in das Design eingefügt und durch eine Dokumentabschnitts-Tittleiste gekennzeichnet (*siehe Abbildung unten; Der in der Abbildung unten gezeigte Hyperlink Kopf-/Fußzeile ein-/ausblenden erscheint, nachdem eine Kopf- oder Fußzeile zum Dokument hinzugefügt wurde*). In der Ausgabe beginnt ein Dokumentabschnitt auf einer neuen Seite.

Dokumentabschnitt	Eigenschaften bearbeiten...	Kopf-/Fußzeile hinzufügen...	Kopf-/Fußzeilen ausblenden

3. Der neue Dokumentabschnitt erhält die Seitenlayouteigenschaften, die dem Dokumentanfangsabschnitt zum Zeitpunkt der Erstellung des neuen Dokumentabschnitts zugewiesen waren. Diese Seitenlayouteigenschaften für diesen Dokumentabschnitt können über den Hyperlink *Eigenschaften bearbeiten* dieses Dokumentabschnitts bearbeitet werden. Gegebenenfalls können Sie zum Dokumentabschnitt (über den Hyperlink *Kopf-/Fußzeile hinzufügen* separate [Kopf- und Fußzeilen](#) hinzufügen). Eine Anleitung zum Definieren von Seiteneigenschaften und [Kopf- und Fußzeilen](#) finden Sie in den jeweiligen Unterabschnitten dieses Dokumentabschnitts.) Wenn eine Kopf- oder Fußzeile hinzugefügt wurde, wird diese im Design in diesem Dokumentabschnitt angezeigt. Über den Hyperlink *Kopf- und Fußzeilen ein-/ausblenden* können Sie Kopf- und Fußzeilen ein- bzw. ausblenden; dieser Hyperlink erscheint, nachdem eine Kopf- oder Fußzeile in einem Dokumentabschnitt hinzugefügt wurde.

Anmerkungen

Beachten Sie bitte die folgenden Punkte:

- In RTF-Ausgabedokumenten, die anhand von XSLT 1.0 SPS generiert werden, werden nur Dokumentabschnitte, die unmittelbare Unterabschnitte der Hauptvorlage sind, als Dokumentabschnitte erstellt. Diese Einschränkung gilt nicht für RTF-Ausgabedokumente, die anhand von XSLT 2.0 SPS erstellt generiert wurden.
- Im Ausgabedokument beginnt jeder Dokumentabschnitt auf einer neuen Seite.
- Seitenrandeigenschaften werden auch auf die HTML-Seite angewendet.
- Wenn in einem Design mehrere Dokumentabschnitte vorhanden sind, werden die Werte für gespiegelte Seitenränder und die damit verknüpften Eigenschaften margin-left und margin-right aus dem Anfangsabschnitt übernommen. Die Werte dieser Eigenschaften in nachfolgenden Dokumentabschnitten werden ignoriert.

Löschen eines Dokumentabschnitts

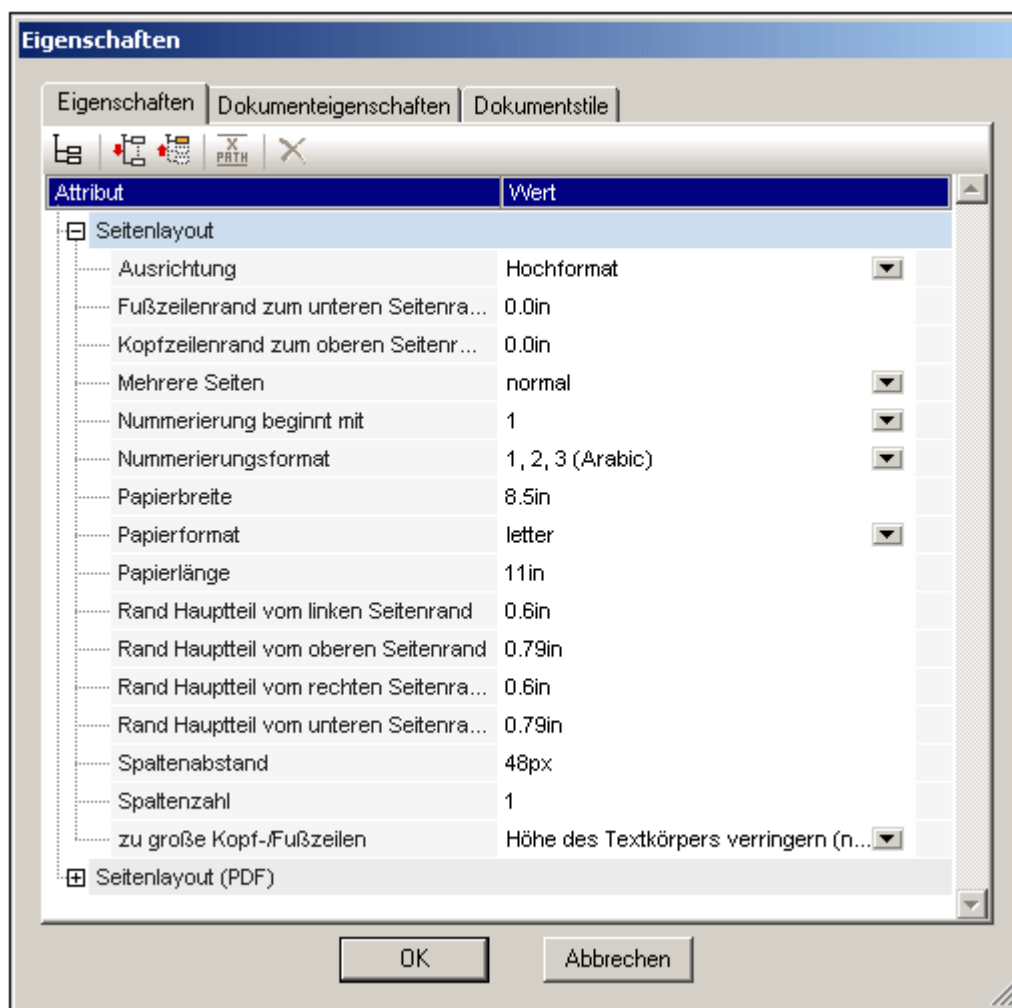
Um einen Dokumentabschnitt zu löschen, klicken Sie in der Tittleiste des Dokumentabschnitts mit der rechten Maustaste auf das Wort *Dokumentabschnitt* und wählen Sie im angezeigten Kontextmenü den Befehl **Bearbeiten | Löschen**. Daraufhin wird der Dokumentabschnitt gelöscht, was durch die Löschung der Tittleiste angezeigt wird. Durch das Löschen des Dokumentabschnitts löschen Sie die Seitenlayouteigenschaften und die Kopf- und Fußzeilen für diesen Dokumentabschnitt. Der Inhalt des Dokumentabschnitts wird jedoch nicht gelöscht.

Dokumentanfangsabschnitt

Ob das Dokument nun einen oder mehrere Dokumentabschnitte hat, die Eigenschaften für das Dokument als Ganzes sind in denen des Dokumentanfangsabschnitts (dem ersten Dokumentabschnitt des Dokuments, *Abbildung der Titelleiste unten*) definiert. Auch [Umschlagseiten](#) werden im Dokumentanfangsabschnitt erstellt.



Um die Eigenschaften des Dokuments zu bearbeiten, klicken Sie in der Titelleiste des Dokumentanfangsabschnitts auf den Hyperlink *Eigenschaften bearbeiten*. Daraufhin wird das Dialogfeld "Eigenschaften" des Dokumentanfangsabschnitts angezeigt (*Abbildung unten*). Dieses Dialogfeld hat drei Register: (i) eines für grundlegende Seitenlayouteigenschaften, (ii) eines für (HTML)-Dokumenteigenschaften, und eines für (iii) Dokumentstile.



Seitenlayouteigenschaften

Die Seitenlayouteigenschaften für den Dokumentanfangsabschnitt gelten für den ersten Dokumentabschnitt des Dokuments; in Dokumenten mit nur einem Dokumentabschnitt gelten

Sie für das gesamte Dokument. Wenn ein neuer Dokumentabschnitt angelegt wird, wird er mit den Seitenlayouteigenschaften erstellt, die der Dokumentanfangsabschnitt zu diesem Zeitpunkt hatte. Die Eigenschaften des neuen Dokumentabschnitts können anschließend bearbeitet werden. Wenn eine Eigenschaft des Dokumentanfangsabschnitts geändert wird, so wird diese Änderung in den bereits bestehenden anderen Dokumentabschnitten nicht übernommen. Neue Dokumentabschnitte werden mit den letzten Werten des Dokumentanfangsabschnitts erstellt. Die verschiedenen Seitenlayouteigenschaften sind im Abschnitt Seitenlayouteigenschaften beschrieben.

HTML-Dokumenteigenschaften

Die Eigenschaften des HTML-Ausgabedokuments werden auf dem Register "Dokumenteigenschaften" definiert.

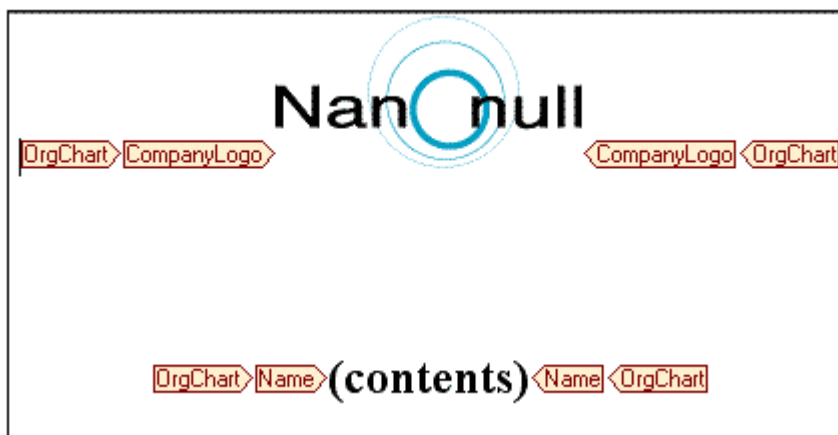
Dokumentstile

Die auf dem Register "Dokumentstile" definierten Stile gelten für das gesamte Dokument. Wenn ein Dokument mehr als einen Dokumentabschnitt hat, so erben die Design-Elemente in den einzelnen Dokumentabschnitten die Stileigenschaften aus dem Dokumentanfangsabschnitt. Um in einem bestimmten Design-Element die vererbten Stile außer Kraft zu setzen, definieren Sie die gewünschten Stilwerte für die einzelnen Design-Elemente. Klicken Sie dazu auf das Design-Element und definieren Sie in der Seitenleiste "Stil" die gewünschten Stile.

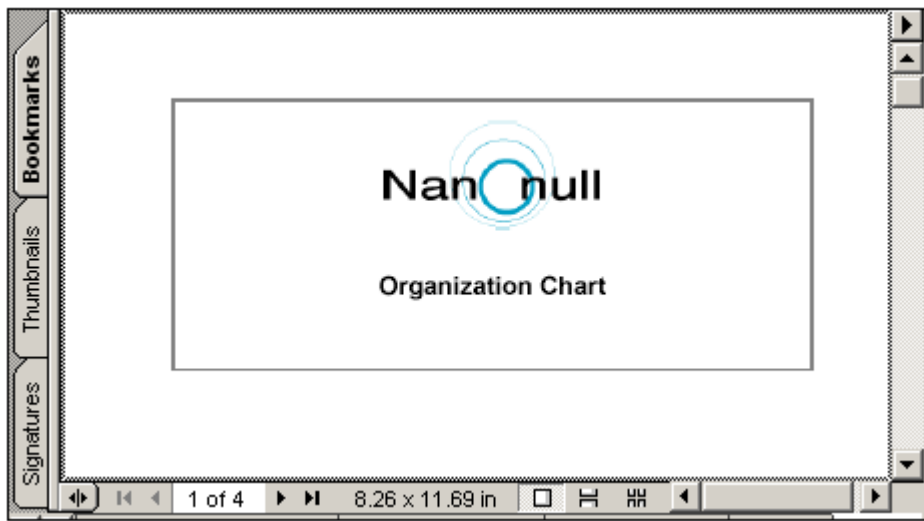
Umschlagseiten

Falls eine Umschlagseite benötigt wird, sollte diese zu Beginn des Dokumentanfangsabschnitts erstellt werden. Um sicherzustellen, dass der Rest des Dokuments auf einer neuen Seite beginnt, fügen Sie unterhalb der Umschlagseitenvorlage einen Seitenumbruch ein (**Einfügen | Seite/Spalte/Dokumentabschnitt einfügen | Neue Seite**). Wenn die Seitenlayouteigenschaften der Umschlagseite andere sein sollen, als die der folgenden Seiten, so sollte der gesamte Dokumentanfangsabschnitt für die Umschlagseite verwendet werden. Die folgenden Seiten sollten dann mit einem neuen Dokumentabschnitt beginnen.

Unten sehen Sie ein Beispiel.



Klicken Sie auf das Register RTF-Vorschau um das Ergebnis im Vorschaufenster zu sehen.



Kopf- und Fußzeilen: Teil 1

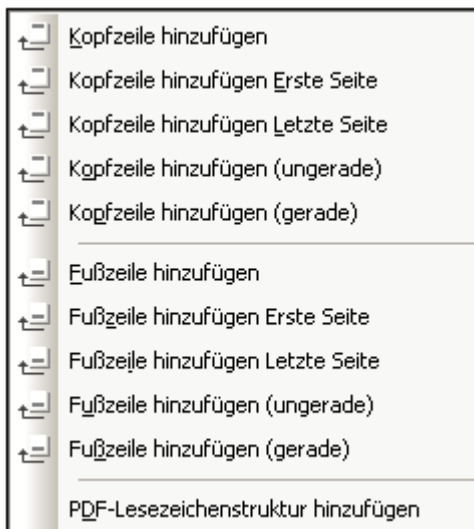
Kopf- und Fußzeilen können für **jeden einzelnen Dokumentabschnitt** einschließlich des Dokumentanfangsabschnitts hinzugefügt werden:

Hinzufügen einer Kopf- oder Fußzeile für einen Dokumentabschnitt

Um eine Kopf- oder Fußzeile für einen Dokumentabschnitt hinzuzufügen, klicken Sie im Design in [der Titelleiste des Dokumentanfangsabschnitts oder eines Dokumentabschnitts](#) auf den Link *Kopf-/Fußzeile hinzufügen* (Abbildung unten).

Dokumentabschnitt	Eigenschaften bearbeiten...	Kopf-/Fußzeile hinzufügen...	Kopf-/Fußzeilen ausblenden

Wählen Sie im angezeigten Menü (Abbildung unten) das gewünschte Element. Sie können Kopf- oder Fußzeilen separat für gerade und ungerade Seitenzahlen oder eine einzige Kopf- bzw. Fußzeile für alle Seiten hinzufügen. Außerdem kann mit den Befehlen (**Kopfzeile hinzufügen Erste Seite, Fußzeile hinzufügen Erste Seite**) bzw. (**Kopfzeile hinzufügen Letzte Seite, Fußzeile hinzufügen Letzte Seite, Nur PDF-Ausgabe**) für die erste Seite und/oder die letzte Seite eines Dokumentabschnitts eine separate Kopfzeile- bzw. Fußzeilenvorlage hinzugefügt werden. Dies ist nützlich, wenn die erste und/oder letzte Seite eines Dokumentabschnitts eine andere Kopfzeile bzw. eine andere Fußzeile erhalten soll: Wenn es sich bei der ersten Seite z.B. um eine Umschlagseite handelt, kann dafür z.B. eine leere Kopfzeilen- bzw. Fußzeilenvorlage erstellt werden.



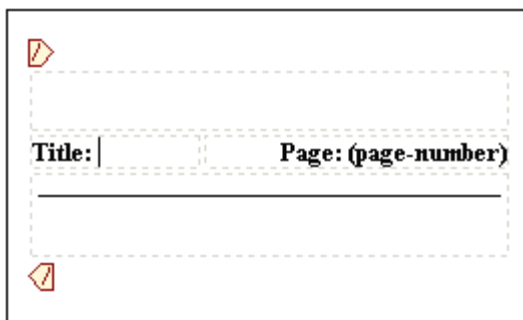
Bei Auswahl der gewünschten Kopf- bzw. Fußzeile wird in diesem Dokumentabschnitt in der [Design-Ansicht](#) eine Vorlage für die Kopf- bzw. Fußzeile erstellt. Sie können Kopf- und Fußzeilen in der Design-Ansicht in einem Dokumentabschnitt ein- und ausblenden. Klicken Sie dazu [im Dokumentanfangsabschnitt oder bei einem neuen Dokumentabschnitt](#) auf den Link *Kopf- und Fußzeilen ein-/ausblenden*, um diese auszublenden bzw. wieder einzublenden. (siehe Abbildung oben).

Anmerkung: Kopf- und Fußzeilen für die letzte Seite werden nur für die PDF-Ausgabe unterstützt.

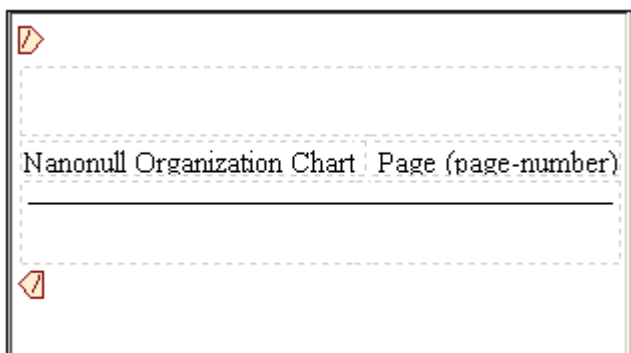
Erstellen der Kopf-/Fußzeile in der Design-Ansicht

Die Kopf-/Fußzeilenvorlage wird wie jede andere Vorlage erstellt. Komponenten können aus der Schema-Struktur in das Design gezogen oder statisch eingegeben und anschließend mit einem

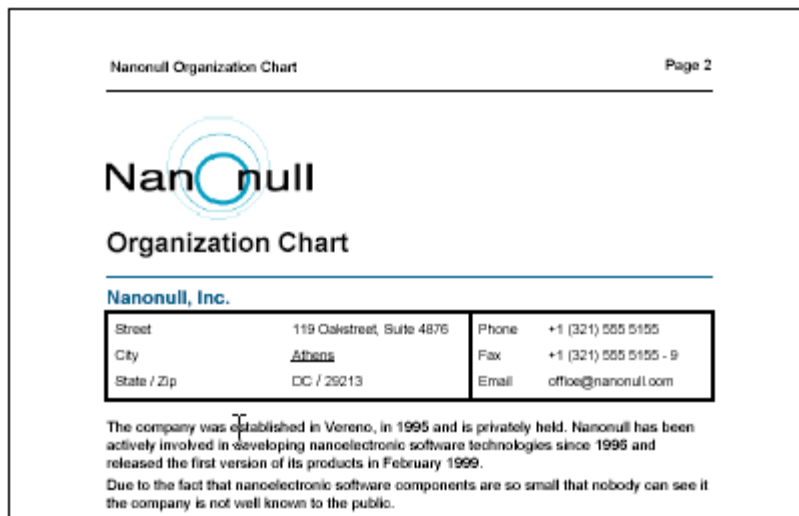
Stil versehen werden. Unten sehen Sie ein Beispiel. Wenn eine Kopfzeile hinzugefügt wird, sieht die Vorlage ungefähr folgendermaßen aus:



Ändern Sie die Kopfzeile Ihren Wünschen gemäß. Beachten Sie, dass Sie sowohl statischen als auch dynamischen Inhalt und sogar Bilder verwenden können.



Klicken Sie auf das Register RTF-Vorschau, um das Ergebnis in den Vorschaufenstern zu sehen. In der Abbildung unten sehen Sie Seite 2 des Dokuments "Organization Chart" im Fenster "PDF-Vorschau", wobei die Kopfzeile wie oben gezeigt definiert ist. Um in der RTF-Ausgabe eine Nummerierung anzuzeigen, müssen Sie in MS Word **Bearbeiten | alle auswählen** (oder **Strg+A**) auswählen und anschließend die Taste **F9** drücken. Siehe auch Anzeige der Seitennummern in der RTF-Ausgabe.



Beachten Sie bitte die folgenden Punkte:

- Die Höhe der Kopf- und Fußzeile sollte nicht größer sein als der entsprechende Rand für den Textkörper (oben bzw. unten) minus die Höhe des Kopf- bzw. Fußzeilenrands (nähere Informationen siehe Seitenlayouteigenschaften). Die vertikale Höhe der Kopf- bzw. -Fußzeile wird folglich durch den oberen/unteren Rand des Textkörpers und den Rand für die Kopf-/Fußzeile bestimmt).
- Sie können eine Kopf-/Fußzeile entweder (i) für alle Seiten im Dokumentabschnitt oder (ii) für gerade und ungerade Seitenzahlen im Dokumentabschnitt separat definieren. Zusätzlich dazu können separate Kopf-/Fußzeilen für die erste und die letzte Seite eingefügt werden. (Nähere Informationen dazu finden Sie unter [Kopf- und Fußzeilen: Teil 2](#))
- Die Seitennummer beginnt mit der Nummer, die Sie in den Seitenlayouteigenschaften definieren.

Kopf- und Fußzeilen: Teil 2

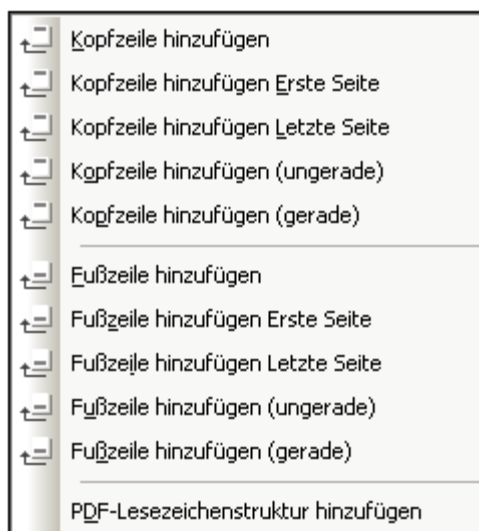
In diesem Abschnitt wird beschrieben, wie man die folgenden Arten von Kopf- und Fußzeilen erstellt:

- [Unterschiedliche Kopf-/Fußzeilen für Seiten mit geraden und ungeraden Nummern](#)
- [Unterschiedliche Kopf- /Fußzeilen für unterschiedliche Dokumentabschnitte](#)
- [Simulieren von Kopf-/Fußzeilen innerhalb einer Seite](#)
- [Kopf-/Fußzeilen mit Zwischensummen](#)

Unterschiedliche Kopf-/Fußzeilen für Seiten mit geraden und ungeraden Nummern

Sie können gerade und ungerade nummerierten Seiten in jedem Dokumentabschnitt unterschiedliche Kopf-/Fußzeilen zuweisen.

Um unterschiedliche Kopfzeilen für Seiten mit geraden und ungeraden Seitennummern zu erstellen, klicken Sie in der Titelleiste des entsprechenden Dokumentabschnitts auf den Link *Kopf-/Fußzeile hinzufügen* und wählen Sie aus dem Menü, das daraufhin angezeigt wird, die Befehle **Kopfzeile hinzufügen (ungerade)** und **Kopfzeile hinzufügen (gerade)** (*Abbildung unten*). Daraufhin werden zwei Kopfzeilenvorlagen erstellt - eine für ungerade nummerierte Seiten und eine für gerade nummerierte Seiten. Geben Sie nun den Inhalt der beiden Kopfzeilen in die Vorlagen ein.



Auf ähnliche Weise können Sie Fußzeilen für gerade und ungerade nummerierte Seiten erstellen.

Unterschiedliche Kopf-/Fußzeilen für unterschiedliche Dokumentabschnitte

Sie können unterschiedliche Kopf-/Fußzeilen für die einzelnen Abschnitte des Dokuments anlegen. Klicken Sie dazu in der [Titelleiste des Dokumentanfangsabschnitts oder eines Dokumentabschnitts](#) auf den Link *Kopf-/Fußzeile hinzufügen*. Daraufhin wird das in der obigen Abbildung gezeigte Menü "Kopf-/Fußzeile hinzufügen" angezeigt. Beachten Sie die folgenden Punkte:

- Sie können für gerade und ungerade nummerierte Seiten separate Kopf-/Fußzeilen hinzufügen. Oder Sie fügen für alle Seiten im Dokumentabschnitt eine gemeinsame Kopfzeile hinzu.
- Sie können eine zusätzliche Kopf-/Fußzeile für die erste und/oder letzte Seite hinzufügen. Diese Kopf-/Fußzeilen werden anstatt anderer Kopf-/Fußzeilen, die eventuell für diesen Dokumentabschnitt definiert wurden, auf der ersten und/oder

- letzten Seite des Abschnitts eingefügt.
- Die Seitennummerierung für den Dokumentabschnitt kann entweder fortlaufend sein und vom vorherigen Abschnitt aus fortgesetzt werden oder mit einer bestimmten Seitennummer beginnen.
- Die Gesamtseitenzahl ist die Anzahl der Seiten des gesamten Dokuments - nicht die des aktuellen Dokumentabschnitts.

Simulieren von Kopf-/Fußzeilen innerhalb einer Seite

Sie können Kopf- und Fußzeilen manuell in einem [Layout Container](#) erstellen. Dabei wird eine einzige Seite als Layout Container erstellt. Die Kopf- und Fußzeile wird innerhalb von [statischen Tabellen](#) erstellt, die sich am oberen bzw. unteren Rand der Seite befinden. Wenn mehrere Seite entworfen werden sollen, können mehrere Layout Container verwendet werden, wobei jeder diese Container vom nächsten durch einen [Seitenumbruch](#) getrennt wird (**Einfügen | Seite / Spalte / Dokumentabschnitt | Neue Seite**).

Kopf-/Fußzeilen mit Zwischensummen und laufenden Zwischensummen

Wenn ein Dokument eine Liste zu summierender numerischer Einträge enthält und wenn die Liste über mehrere Seiten fortläuft, müssen in der Kopf-/ und/oder Fußzeile jeder Seite eventuell Zwischensummen für jede Seite und/oder fortlaufende Zwischensummen angezeigt werden. Im Beispiel `Subtotals.sps` aus dem [Ordner \(Eigene\) Dateien](#), `C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/Subtotals\` sehen Sie, wie man laufende Zwischensummen erstellen und in Kopf- und Fußzeilen inkludieren kann.

Bei Entwerfen dieses SPS wurde nach der folgenden Strategie vorgegangen:

- Da sich die Liste in einer Tabelle befindet und da eine Tabelle nicht automatisch an die Größe einer gedruckten Seite angepasst werden kann, müssen Sie definieren, wie viele Zeilen auf einer Seite Platz haben. Diese Zahlen werden in Form von zwei Variablen angegeben, die in der Vorlage auf der obersten Ebene, nämlich der für die `$XML` Vorlage, definiert wurden; diese beiden Variablen erhalten die Namen `RowsOnFirstPage` und `RowsPerPage`.
- Die Seitenzahl wird durch Division der Gesamtanzahl aller Listeneinträge durch die Anzahl der Zeilen pro Seiten errechnet (wobei die unterschiedliche Anzahl der Zeilen auf der ersten Seite berücksichtigt wurde). Die Seitenanzahl wird in einer Variablen namens `CountOfPages` (definiert in der `$XML` Vorlage) gespeichert.
- Für die Sequenz 1 bis `$CountOfPages` wird eine benutzerdefinierte Vorlage erstellt. In dieser Vorlage wird eine statische Tabelle erstellt. Weiters werden in dieser Vorlage zwei Variablen definiert, anhand derer berechnet wird, welche Zeile jeweils die erste (`$RowFrom`) und die letzte Zeile (`$RowTill`) auf jeder Seite sein wird. Die Zeilen in der Tabelle werden von einer benutzerdefinierten Vorlage generiert, die die Einträge in der XML-Datei (`file` Elemente) auf Basis ihrer Position im Vergleich zu den `$RowFrom`- und `$RowTill` Werten auswählt. Wenn es sich bei der Position des `file` Elements um einen Ganzzahlwert handelt, der sich in dem durch die Werte der Variablen `$RowFrom` und `$RowTill` begrenzten Bereichs befindet, so wird für das aktuelle `file` Element eine Zeile generiert.
- Die laufenden Zwischensummen werden mit Hilfe automatischer Berechnungen generiert und in die Zeilen am oberen und unteren Rand der Tabellen eingefügt. Beachten Sie, dass sich die XPath-Ausdrücke zur Berechnung der laufenden Zwischensummen am oberen und unteren Seitenrand voneinander unterscheiden.
- Die Kopf- und Fußzeilen werden in Tabellen oberhalb bzw. unterhalb der Haupttabelle auf der Seite erstellt. Die automatischen Berechnungen zum Generieren der laufenden Zwischensummen werden in die Kopf- und Fußzeilenvorlage eingefügt.
- Am Ende jeder Seite wird ein Seitenumbruch eingefügt.

Zusammenhalten und Umbrechen

In PDF-Dokumenten (*nur Enterprise Edition*) können in der Stilgruppe *XSL-FO* (Seitenleiste "Stile", *Abbildung unten*) Witzenzeilen und Umbrüche für Seiten und Spalten definiert werden. In dieser Stilgruppe können Sie festlegen, ob vor oder nach dem aktuellen Block im Design-Dokument (d.h. dem Block, in dem sich der Cursor gerade befindet) ein Seiten- oder Spaltenumbruch erfolgen soll oder ob der Block mit benachbarten Blöcken zusammengehalten werden soll. Außerdem können Sie mit Hilfe der `table-omit`-Eigenschaften definieren, ob Tabellenkopf- und Fußzeilen nach Seitenumbrüchen weggelassen (oder wiederholt) werden sollen. Nähere Informationen zu diesen Eigenschaften finden Sie in der XSL-FO-Spezifikation.



Für die Druckversion von HTML-Seiten können Einstellungen für Seitenumbrüche und Witzenzeilen über die entsprechenden Eigenschaften in der Stilgruppe *Paginierung* (*Abbildung oben*) vorgenommen werden.

Pixel-Auflösung

Wenn Sie in Ihrem SPS als Maßeinheit Pixel verwenden, sollten Sie sich darüber im Klaren sein, dass es sich bei in Pixeln definierten Längeneinheiten um eine Funktion der Bildschirmauflösung handelt. Die entsprechende absolute Länge im gedruckten Dokument könnte sich von der am Bildschirm angezeigten Länge beträchtlich unterscheiden. In diesem Abschnitt werden folgende Punkte erläutert:

1. Es wird erklärt, warum in Pixeln definierte Größen zwei Formen haben: (i) eine abstrakte in Pixeln definierte Form; (ii) eine tatsächliche Größe, die durch Auflösung der abstrakten Form unter Anwendung einer bestimmten Bildschirmauflösung berechnet wird.
2. Es werden StyleVision-Funktionalitäten für diese Zwecke beschrieben.

Von Pixeln in Punkte

Hier einige Erläuterungen zum besseren Verständnis der Faktoren, die sich auf die abstrakte und die tatsächliche Größe von in Pixeln definierten Längen auswirken:

- Bei Pixeln handelt es sich um eine relative Maßeinheit: Ihre Größe ist von der Bildschirmauflösung abhängig. Je höher die Auflösung, desto kleiner ist ein Pixel. Die Bildschirmauflösung wird in dpi (dots per inch) angegeben. Im Fall der Bildschirmauflösung ist ein Punkt ein Pixel. Wenn die Bildschirmauflösung also 72 dpi beträgt, dann enthält ein Zoll einer Pixelzeile 72 Pixel (Punkte). Wenn die Bildschirmauflösung 96 dpi beträgt, dann enthält ein Zoll 96 Pixel. Wie viele Pixel ein Zoll auf dem Bildschirm enthält, hängt aber von der Bildschirmauflösung ab. Die häufigsten Bildschirmauflösungen heutzutage sind 72 dpi, 96 dpi und 120 dpi. Je höher der dpi-Wert, desto kleiner sind die Pixel.
- Bei der als "Punkt" bezeichneten Maßeinheit handelt es sich um eine absolute Längeneinheit, die meist in der Druckereibranche verwendet wird: 72 Punkte ergeben einen Zoll.
- Aufgrund dessen entspricht ein Pixel nur bei einer Bildschirmauflösung von 72 dpi einem Punkt. Bei anderen Bildschirmauflösungen kann die absolute Länge berechnet werden. In der Tabelle unten ist die absolute Länge (in Punkten) von 100 px bei verschiedenen Bildschirmauflösungen aufgelistet.

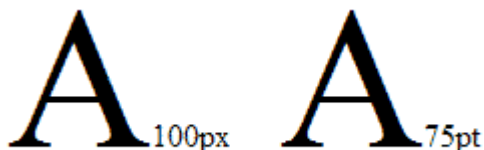
Auflösung	Pixel	Punkte	Faktor
72 dpi	100	100	1.00
96 dpi	100	75	0.75
120 dpi	100	60	0.60

Um Pixel für die einzelnen Bildschirmauflösungen in Punkte zu konvertieren, können wir einen Faktor anwenden, der das Verhältnis von Punkten in einem Zoll (72) zu Pixeln in einem Zoll (dpi) wiedergibt, d.h. 72 dividiert durch die dpi. Wenn Sie die Länge in Pixeln mit dem entsprechenden Konvertierungsfaktor multiplizieren, erhalten Sie die absolute Länge in Punkten. Da 72 Punkte einen Zoll ergeben, muss die Längen in Punkten durch 72 dividiert werden, um die Länge eines Pixels in Zoll zu erhalten. So sind z.B. 100 Punkte gleich $100 \div 72 \text{ Zoll} = 1,389 \text{ Zoll}$.

Bildschirmauflösung und absolute Länge

Im vorigen Abschnitt wurde erklärt, dass nur bei einer Bildschirmauflösung von 72 dpi die absolute Länge in Punkten identisch mit der Pixelanzahl ist, mit der diese Länge definiert ist. Die Bildschirmauflösung auf Windows-Systemen beträgt allerdings normalerweise nicht 72 dpi

sondern 96 dpi. Das bedeutet, dass die Anzahl der Punkte einer in Pixeln definierten Länge auf einem solchen Bildschirm 75% der Anzahl der Pixel beträgt. So haben z.B. die beiden Zeichen im Screenshot unten in der StyleVision Design-Ansicht 100 px bzw. 75 pt.



Der Grund dafür, dass sie dieselbe Höhe haben, ist, dass die Bildschirmauflösung, in der sie angezeigt werden, 96 dpi ist. In dieser Auflösung entsprechen 100px 75pt.

Beachten Sie daher bitte beim Definieren von Pixeln als Längeneinheit die Auflösung Ihres Bildschirms (normalerweise bei Windows-Systemen 96 dpi); Sie bestimmt die absolute am Bildschirm angezeigte Länge der in Pixeln definierten Länge.

Auflösung in der Druckausgabe

Wenn ein SPS in ein Druckformat wie PDF, RTF oder Word 2007+ transformiert wird, müssen nicht-absolute Pixellängen in absolute Längen wie z.B. Punkte oder Zoll konvertiert werden (da eine Länge auf Papier nicht in Pixeln definiert werden kann). Die Frage ist: Welcher Faktor (oder welche Bildschirmauflösung) soll man zur Konvertierung von Pixeln in Punkte verwenden?

In StyleVision können Sie die Ausgabe-dpi für jedes SPS einzeln auswählen. Wenn Sie z.B. 96 dpi auswählen, wird ein 100px-Zeichen in der Druckausgabe als ein 75pt-Zeichen dargestellt. Wenn Sie 72 dpi auswählen, so wird dasselbe Zeichen in der Druckausgabe als 100 pt-Zeichen dargestellt. Dieses System gilt für alle in Pixeln definierte Längen.

Hinweis: Die Konvertierung in eine absolute Maßeinheit wird für die HTML-Ausgabe nicht durchgeführt. Bei HTML werden die ursprünglichen Pixel unverändert an die HTML-Datei übergeben.

Einstellen der Auflösung für die Druckausgabe bei einem SPS

Um bei einem SPS die Auflösung für die Druckausgabe einzustellen, wählen Sie **Datei | Eigenschaften**. Daraufhin wird das Dialogfeld "Eigenschaften" angezeigt (*Abbildung unten*).

Ausgabekodierung auswählen für HTML: Unicode UTF-8 RTF: ISO 8859-1 (Latin-1/West European) PDF: Unicode UTF-8	Einbettung von Bildern ☒ Bilder für RTF und Word 2007 einbetten (XSLT 2 erforderlich) Wichtig: - Das RTF-Format unterstützt nur die Einbettung von Bildern in EMF-, JPG- und PNG-Dateien. - Die Generierung von RTF und Word 2007 steht in einigen Editionen von StyleVision nicht zur Verfügung.
HTML-Ausgabemodus auswählen ☒ HTML 4.01 Transitional ☐ XHTML 1.0 Transitional	Relative Bildpfade in Authentic auflösen relativ zu ☒ SPS-Datei ☐ XML-Arbeitsdatei der aktuellen Schemaquelle
Internet Explorer CSS-Unterstützung auswählen ☐ Kompatibilitätsmodus ☒ Standardkompatibler Modus (streng)	Auflösung Wenn eine CSS-Länge (wie z.B. Breite oder Höhe) in Pixeln [px] angegeben ist, hängt die tatsächliche Länge bei der Ansicht des Dokuments von der Auflösung des Bildschirms bzw. des Druckers ab. Eine CSS-Länge, die in absoluten Maßeinheiten (in cm, pt, usw.) angegeben wird, ist von dieser Option nicht betroffen. Generiere Dokumente für eine Bildschirmauflösung von 96 dpi 72 <input checked="" type="text" value="96"/> 120
XSLT-Version auswählen ☐ XSLT Version 1.0 ☒ XSLT Version 2.0 ☒ xsl:import-schema Deklaration verwenden	Standardeinfügemodus für Authentic ☒ Als XML einfügen ☐ Als Text einfügen
OK Abbrechen	

Wählen Sie im Bereich "Auflösung" die gewünschte Druckauflösung. Den Konvertierungsfaktor für die einzelnen aufgelisteten dpi-Optionen finden Sie in der Konvertierungstabelle weiter oben (im Abschnitt *Von Pixeln in Punkte*). Durch Multiplikation der Pixelanzahl mit dem Konvertierungsfaktor erhalten Sie die absolute Länge in Punkten. Beachten Sie, dass die Konvertierung in absolute Maßeinheiten nur auf Druckausgabeformate angewendet wird. In der HTML-Ausgabe wird die ursprüngliche Definition in Pixeln beibehalten. Beachten Sie außerdem, dass ihre Bildschirmauflösung durch diese Einstellung nicht geändert wird.

Kapitel 12

SPS-Datei: Zusätzliche Funktionen

12 SPS-Datei: Zusätzliche Funktionen

Zusätzlich zu den Funktionen zur [Bearbeitung](#), [Gliederung](#) und [Darstellung](#) von Inhalt sowie den [komplexen SPS-Funktionen](#) 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](#). Globale Ressourcen bieten Flexibilität beim Auswählen der Ressourcen. So können z.B. einem Alias mehrere Ressourcen (wie z.B. Dateien und Datenbanken) 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.
- [Authentic Node-Eigenschaften](#). Einzelne Nodes im XML-Dokument haben Eigenschaften, die für die Authentic-Ansicht spezifisch sind. Sie können festlegen, dass Nodes nicht editierbar sein sollen, dass Sie mit Markup-Tags angezeigt werden, dass beim Darüberplatzieren der Maus Benutzerinformationen angezeigt werden usw.
- [Parent Node bei Klick ersetzen durch](#). Der Wert des übergeordneten Nodes einer Schaltfläche oder eines Hyperlink kann vom Benutzer der Authentic-Ansicht ausgewählt werden. Das SPS kann so konzipiert werden, dass die Darstellung sich ändert, je nachdem, welchen Wert der Benutzer der Authentic-Ansicht auswählt.
 - [Zusätzliche Validierung](#). Ein Node kann mittels eines XPath-Ausdrucks getestet werden, sodass ein Boolescher Wert zurückgegeben wird, der angibt, ob die Eingabe des Benutzers für diesen Node gültig ist. Dieser Test erfolgt zusätzlich zur Dokumentvalidierung gegen ein Schema.
 - [Arbeiten mit Datumsangaben](#). In der Authentic-Ansicht stellt eine Datumswahlfunktion sicher, dass Datumsangaben im korrekten XML-Schema-Format eingegeben werden. Datumsangaben können außerdem bedarfsgemäß bearbeitet und formatiert werden.
 - URIs von ungeparsten Entities. 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 Scripts](#). 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](#). Ein HTML-Dokument kann in StyleVision importiert werden und anhand dieser Datei kann eine XML-, eine XSD- und eine SPS-Datei erstellt werden.
 - [Neu von XSLT-Datei](#). 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.

12.1 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.

Bei Verwendung einer globalen Ressource als Input können Sie mittels der Schaltflächen auf der Benutzeroberfläche jederzeit zwischen den verschiedenen Konfigurationen für die globale Ressource wechseln. So können z.B. bei Zuweisung eines XSLT Stylesheet zur Transformationen eines XML-Dokuments über eine globale Ressource 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.

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

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

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

Definieren globaler Ressourcen

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

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



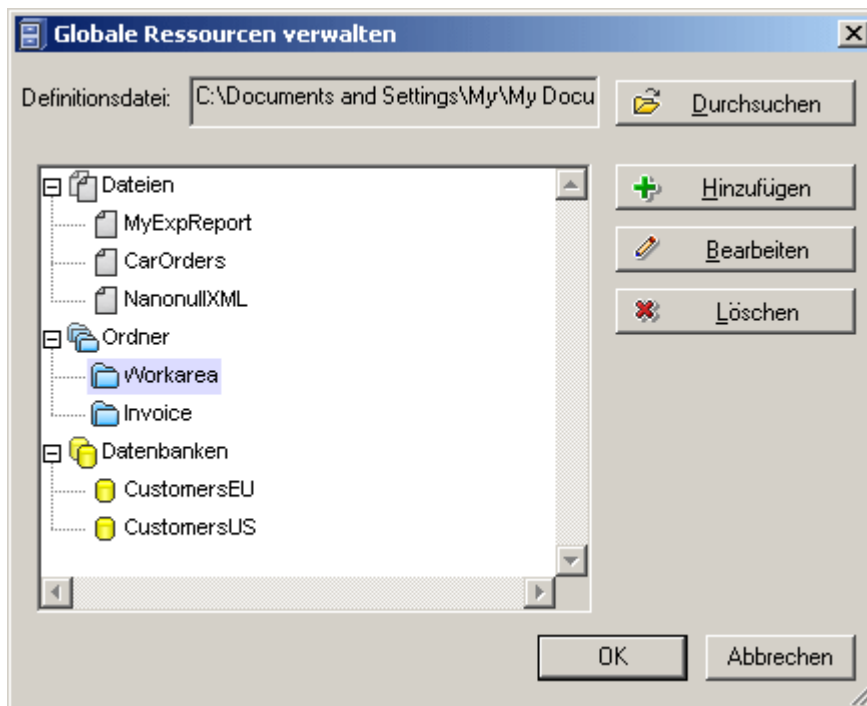
Sobald die Symbolleiste angezeigt wird, klicken Sie auf die Schaltfläche "Globale Ressourcen verwalten". Daraufhin wird das Dialogfeld "Globale Ressourcen verwalten" aufgerufen.

Die XML-Datei für globale Ressourcen

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

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

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



Verwalten globaler Ressourcen: Hinzufügen, Bearbeiten, Löschen

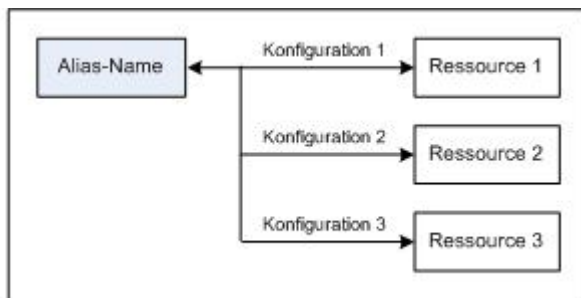
Im Dialogfeld "Globale Ressourcen verwalten" (Abbildung oben) können Sie eine globale Ressource zur ausgewählten XML-Datei für globale Ressourcen hinzufügen oder eine ausgewählte globale Ressource löschen oder bearbeiten. In der XML-Datei für globale Ressourcen werden die von Ihnen hinzugefügten Aliasse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

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

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

Hinzufügen einer globalen Ressource

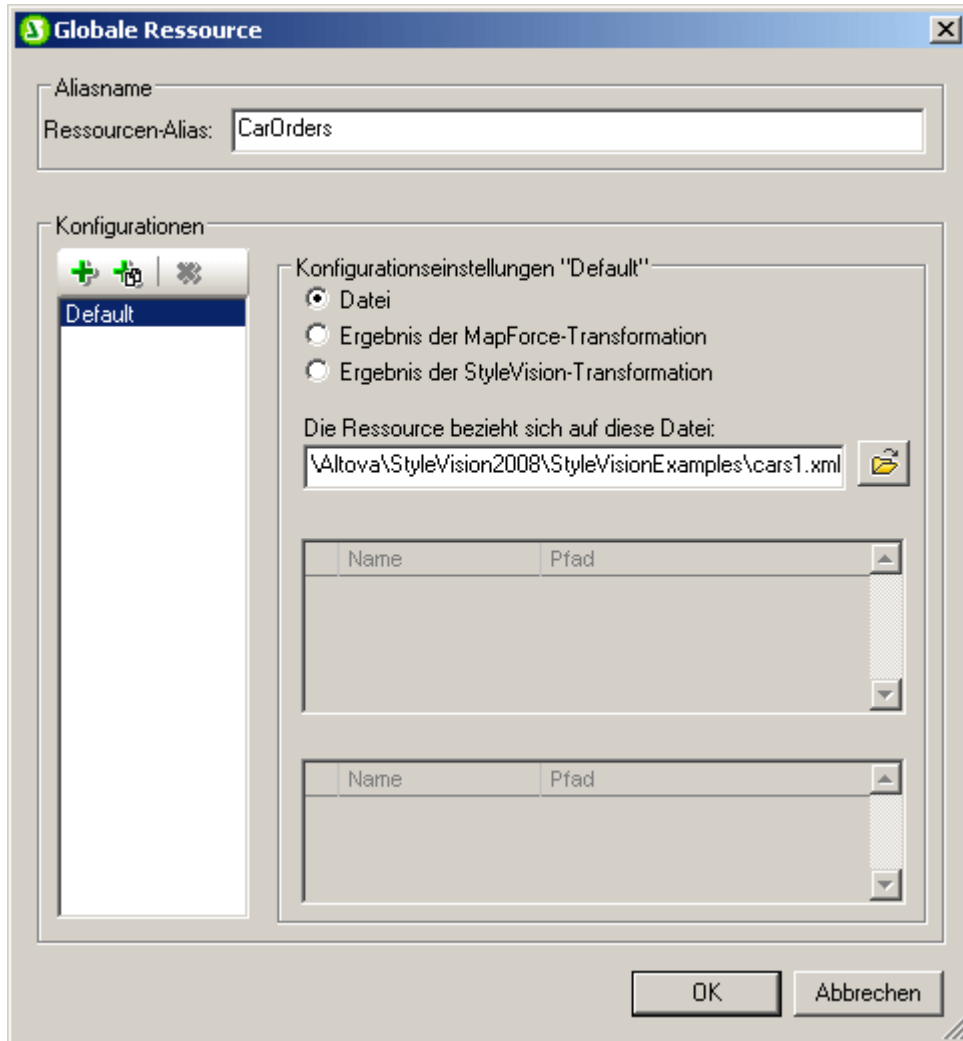
Beim Erstellen einer globalen Ressource wird ein Aliasname auf eine oder mehrere Ressourcen (Datei, Ordner oder Datenbank) gemappt. Jedes Mapping wird als Konfiguration bezeichnet. Ein einziger Aliasname kann daher mittels unterschiedlicher Konfigurationen mit mehreren Ressourcen verknüpft werden (Abbildung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



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

Dateien

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



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

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



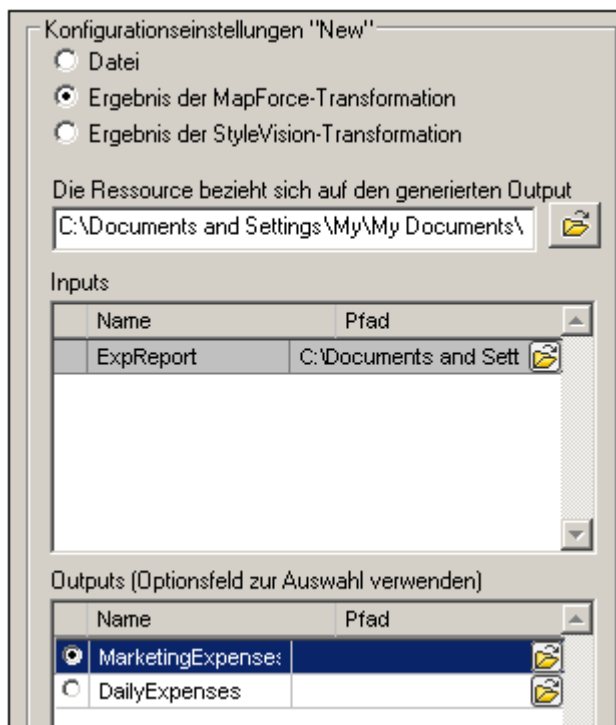
Aktivieren Sie das gewünschte Optionsfeld (in der Abbildung oben ist "HTML-Ausgabe" ausgewählt). Das Ergebnis einer Transformation kann selbst als globale Ressource oder als Dateipfad (über die Schaltfläche und anschließendes Auswählen von "Globale Ressource" oder "Durchsuchen") gespeichert werden. Wenn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

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

Auswählen des Ergebnisses einer MapForce-Transformation als globale Ressource

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

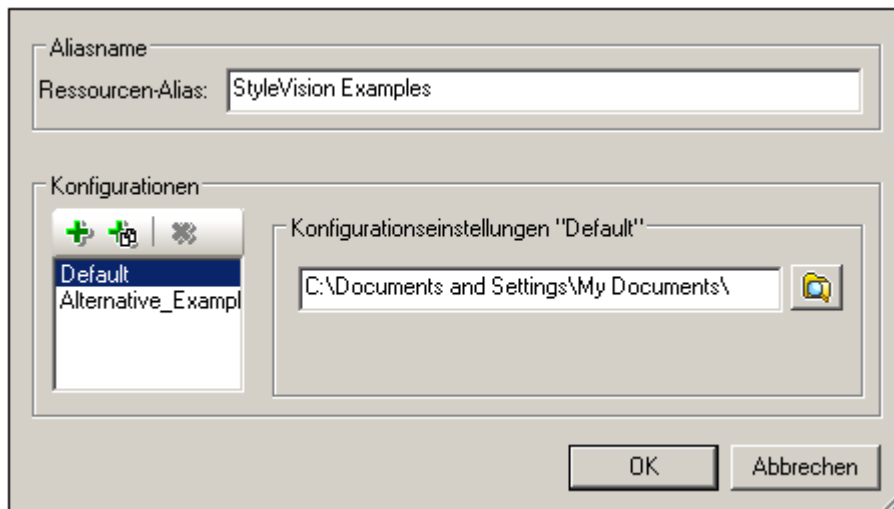
In einer MapForce-Transformation, die mehrere Ausgabeschemas hat, können Sie durch Klicken auf das entsprechende Optionsfeld auswählen, welches der Ausgabeschemas für die globale Ressource verwendet werden soll (*Abbildung unten*). Die XML-Datei, die für dieses Schema generiert wird, kann als globale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



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

Ordner

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

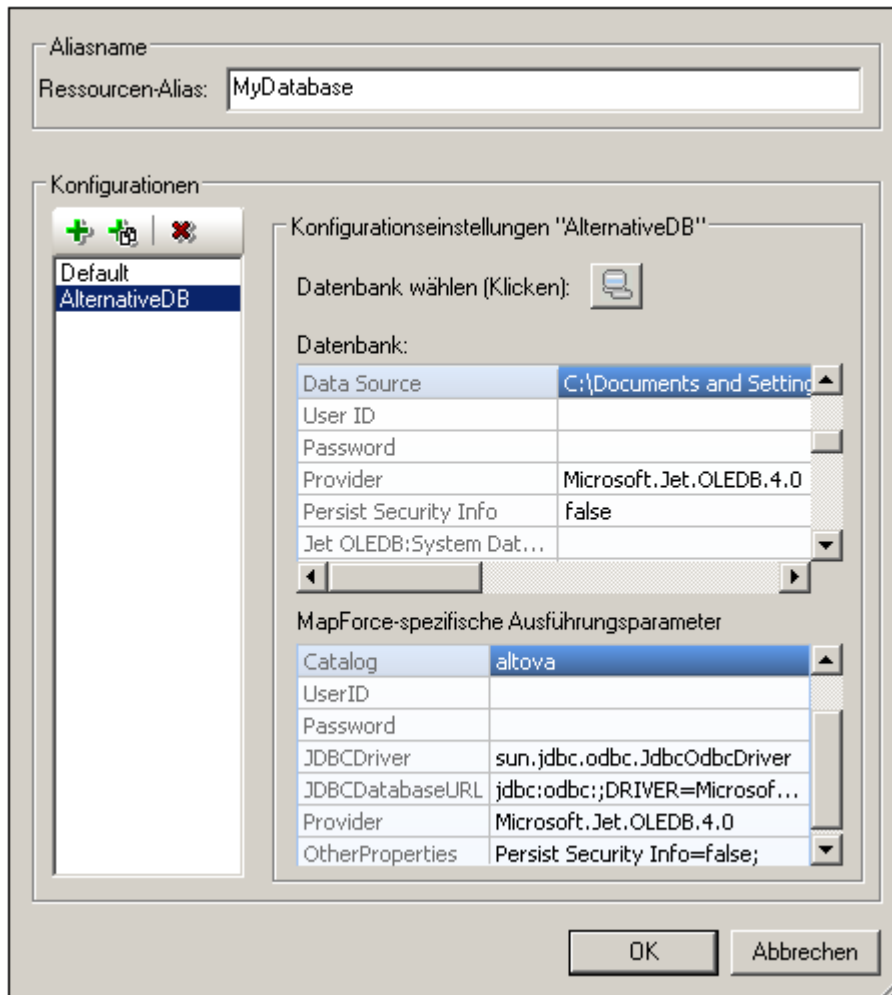


Geben Sie einen Aliasnamen ein.

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

Datenbanken

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



So definieren Sie eine globale Ressource vom Typ Datenbank:

1. Geben Sie einen Aliasnamen ein.
2. Im Fenster "Konfigurationen" finden Sie eine Konfiguration mit dem Namen "Default" (*Abbildung oben*). Diese Standardkonfiguration kann nicht gelöscht werden und der Name kann nicht geändert werden. Sie können beliebig viele zusätzliche Konfigurationen für den ausgewählten Alias hinzufügen. Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen**  und geben Sie im angezeigten Dialogfeld **Konfiguration hinzufügen** den Konfigurationsnamen ein. Klicken Sie auf **OK**. Die neue Konfiguration wird im Fenster "Konfigurationen" aufgelistet. Wiederholen Sie diesen Schritt um so viele Konfigurationen für diesen Alias (globale Ressource) hinzuzufügen, wie erforderlich.
3. Wählen Sie im Fenster "Konfigurationen" eine der Konfigurationen aus und klicken Sie auf die Schaltfläche **Datenbank wählen**. Daraufhin wird das Dialogfeld Verbindung zu Globaler Ressource erstellen angezeigt (*Abbildung unten*).
- 4.



Wählen Sie aus, ob Sie die Verbindung zur Datenbank mit dem Verbindungsassistenten herstellen, eine bestehende Verbindung verwenden möchten oder eine ADO-Verbindung oder eine ODBC-Verbindung herstellen möchten. Stellen Sie die Definition der Verbindungsmethode fertig, wie im Abschnitt [Herstellen einer Verbindung zu einer Datenbank](#) beschrieben. Sie können den [Verbindungsassistenten](#), [ADO-Verbindungen](#) oder [ODBC-Verbindungen](#) verwenden.. Wenn von StyleVision aus bereits eine Verbindung zu einer Datenbank hergestellt wurde, können Sie auf die Schaltfläche "Bestehende Verbindungen" klicken und die Datenbank aus der angezeigten Liste der Verbindungen auswählen.

4. Bei Herstellung einer Verbindung zu einem Datenbankserver werden Sie aufgefordert, ein Datenbank-Root-Objekt auf dem Server auszuwählen. Das ausgewählte Root-Objekt wird dann bei Verwendung dieser Konfiguration als Root-Objekt geladen. Wenn Sie kein Root-Objekt auswählen, können Sie dies nachholen, wenn die globale Ressource geladen wird.
5. Geben Sie für jede Konfiguration eine Datenbankressource an (d.h. wiederholen Sie Schritt 3 und 4 für die verschiedenen von Ihnen erstellten Konfigurationen).
6. Klicken Sie im Dialogfeld **Globale Ressource verwalten** auf **OK**, um den Alias und alle seine Konfigurationen als globale Ressource zu speichern. Die globale Ressource wird im Dialogfeld Globale Ressourcen verwalten unter "Datenbanken" aufgelistet.

Kopieren von Konfigurationen

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

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

Verwendung globaler Ressourcen

Es gibt mehrere Arten von globalen Ressourcen (Datei, Ordner und Datenbank). In bestimmten Szenarios können Sie in StyleVision bestimmte Arten von globalen Ressourcen verwenden. So können Sie z.B. für eine XML-Arbeitsdatei oder eine CSS-Datei globale Ressourcen vom Typ Datei oder Ordner verwenden. Oder Sie können mit Hilfe einer Ressource vom Typ Datenbank ein neues datenbankbasiertes SPS erstellen. Die verschiedenen Szenarios, in denen Sie globale Ressourcen in StyleVision verwenden können, sind im Unterabschnitt dieses Abschnitts aufgelistet: [Dateien und Ordner](#) und [Datenbanken](#).

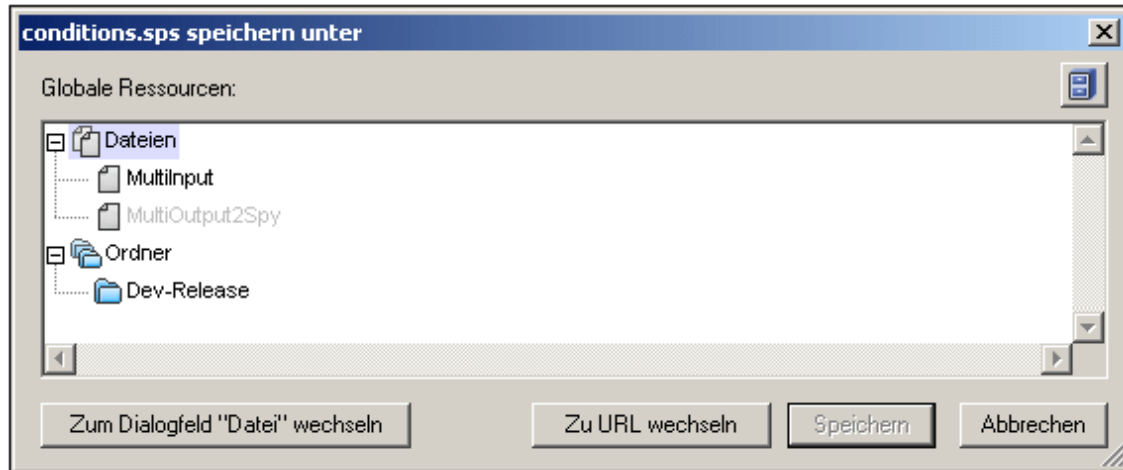
Auswahloptionen, die festlegen, welche Ressource verwendet wird

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

- *Die aktive XML-Datei für globale Ressourcen* wird im Dialogfeld "Globale Ressource" ausgewählt. Die globalen Ressource-Definitionen, die in der aktiven XML-Datei für globale Ressourcen enthalten sind, stehen 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](#) oder über die [Symbolleiste "Globale Ressourcen"](#) ausgewählt. Bei Auswahl dieses Befehls wird eine Liste der Konfigurationen für alle Aliasse angezeigt. Bei Auswahl einer Konfiguration wird diese in der gesamten Applikation aktiv. D.h. immer wenn eine globale Ressource (oder ein Alias) verwendet wird, wird die Ressource, die der aktiven Konfiguration der einzelnen verwendeten Aliasse entspricht, geladen. Die aktive Konfiguration wird auf alle verwendeten Aliasse angewendet. Wenn ein Alias keine Konfiguration mit dem Namen der aktiven Konfiguration hat, wird die Standardkonfiguration dieses Aliasses verwendet. Die aktive Konfiguration spielt beim Zuweisen von Ressourcen keine Rolle; Sie ist nur dann von Bedeutung, wenn die Ressourcen tatsächlich verwendet werden.

Zuweisen von Dateien und Ordnern

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



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

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

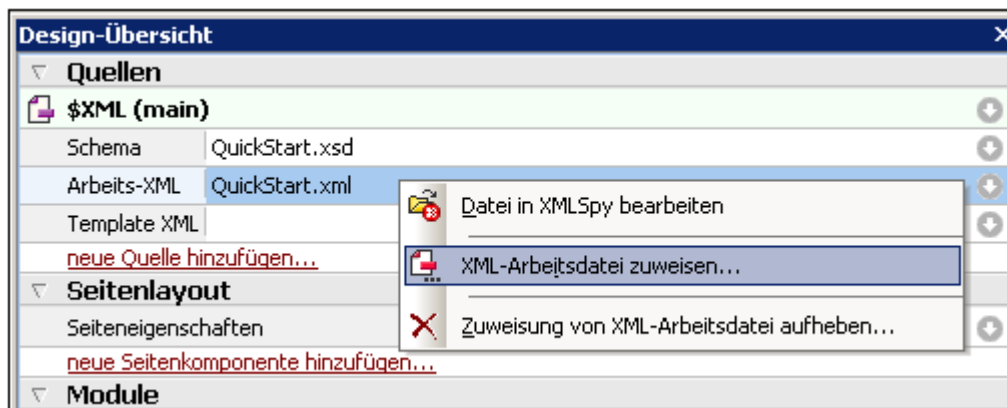
Verwendungsszenarios

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

- [Hinzufügen und Ändern von Schemaquellen und XML-Arbeitsdateien und XML-Vorlagendateien](#)
- [Speichern als globale Ressource](#)
- [Hinzufügen von Modulen und CSS-Dateien](#)
- [Hinzufügen von globalen Ressourcen zu einem Projekt](#)

Schema, XML-Arbeitsdatei, XML-Vorlagendatei

In der Seitenleiste "Design-Übersicht" (*Abbildung unten*) enthält das Kontextmenü für das Schema die XML-Arbeitsdatei und die XML-Vorlagendatei einen Befehl, mit dem das Dialogfeld "Öffnen" aufgerufen wird, in dem Sie das [Schema](#) oder die XML-Arbeitsdatei über eine globale Ressource zuweisen können. Wenn Sie auf den Befehl **Zu globalen Ressourcen wechseln** 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](#).)



Wenn eine globale Ressource als Dateiquelle ausgewählt wurde, wird sie im entsprechenden Eintrag in der Seitenleiste "Design-Übersicht" angezeigt (*Abbildung unten*).

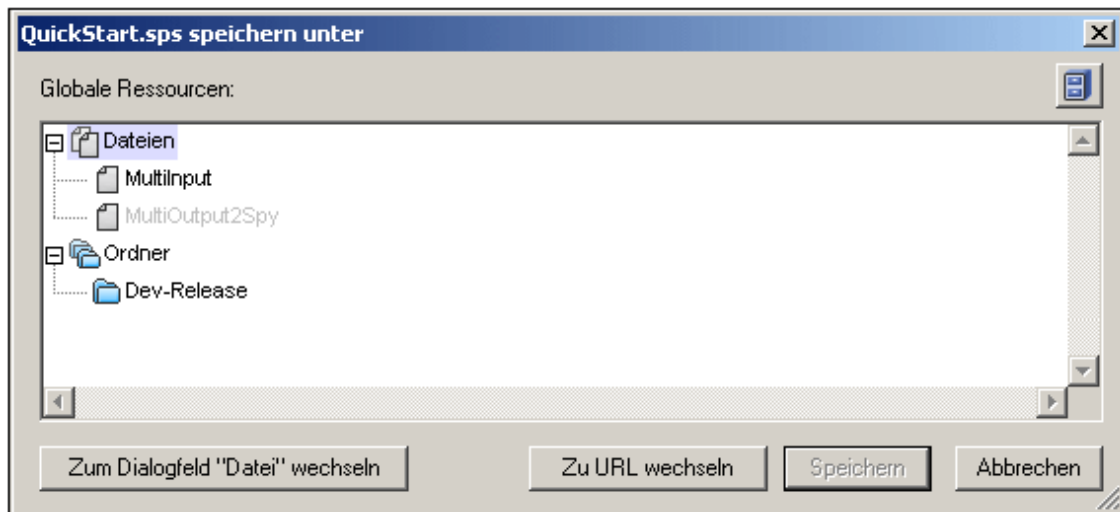


Hinzufügen von Modulen und CSS-Dateien aus einer globalen Ressource

Mit dem Befehl **Neues Modul hinzufügen** und **Neue CSS-Datei hinzufügen** aus der Seitenleiste "Design-Übersicht" wird das Dialogfeld "Öffnen" aufgerufen, in dem Sie auf "Zu globalen Ressourcen wechseln" 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 **Zu globalen Ressourcen wechseln**, um die verfügbaren globalen Ressourcen (*Abbildung unten*) aufzurufen. Dies sind die Aliasnamen, die in der aktuellen XML-Datei für globale Ressourcen gespeichert sind.



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

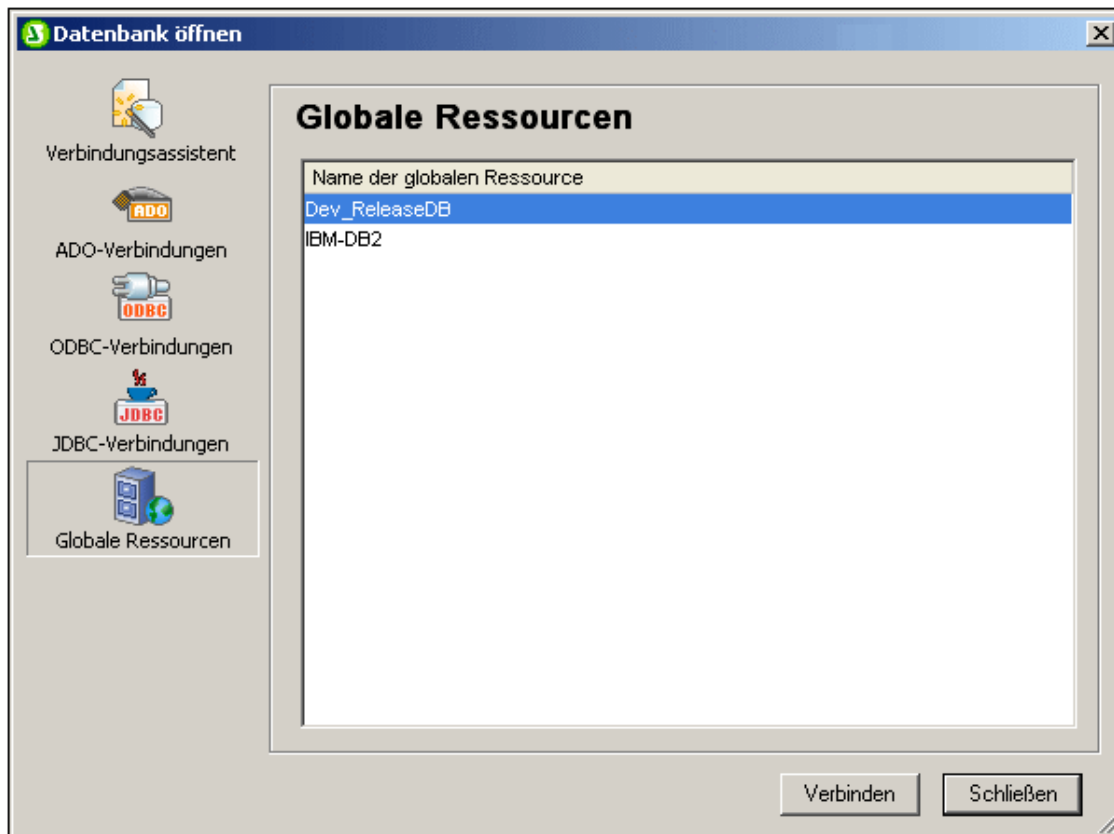
Anmerkung: Jede Konfiguration verweist auf einen bestimmten Dateipfad, der in der Definition dieser Konfiguration festgelegt ist. Wenn die Datei, die Sie als globale Ressource speichern, nicht dieselbe Dateityperweiterung wie die Datei unter dem aktuellen Pfad der Konfiguration hat, kann es beim Öffnen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

Zuweisen von Datenbanken

Wenn ein SPS mit Hilfe des Befehls **Datei | Neu von DB** anhand einer Datenbank erstellt wird, können Sie die Option zur Verwendung einer globalen Ressource auswählen (*Abbildung unten*).



Wenn Sie im Dialogfeld "Datenbank öffnen" auf die Schaltfläche "Globale Ressourcen" klicken, werden alle globalen Ressourcen vom Typ Datenbank, die in der derzeit aktiven [XML-Datei für globale Ressourcen](#) definiert wurden, angezeigt. Wählen Sie die gewünschte globale Ressource aus und klicken Sie auf **Verbindung herstellen**. Wenn die ausgewählte globale Ressource mehr als eine Konfiguration hat, wird die Datenbankressource für die gerade aktive Konfiguration (überprüfen Sie diese unter **Extras | Aktive Konfiguration** oder über die Symbolleiste "Globale Ressourcen") verwendet und die Verbindung wird hergestellt. Sie müssen nun auswählen, welche Datenstrukturen und Daten verwendet werden sollen (Beschreibung siehe Auswählen der Datenbankdaten).

Siehe auch:

[Definieren globaler Ressourcen](#): Hier finden Sie Informationen zum Definieren globaler Ressourcen.

[Extras | Globale Ressourcen](#): Hier wird der Menübefehl zum Aufrufen des Dialogfelds "Globale Altova-Ressourcen" beschrieben.

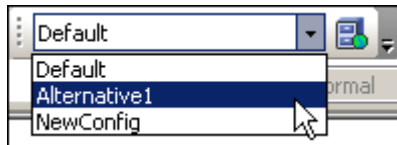
[Extras | Aktive Konfiguration](#): Hier wird der Menübefehl beschrieben, mit dem die aktive Konfiguration der Applikation geändert werden kann.

Wechseln von Konfigurationen

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

Betrachten wir als Beispiel, wie man eine Konfiguration wechselt, einen Fall, in dem einem SPS mittels einer globalen Ressource mit mehreren Konfigurationen eine XML-Arbeitsdatei zugewiesen wurde. Die XML-Arbeitsdatei kann gewechselt werden, indem Sie einfach die Konfiguration der globalen Ressource ändern. Dies kann auf zwei Arten erfolgen:

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



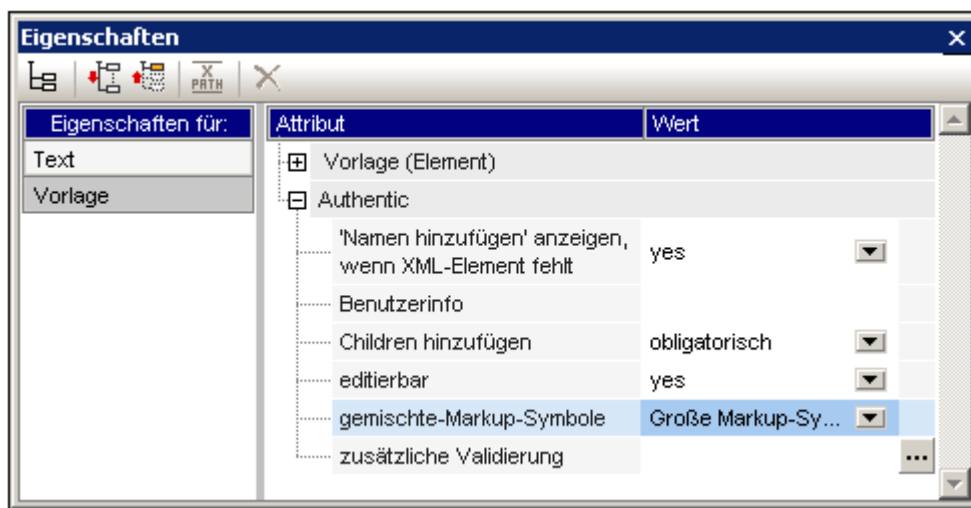
Die XML-Arbeitsdatei wird sofort gewechselt.

Auf diese Art können Sie durch Ändern der aktiven Konfiguration Quelldateien auswechseln, die über eine globale Ressource zugewiesen sind. Beachten Sie jedoch, dass die neu ausgewählten Dateien vom Schema her mit dem SPS übereinstimmen müssen, damit das SPS korrekt funktioniert.

12.2 Authentic Node-Eigenschaften

Authentic Node-Eigenschaften sind Eigenschaften, die Sie für die Anzeige in der **Authentic -Ansicht** definieren. So kann ein Node z.B. mit XML Markup-Tags angezeigt werden, als nicht editierbar definiert werden, oder so konfiguriert werden, dass Benutzerinformationen angezeigt werden, wenn der Cursor über den Node platziert wird. Authentic Node-Eigenschaften können für verschiedene SPS Komponenten definiert werden. Welche Eigenschaften für eine Komponente verfügbar sind, hängt vom Typ der Komponente ab.

Um Authentic Node-Eigenschaften zuzuweisen, wählen Sie die gewünschte Komponente im Design aus. Definieren Sie anschließend in der Seitenleiste "Eigenschaften" in der Eigenschaftsgruppe *Authentic* (Abbildung unten) die gewünschten Einstellungen für den Authentic Node. Alternativ dazu können Sie mit der rechten Maustaste auf die Node-Vorlage klicken und den Befehl **Authentic-Eigenschaften bearbeiten** auswählen.



Definieren der Authentic-Eigenschaften

Um das Verhalten einzelner Nodes in der Authentic-Ansicht zu steuern, können die folgenden Node-Einstellungen definiert werden:

Children hinzufügen

Diese Einstellung ist verfügbar, wenn der ausgewählte Node ein Element ist. Sie können damit definieren, welche Child-Elemente beim Einfügen des ausgewählten Elements eingefügt werden. Die Optionen sind: alle Child-Elemente, obligatorische Child-Elemente und keine Child-Elemente.

Inhalt ist editierbar

Definiert, ob der Node bearbeitet werden kann oder nicht. Standardmäßig ist der Node editierbar. Diese Einstellung ist verfügbar, wenn es sich beim ausgewählten Node um ein Element, Attribut oder Inhalt handelt. Das Ergebnis der automatischen Berechnung kann nicht bearbeitet werden, da der Wert mit einem XPath-Ausdruck berechnet wird, den Sie für die automatische Berechnung eingeben; diese Option ist daher für automatische Berechnungen nicht verfügbar.

Gemischte Markup-Symbole

Diese Einstellung ist verfügbar, wenn es sich beim ausgewählten Node um ein Element oder Attribut handelt. Sie können damit festlegen, mit welchen Markup-Tags einzelne Nodes im Authentic-Ansichtsmodus "gemischte Markup-Symbole" versehen werden. Es stehen die folgenden Optionen zur Verfügung: Große Markup-Symbole (Tags mit Node-Namen); kleine

Markup-Symbole (Tags ohne Node-Namen) und keine Markup-Symbole.

"Namen hinzufügen" anzeigen, wenn XML-Element fehlt

Legt fest, ob eine Eingabeaufforderung ("[Element/Attributnamen] hinzufügen") in der Authentic-Ansicht angezeigt werden soll, wenn das ausgewählte Element oder Attribut fehlt.

Standardmäßig wird die Eingabeaufforderung angezeigt. Diese Einstellung ist verfügbar, wenn der ausgewählte Node ein Element oder Attribut ist.

Benutzerinfo

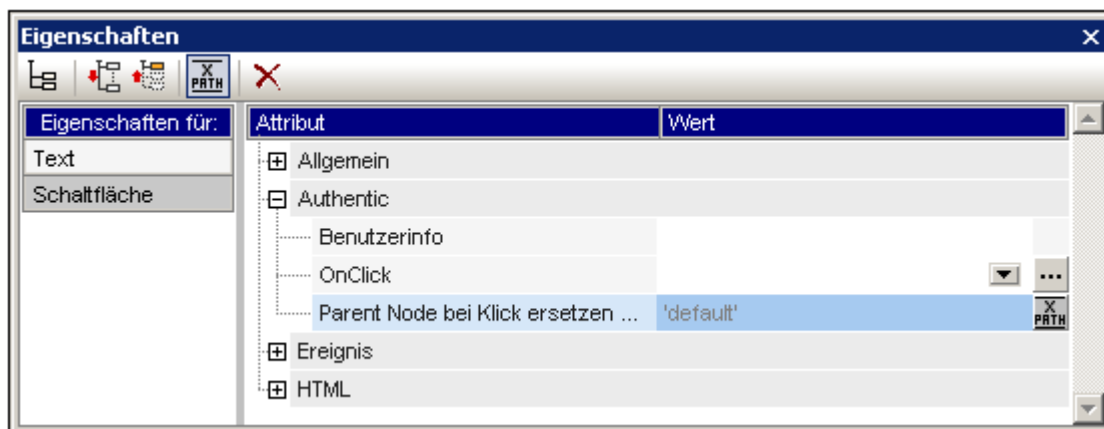
Text, der in diesem Textfeld eingegeben wird, wird als Tooltip angezeigt, wenn der Mauszeiger über den Node platziert wird. Die Option steht zur Verfügung, wenn es sich beim ausgewählten Node um ein Element, ein Attribut, Inhalt oder eine automatische Berechnung handelt. Wenn sowohl der Element/Attribut-Node als auch der Inhalt-Node eine Benutzerinfo hat, wird die Benutzerinfo für den Inhalt-Node als Tooltip angezeigt, wenn der Mauszeiger über den Node platziert wird.

Maximale Anzahl an anzuzeigenden Datensätzen

Diese Node-Einstellung gilt für Nodes, die eine Datenbanktabelle darstellen, d.h. Row Elemente, die Child-Elemente eines Tabellenelements auf oberster Ebene sind. Der hier angegebene Wert definiert die Anzahl der Datensätze, die in der Tabelle angezeigt werden.

12.3 Parent Node bei Klick ersetzen durch

Mit der Eigenschaft *Parent Node bei Klick ersetzen durch* (aus der Eigenschaftsgruppe "Authentic" in der Seitenleiste "Eigenschaften", siehe Abbildung unten) können Sie den Textinhalt des Parent Node einer [Schaltfläche](#) oder eines [Hyperlink](#) definieren.



Wenn der Benutzer der Authentic-Ansicht auf die Design-Komponente (eine Schaltfläche oder einen Hyperlink) klickt, wird der Text, den Sie als Verfasser des SPS definiert haben, als XML-Inhalt des Parent Node der Design-Komponente eingefügt. Auf diese Art können Sie dem Benutzer der Authentic-Ansicht die Möglichkeit geben, durch Klicken auf eine Schaltfläche oder einen Hyperlink eine bestimmte Auswahl zu treffen, durch die sich die Darstellung des XML-Dokuments in der Authentic-Ansicht ändert.

Verwendungsweise

Diese Funktion kann folgendermaßen eingesetzt werden:

1. Fügen Sie die Design-Komponente (Schaltfläche oder Hyperlink) im Design in den Node ein, in den der Textinhalt eingefügt werden soll.
2. Wählen Sie die Design-Komponente im Design aus und wählen Sie diese Design-Komponente anschließend in der Seitenleiste "Eigenschaften" (in der Spalte *Eigenschaften für*) aus und geben Sie für die Eigenschaft *Parent Node bei Klick ersetzen durch* (die sich in der Eigenschaftsgruppe "Authentic" befindet; siehe Abbildung oben) einen Wert ein. Dieser Wert kann entweder direkt als Textwert oder über einen XPath-Ausdruck eingegeben werden. (Um den XPath-Ausdruck einzugeben, klicken Sie in der Seitenleiste "Eigenschaften" auf die Schaltfläche "XPath"). In der Abbildung oben ist der Wert ein XPath-String-Ausdruck 'default'. Wenn der Benutzer der Authentic-Ansicht nun in der Authentic-Ansicht auf die Design-Komponente klickt, wird der Text `default` als der Textinhalt des Parent Node der Design-Komponente eingegeben. Dasselbe Ergebnis erzielen Sie, wenn Sie einen Textwert (nicht einen XPath-Ausdruck) von `default` direkt als den Wert der Eigenschaft *Parent Node bei Klick ersetzen durch* eingeben.
3. Jetzt, da Sie wissen, welchen Inhalt der Parent Node haben wird, wenn der Benutzer der Authentic-Ansicht auf die Design-Komponente klickt, können Sie auf Basis des Textinhalts des Parent Node eine Verarbeitung definieren. So können Sie z.B. mit Hilfe einer [Bedingung](#) definieren, dass eine bestimmte Standardformatierung auf den Inhalt des Parent Node angewendet wird, wenn dieser `default` lautet. Oder Sie können festlegen, dass der Text eines Node ausgeblendet wird, wenn der Inhalt `hide` lautet. Auf diese Art können Sie mit Hilfe dieser Funktion beeindruckende Effekte erzielen, die man sonst nur mit Hilfe von [Scripts](#) erzielen hätte können.

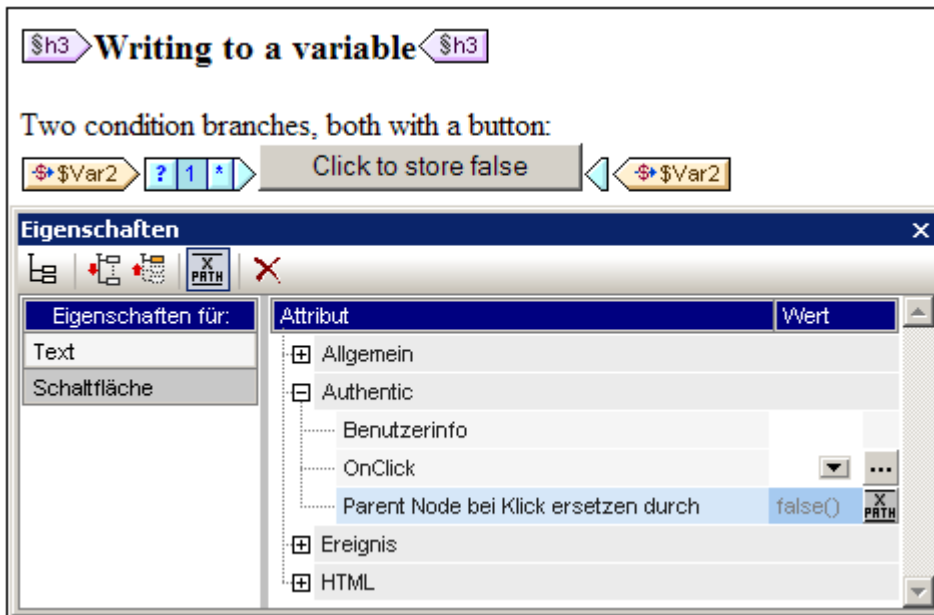
Anmerkung: Damit diese Funktion ordnungsgemäß funktioniert, muss im Schema definiert

sein, dass der Parent Textinhalt enthalten darf.

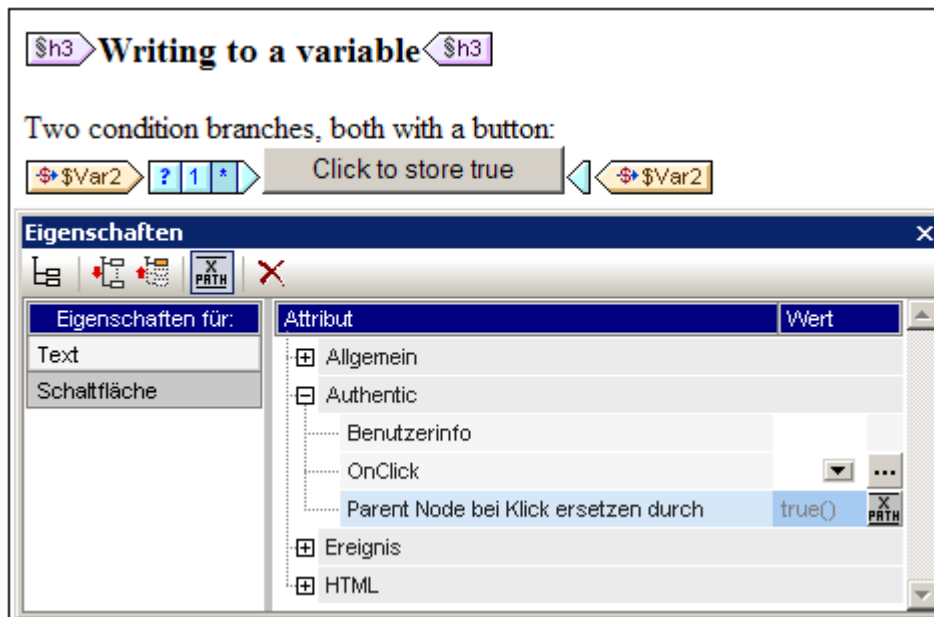
Verwendung mit Variablen

Beim Parent Node kann es sich auch um eine [Variable](#) handeln, die für einen darüberliegenden Node definiert wurde. Der Wert der Variablen kann dann vom SPS Designer mit Hilfe der Eigenschaft *Parent Node bei Klick ersetzen durch* definiert werden. Wenn der Benutzer der Authentic-Ansicht auf die Design-Komponente ([Schaltfläche](#) oder [Hyperlink](#)) klickt, wird der Variablen der für die Design-Komponenten-Eigenschaft *Parent Node bei Klick ersetzen durch* definierte Wert zugewiesen. Sie können nun auf Basis des Werts der Variablen alternative Verarbeitungsoptionen festlegen.

Unten sehen Sie ein einfaches Beispiel. Wenn der Wert der Variablen `Var2` auf Boolean `true` () gesetzt ist, wird eine Schaltfläche angezeigt, über die der Wert in Boolean `false` () geändert werden kann (siehe *Abbildung unten*). Diese Verarbeitung wird in der ersten Verzweigung einer [Bedingung](#) definiert.



Die *Andernfalls* Verzweigung der Bedingung definiert eine Verarbeitung, bei der eine Schaltfläche angezeigt wird, über die der Wert der Variablen in Boolean `true` () (*Abbildung unten*) geändert werden kann. Entsprechend dem Wert der Variablen `Var2` (`true` () oder `false` ()) können Sie zusätzliche Verarbeitungen definieren.



Auf diese Art können Sie als Verfasser des SPS dem Benutzer der Authentic-Ansicht die Möglichkeit geben, eine von mehreren Optionen auszuwählen, aufgrund derer dann eine entsprechende Verarbeitung für das Dokument erfolgt.

12.4 Zusätzliche Validierung

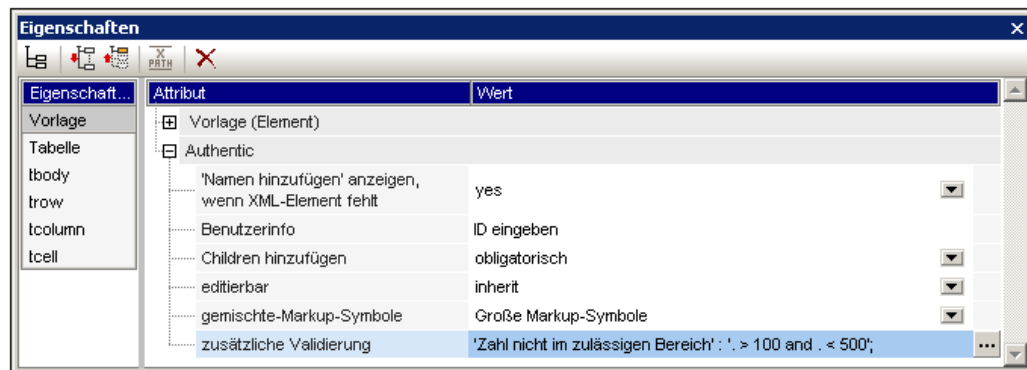
Die Einstellung "zusätzliche Validierung" ist verfügbar, wenn es sich bei der ausgewählten Komponente um einen Element- oder Attribut-Node, Inhalt (Platzhalter `contents`), ein Dateneingabeelement oder eine automatische Berechnung handelt. Sie können einen XPath-Ausdruck definieren, um die Gültigkeit des XML-Werts des Node oder der automatischen Berechnung zu definieren. Ein XML-Wert, der nicht in den definierten Bereich fällt, ist ungültig. Wenn der XML-Wert des Node ungültig ist, wird der Benutzer der Authentic-Ansicht bei der Validierung des XML-Dokuments (**F8**) mittels einer Fehlermeldung darauf aufmerksam gemacht. Die angezeigte Fehlermeldung wird von Ihnen in den Einstellungen für die zusätzliche Validierung im Feld "Fehlermeldung" eingegeben.

Konfigurieren der zusätzlichen Validierung

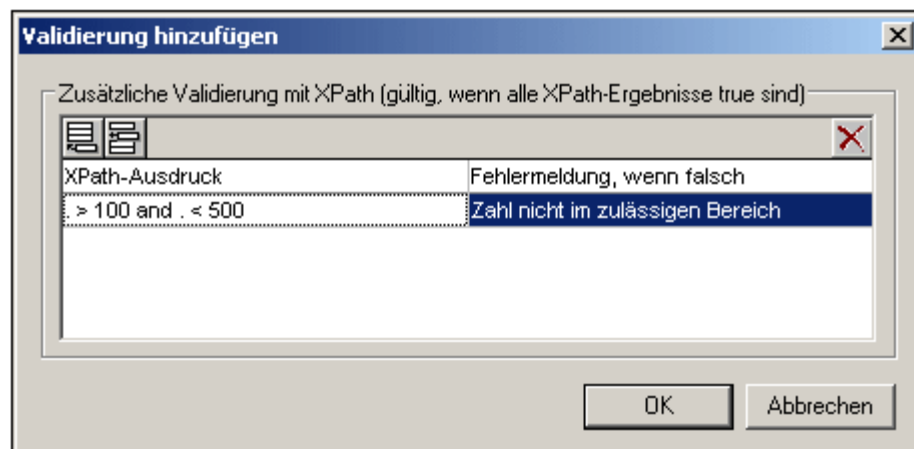
Um eine zusätzliche Validierung zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die Komponente für die eine zusätzliche Validierung erforderlich ist.
2. Wählen Sie in der Seitenleiste "Eigenschaften" die Eigenschaftsgruppe *Authentic* und

klicken Sie auf die **Bearbeiten**-Schaltfläche  der Eigenschaft "Zusätzliche Validierung" (Abbildung unten). Daraufhin wird das Dialogfeld "zusätzliche Validierung" angezeigt.



3. Fügen Sie im Dialogfeld "Zusätzliche Validierung" (Abbildung unten) eine Zeile für einen Eintrag für eine zusätzliche Validierung hinzu und klicken Sie auf die Schaltfläche **Hinzufügen** links oben im Fensterbereich.



4. Geben Sie in der Spalte "XPath-Ausdruck" einen XPath-Ausdruck ein, um den Validierungsbereich der XML-Daten in dieser Komponente zu definieren.

5. Geben Sie eine Fehlermeldung ein, die angezeigt werden soll, wenn die Daten ungültig sind.
6. Klicken Sie zum Fertigstellen auf **OK**.

12.5 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](#) angezeigt.
5. Sie können die SPS-Datei nun wie gewohnt bearbeiten. So könnten Sie z.B. Nodes aus der [Schema-Struktur](#) in das Design ziehen, die Stile und Darstellung ändern oder zusätzliche Stile hinzufügen sowie StyleVision-Funktionen wie z.B. [automatische Berechnungen](#) und [Bedingungsvorlagen](#) verwenden.
6. Sie können das SPS speichern und anhand einer [XML-Arbeitsdatei](#) eine Vorschau der [verschiedenen Ausgabeformate](#) anzeigen. Anschließend können Sie mit Hilfe des Befehls [Generierte Dateien speichern Stylesheets und Ausgabedateien generieren](#).

Beispiel

Das unten beschriebene Beispiel befindet sich im [Ordner \(Eigene\) Dateien](#), C:/Documents and Settings/<username>/My Documents/Altova/StyleVision2012/StyleVisionExamples/Tutorial/NewFromXSLT. Dieser Ordner enthält die Dateien: SimpleExample.xslt, SimpleExample.xsd und SimpleExample.xml.

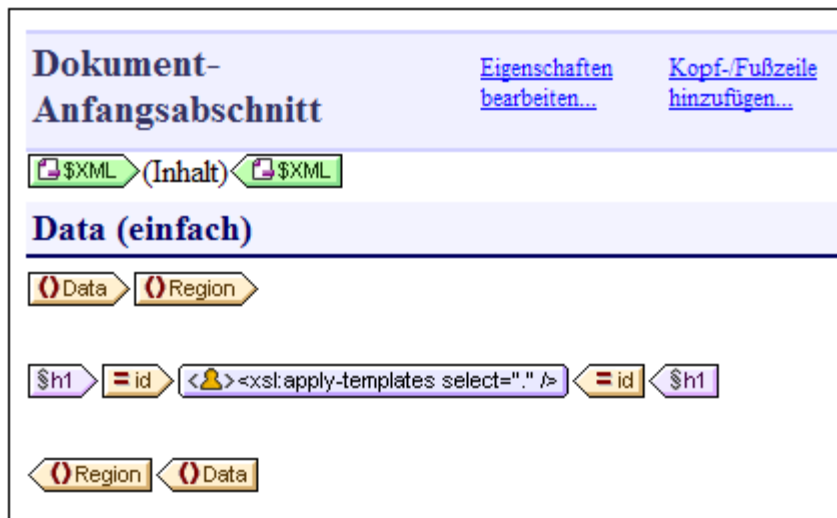
Unten sehen Sie die XML-Datei.

```
<?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>
```

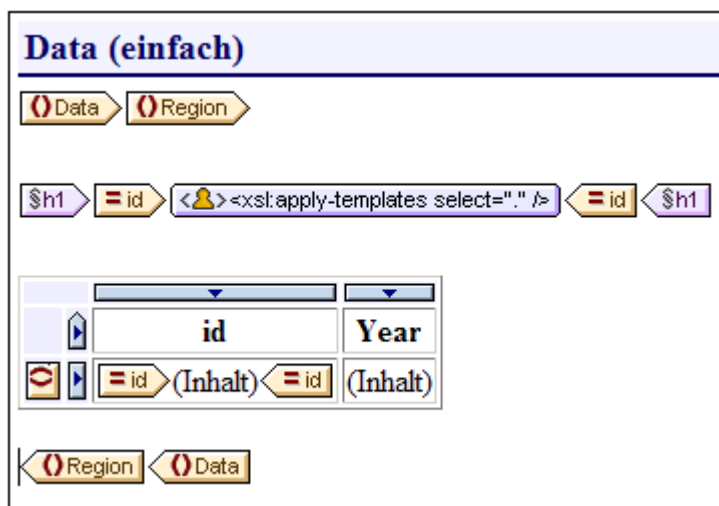
Führen Sie die oben aufgelisteten Schritte 1 bis 4 durch, um das SPS in der Design-Ansicht zu erhalten. Das SPS sollte ungefähr folgendermaßen aussehen:



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.



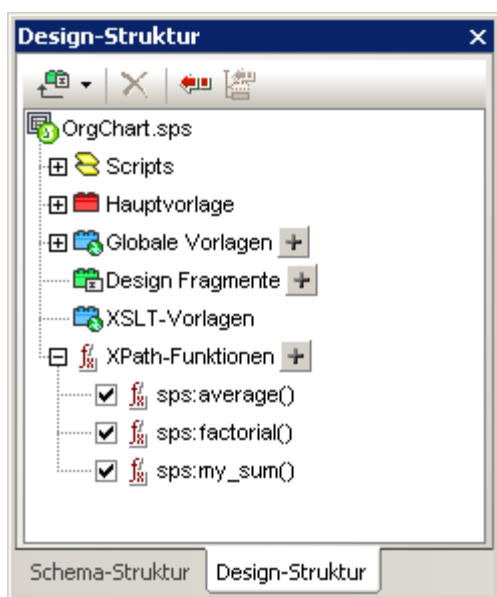
Sie können weitere Änderungen am Inhalt, der Struktur und den Darstellungseigenschaften des Dokuments machen, dann eine Vorschau der Ausgabe anzeigen und die Dateien mit dem Befehl [Generierte Dateien speichern](#) speichern.

12.6 Benutzerdefinierte XPath-Funktionen

Der SPS-Designer kann angepasste **XPath 2.0**-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.

XPath-Funktionen

Funktionsname:

Rückgabebetyp (optional):

Parameter

Name	Type (optional)	Instanzen

Funktionsrumpf (XPath-Ausdruck)

`$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]`

XPath ...

OK Abbrechen

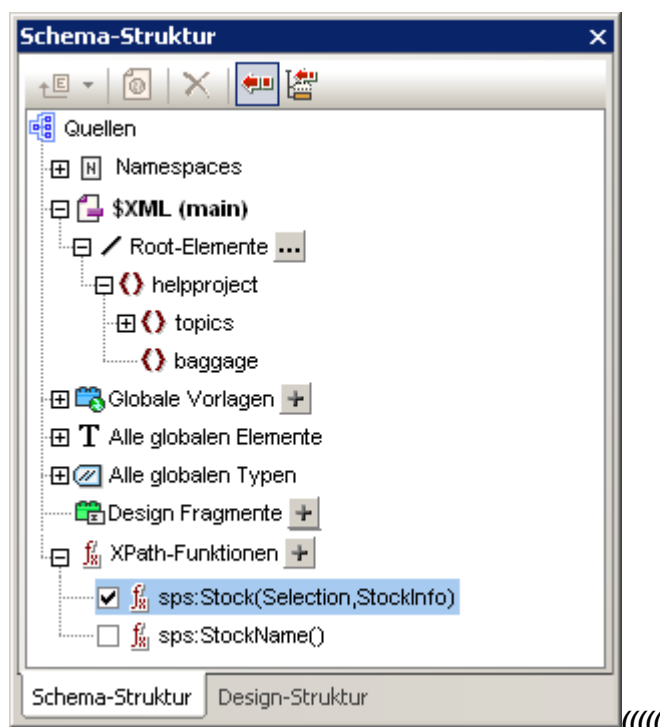
Nachdem eine benutzerdefinierte XPath-Funktion erstellt worden ist, kann sie an einer beliebigen Stelle des Designs verwendet werden.

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 XPath-Funktionen 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).



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 der oben definierte benutzerdefinierte XPath-Ausdruck `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.

Definieren einer XPath-Funktion

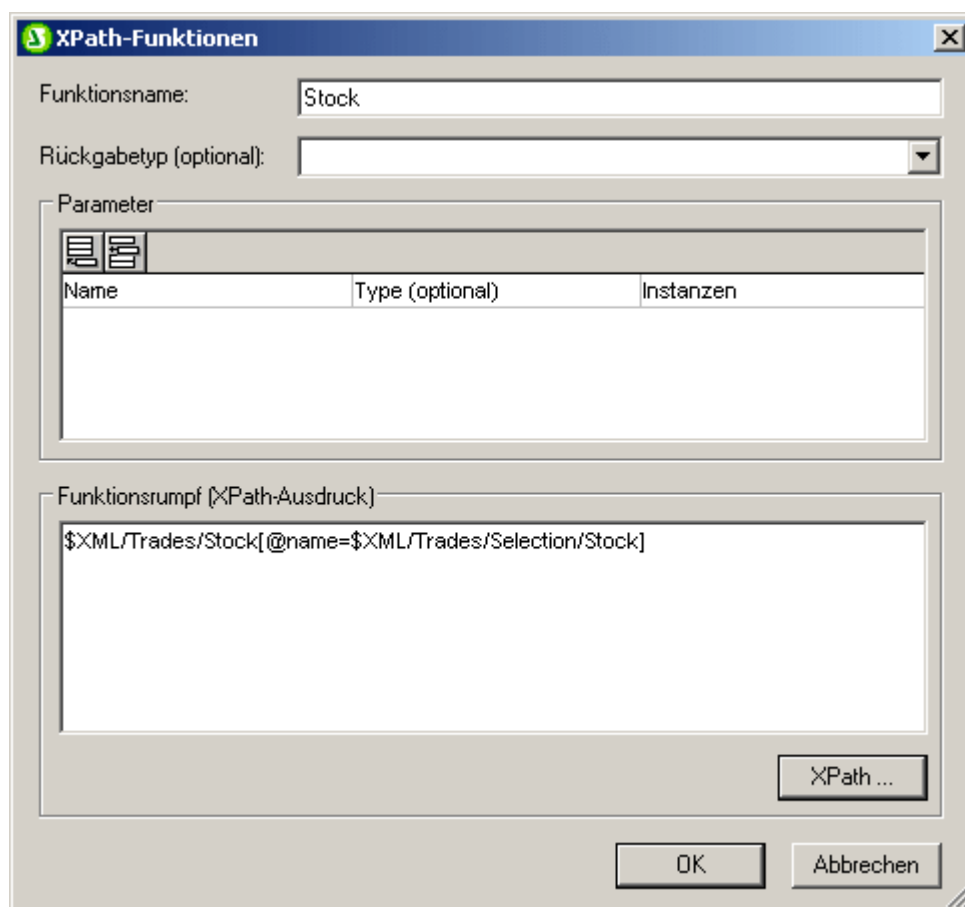
Eine benutzerdefinierte XPath-Funktion benötigt (i) einen Namen (einen Text-String) und (ii) eine Definition (einen XPath-Ausdruck).

Darüber hinaus können Sie für die Funktion einen oder mehrere **Parameter** definieren. Eine benutzerdefinierte XPath-Funktion kann auch einen optionalen **Rückgabotyp** haben (wird durch Auswählen eines Typs aus der Dropdown-Liste des Listenfelds *Rückgabotyp* angegeben). Ein Rückgabotyp 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. 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 Rückgabotyp 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.



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](#) und [Parameter und Nodes](#)). Ein Rückgabebetyp 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 mit 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](#).

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 Rückgabebetyp 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.

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.

The dialog box 'XPath-Funktionen' is shown. It has a title bar with a green icon and a close button. Inside, there are several fields and buttons:

- Funktionsname:** A text box containing 'Stock'.
- Rückgabebetyp (optional):** A dropdown menu that is currently empty.
- Parameter:** A section containing a table with three columns: 'Name', 'Type (optional)', and 'Instanzen'. The table is currently empty.
- Funktionsrumpf (XPath-Ausdruck):** A large text area containing the XPath expression: `$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]`.
- Buttons:** At the bottom right, there are three buttons: 'XPath ...', 'OK', and 'Abbrechen'.

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.

Parameter in XPath-Funktionen

Einer Funktion kann eine beliebige Anzahl von Parametern zugewiesen werden. Dies geschieht im Bereich "Parameter" des Dialogfelds "XPath-Funktionen" (siehe Abbildung unten).

Parameter, die definiert worden sind, können dann bei der Definition der Funktion (im Bereich "Funktionsrumpf") verwendet werden.

Beim Aufruf der Funktion erhalten die Parameter Parameterwerte. Sie sollten beachten, dass Parameterwerte überprüft werden bevor sie der Funktion übergeben werden. Fehler werden zurückgegeben wenn (i) ein Parameter definiert ist, aber in der Funktionsdefinition nicht verwendet wird, oder (ii) die Anzahl der Eingabeparameterwerte nicht der Anzahl der Parameter entspricht, oder (iii) der Typ eines beliebigen Parameterwerts nicht dem Typ entspricht, der für ihn in der Parameterdefinition angegeben wurde.

XPath-Funktionen

Funktionsname:

Rückgabetyt (optional):

Parameter

Name	Type (optional)	Instanzen
ContextStock	xs:string	Genau eines
Selection	xs:string	Genau eines
StockInfo	xs:string	Genau eines

Funktionsrumpf (XPath-Ausdruck)

`$ContextStock[@name=$Selection]$StockInfo`

XPath ...

OK Abbrechen

Reihenfolge von Parametern

Die Reihenfolge der Parameter ist wichtig, weil die Parameterwerte beim Funktionsaufruf an die Parameter in der Reihenfolge wie sie im Bereich "Parameter" (siehe oben) definiert sind, übergeben werden. Wenn also die XPath-Funktion `sps:Stock` XPath so wie in der Abbildung oben gezeigt definiert ist und wenn Sie mit dem XPath-Ausdruck `sps:Stock($XML, Node1, Node2)` aufgerufen wird, dann werden diese drei Werte – `$XML`, `Node1`, `Node2` – den Parametern – `$ContextStock`, `$Selection` und `$StockInfo` – zugewiesen. Beachten Sie, dass jeder Eingabeparameterwert vom nächsten durch ein Komma getrennt wird.

Die Reihenfolge der Parameter im Bereich "Parameter" kann mit den dort verfügbaren Symbolen **Anhängen**, **Einfügen** und **Löschen** gesteuert werden.

Datentyp von Parametern

Optional kann der erwartete Datentyp eines Parameters angegeben werden. Wenn ein Datentyp angegeben ist, wird der Datentyp des Eingabeparameterwerts gegen den angegebenen Datentyp geprüft und eine Fehlermeldung ausgegeben, wenn die Typen nicht übereinstimmen. Mit Hilfe dieser Funktion können die Eingabedaten geprüft werden.

Instanzen

Man kann jeden Parameter als einer Sequenz zugehörig betrachten. Die Eigenschaft *Instanz* jedes Parameters gibt an, wie viele Objekte in der Sequenz für diesen Parameter auftauchen dürfen. Da sowohl für die Trennung der Objekte in der Sequenz als auch für die Trennung der Parameter (bzw. Parameterwerte) voneinander Kommas verwendet werden, werden zur Abgrenzung der Parameter (bzw. Parameterwerte) falls nötig Klammern eingesetzt um den XPath-Ausdruck eindeutig zu machen. In diesem Zusammenhang sollten die folgenden Beispiele und Punkte beachtet werden:

- Einige XPath 2.0-Funktionen verwenden eine einfache Sequenz als Argument und diese einfache Sequenz muss in Klammern eingeschlossen sein. In der Funktion `avg` (`( count($a), $b, $c )`) verwendet die XPath 2.0 `avg()`-Funktion zum Beispiel die Sequenz `count($a), $b, $c` als ihr Argument, diese Sequenz muss in Klammern eingeschlossen sein. Das äußere Klammernpaar dient als Trennzeichen für die Funktion.
- Wenn Funktionen aufgerufen werden, werden mit dem Aufruf Parameterwerte übergeben. In den Eingabeparameterwerten müssen die Klammern korrekt gesetzt werden. Wenn, zum Beispiel, die benutzerdefinierte XPath-Funktion mit dem Namen `MyAverage()` mit dem XPath 2.0-Ausdruck `avg( count($a), $b, $c )` definiert wird, dann wären folgende Eingabeparameterwerte gültig: `MyAverage( ( 1,2,3 ), 4, 5 )`. Die Werte, die den drei Parametern `$a`, `$b`, und `$c` entsprechen, wären die Sequenz `( 1, 2, 3 )`, die Einzelsequenz `4` und die Einzelsequenz `5`. Einzelsequenzen können optional von Klammern umschlossen werden.
- Wenn für einen Parameter ein Datentyp angegeben wird, der mehr als eine Instanz (das heißt, mehr als ein Objekt in seiner Sequenz) haben kann, dann muss jedes Objekt in der Eingabesequenz vom angegebenen Datentyp sein.

Parameter und Sequenzen

Es ist wichtig die Beziehung zwischen Parametern und Sequenzen zu beachten. Diese muss in zwei verwandten Bereichen berücksichtigt werden:

- Die Beziehung zwischen Parametern und Sequenzen in der Definition der Funktion.
- Die Beziehung zwischen Eingabeparameterwerten und Sequenzen. Der Eingabeparameterwert wird beim Aufruf der Funktion übergeben.

Wenn die Funktionsdefinition ein XPath-Ausdruck ist, der eine Sequenz als Eingabe verwendet (*siehe Abbildung unten als Beispiel*), müssen Sie sorgfältig beim Verwenden der korrekten Syntax sein. Klammern spielen dabei eine wichtige Rolle, indem sie angeben, dass der Klammerinhalt zuerst geprüft werden soll. In Sequenzen, bei denen Kommas verwendet werden um Objekte voneinander zu trennen, dienen Klammern zum Trennen von Sequenzen. Ein Reihe von XPath 2.0-Funktionen verwendet eine Sequenz als ihr einfaches Argument und operiert anhand der Objekte der Sequenz. Hier finden Sie ein paar Beispiele:

- `avg( (10, 20, 30) )` Die XPath 2.0-Funktion `avg` verwendet eine Sequenz von Objekten als ihre Eingabe. Diese Sequenz, im Beispiel fett dargestellt, wird durch Klammern getrennt und ihre Objekte sind durch Kommas voneinander getrennt. (Das äußere Klammernpaar sind die Klammern der Funktion.)
- `avg( //Person/@salary )` Der Lokalisierungsschritt wählt ein Set von `salary`-Attribut-Nodes aus und gibt ihre Attributwerte als die Sequenz zurück, die geprüft (das heißt, deren Mittelwert ermittelt) werden soll, zurück. Es sind keine Klammern notwendig, weil die Sequenz nicht direkt eingegeben wird. Stattdessen werden die erforderlichen Werte aus dem XML-Dokument bezogen und der Funktion als Sequenz übergeben.
- `count( (10 to 34) )` Die Klammern geben an, dass der Bereich eine einfache Sequenz von Objekten übergibt.
- `count( (10 to 34, 37) )` Die Klammern geben an, dass die Sequenz aus den Objekten, die vom Bereich übergeben werden, sowie dem Objekt `37` besteht.
- `count( //Person )` Um den Lokalisierungsschritt, der die Sequenz der Werte des überprüften Nodesets zurückgibt, gibt es keine Klammern.

Wenn **Parameter** in einer Funktionsdefinition verwendet werden, die eine Sequenz als ihr Argument benötigt (*siehe Definition in der Abbildung unten*), dann müssen diese Parameter innerhalb von Klammern platziert werden um die Abgrenzung der Sequenz anzugeben. Jedes Objekt in der Sequenz ist vom folgenden Objekt durch ein Komma getrennt.

XPath-Funktionen

Funktionsname:

Rückgabebetyp (optional):

Parameter

Name	Type (optional)	Instanzen
a	xs:integer	Genau eines
b	xs:integer	Genau eines
c	xs:integer	Genau eines

Funktionsrumpf (XPath-Ausdruck)

`avg( $a, $b, $c )`

XPath ...

OK Abbrechen

Jeder Parameter kann als Einzelsequenz betrachtet werden. Die Anzahl von Objekten, die innerhalb der Sequenz zulässig ist, 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. Jeder Parameterwert muss daher als eine Sequenz von einem Objekt angesehen werden. Diese Einzelsequenzen stellen wiederum die Objekte dar, die die Sequenz darstellen, die das Einzelargument der Funktion ist. Wenn einer der Eingabeparameterwerte keine Einzelsequenz ist, wird eine Fehlermeldung zurückgegeben.

Im Folgenden finden Sie Beispiele für die Verwendung von Parametern in **Aufrufen** der XPath-Funktion `ThreeAverage()`, die in der obigen Abbildung zu sehen ist.

- `sps:ThreeAverage( 10, 20, 30 )` gibt 20 zurück. Es gibt drei gültige Eingabeparameterwerte, die den drei Parametern entsprechen.
- `sps:ThreeAverage( ( 10 ), ( 20 ), ( 30 ) )` Es gibt drei gültige Eingabeparameterwerte, die den drei Parametern entsprechen. Jeder Eingabeparameterwert ist von Klammern umschlossen (was redundant aber kein Fehler ist, da jede Sequenz eine Einzelsequenz ist).
- `sps:ThreeAverage( ( 10 ), 20, 30 )` gibt 20 zurück. Es gibt drei gültige Eingabeparameterwerte, die den drei Parametern entsprechen. Der erste Eingabeparameterwert ist von Klammern umschlossen (redundant aber kein Fehler).
- `sps:ThreeAverage( ( 10, 20 ), ( 30 ), ( 40 ) )` gibt einen Fehler zurück weil der erste Eingabeparameterwert 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 Eingabeparameterwert übergeben wird. Darüber hinaus ist der

Eingabeparameterwert 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 von einem oder mehreren Objekten definiert.

XPath-Funktionen

Funktionsname:

Rückgabebetyp (optional):

Name	Type (optional)	Instanzen
a	xs:integer	Mindestens eines
b	xs:integer	Genau eines
c	xs:integer	Genau eines

Funktionsrumpf (XPath-Ausdruck):

```
avg( (count($a ), $b, $c) )
```

Buttons: XPath ... OK Abbrechen

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 wird, ermittelt wird, das Resultat zu den beiden Integer-Werten, die von den beiden Parametern übergeben werden, hinzugefügt und dann das Resultat durch drei geteilt wird, um den Durchschnitt zu erhalten. Beachten Sie folgendes:

- Der erste Parameter ist nicht von den Klammern umschlossen, die als Trennzeichen für die Sequenz dienen. (Die Klammern um den ersten Parameter sind jede der `count()` - Funktion). Das liegt daran, dass der Eingabeparameterwert die erforderliche Einzelsequenz, nämlich das Einzelargument der `count()`-Funktion, übergeben wird.
- Die Sequenz, die das Argument der `avg()` -Funktion ist, ist von Klammern umschlossen. Damit wird angegeben, dass das Argument der `avg()` -Funktion eine Einzelsequenz ist, die aus drei Objekten, nämlich den drei Integer-Werte sind, die von der `count()` -Funktion übergeben werden und den beiden Parametern `b` und `c`, besteht.

Hier finden Sie Beispiele zur Verwendung von Parametern in **Aufrufen** der XPath-Funktion `Average()`, die in der obigen Abbildung gezeigt wird.

- `sps: Average( ( 1, 2 ), 3, 4 )` gibt 3 zurück. Es gibt drei gültige Eingabeparameterwerte, die den drei Parametern entsprechen. Der erste Parameter ist in Klammern eingeschlossen, um ihn als den ersten Parameterwert abzugrenzen. 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 Eingabeparameterwerte. Der erste Parameter darf eine Sequenz von einem Objekt haben. Es werden keine Klammern benötigt um separate Eingabeparameterwerte anzugeben.
- `sps: Average( ( 1 to 10 ), ( 20 ), 30 )` gibt 20 zurück. Es gibt drei gültige Eingabeparameterwerte, die den drei Parametern entsprechen. Klammern können verwendet werden, um separate Eingabeparameterwerte anzugeben.
- `sps: Average( ( 10, 20 ), ( 30 ), ( 40, 50 ) )` gibt einen Fehler zurück weil der dritte Parametereingabewert ungültig ist. Es handelt sich um eine Sequenz von mehreren Objekten, wohingegen der dritte Parameter als Einzelsequenz definiert wurde.
- `sps: Average( ( 10, 20, 30 ), 40, ( ) )` gibt einen Fehler zurück weil der dritte Eingabeparameterwert eine leere Sequenz ist. Es muss sich um eine Sequenz mit einem einzelnen Integer-Wert handeln.

Weitere interessante Punkte

Die folgenden zusätzlichen Punkte sollten beachtet werden:

- Wenn ein Parameter mit einer `Mindestens eines`-Instanz definiert ist, könnte eine Funktion wie `MyAverage()` mit einem XPath-Ausdruck wie `avg( ( $a ) )` definiert werden. Diese Funktion würde eine Einzelsequenz von einem oder mehreren Objekten als Argument akzeptieren. Die Funktion könnte wie folgt aufgerufen werden: `sps: MyAverage( ( 2, 3, 4 ) )`, und würde den Wert 3 zurückgeben. Der Eingabeparameterwert muss in Klammern eingeschlossen werden, um sicherzustellen, dass die Eingabe als einzelne Sequenz und nicht als drei Einzelsequenzen gelesen wird (was der Fall sein würde, wenn die Objekte durch Kommas getrennt und nicht von Klammern umschlossen wären).
- Wenn ein Parameter `$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 als erster Eingabeparameterwert 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 nicht als Eingabewert zählt). Folgendes würde jedoch einen Fehler generieren: `sps: MyAverage( 20, 10 )`, weil der erste Eingabeparameterwert nicht zurückgegeben 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 ihrem 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.

Parameter und Nodes

Wenn Parameter in XPath-Funktionen, die Nodes lokalisieren, verwendet werden, ist es wichtig zu beachten, dass die Funktion keinen Kontext-Node hat, ganz gleich wo sie sich im Design befindet. 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 Parameterwerte übergeben werden.

Ziehen Sie die benutzerdefinierte XPath-Funktion `stock()` in Betracht, die mit drei Parametern definiert ist, wie in der Abbildung unten zu sehen ist.

XPath-Funktionen

Funktionsname:

Rückgabotyp (optional):

Parameter

Name	Type (optional)	Instanzen
ContextStock	xs:string	Genau eines
Selection	xs:string	Genau eines
StockInfo	xs:string	Genau eines

Funktionsrumpf (XPath-Ausdruck)

`$ContextStock[@name=$Selection]$StockInfo`

XPath ...

OK Abbrechen

Die Definition im Funktionsrumpf ist `$ContextNode[@name=$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 )
```

Die Argumente der Funktion sind ihre drei Parameterwerte, von denen jeder entweder Information zum Kontext-Node oder Information zur Node-Lokalisierung bietet. Alternativ dazu kann der folgende XPath-Ausdruck als Funktionsaufruf verwendet werden:

```
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. Wenn eine Funktion von einem XPath-Ausdruck in einer Design-Komponente aufgerufen wird, dann wird jeder Parameterwert im Funktionsaufruf überprüft und das Ergebnis der Überprüfung wird der Funktion als Parameterwert übergeben.

Beachten Sie, dass es im obigen Funktionsaufruf drei Eingabeparameterwerte gibt, die jeweils den drei Parametern entsprechen:

- `$ContextNode = $XML/Trades/Stock` (das Element `/Trades/Stock`)
- `$Selection = $XML/Trades/Selection/Stock` (das Element `/Trades/Selection/Stock`)
- `$StockInfo = @name`

Der XPath-Ausdruck in der Funktionsdefinition lautet:

```
$ContextNode[@name=$Selection]/$StockInfo
```

Wenn die Eingabeparameterwerte substituiert 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 Lokalisierungsschritte werden auf diese Weise über Parameter an die Funktion übergeben. Funktionen können dann die benötigten Nodes auffinden.

12.7 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-Funktionen zur Bearbeitung von Datum und Uhrzeit zur Verfügung (siehe [Beispiele unten](#)). StyleVision bietet zur Unterstützung der Datentypen `xs:date` oder `xs:dateTime` folgende Funktionen:

1. Eine grafische [Datumswahl](#), über die der Benutzer der Authentic-Ansicht Daten im korrekten lexikalischen Format des Datentyps für diesen Node eingeben kann.
2. Eine breite Palette an Optionen zur [Datumsformatierung](#) über die Funktion [Werteformatierung](#).

In den Unterabschnitten dieses Abschnitts mit den Titeln [Verwendung der Datumswahl](#) und [Datumsformatierung](#) sind diese StyleVision Funktionen näher beschrieben. Im restlichen Teil des Abschnitts erfahren Sie, wie Sie mit Hilfe von XPath 2.0 Berechnungen mit Datumswerten durchführen können.

Anmerkung: Datums- und Uhrzeitwerte können mit XPath 1.0 nicht bearbeitet werden. Sie können jedoch auch mit XPath 1.0 die [Datumswahl](#) verwenden, um sicher zu stellen, dass die Daten korrekt eingegeben werden und die Datumswerte mit Hilfe der Werteformatierung [formatieren](#).

Datumsberechnungen mit XPath 2.0

Datumswerte können in [automatischen Berechnungen](#) 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 365 *
        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 Zeitzonekomponente (durch das Präfix `±`) optional ist. Mit Hilfe der Datumswahl stellen Sie sicher, dass das Datum in lexikalisch korrekter Form eingegeben wird.

Verwendung der Datumswahl

Die Datumswahl (*Abbildung unten*) ist ein grafischer Kalender in der Authentic-Ansicht zur Eingabe des Datums im korrekten lexikalischen Format für Nodes vom Datentyp `xs:date` und `xs:dateTime`.



Das lexikalische Format wird je nach Datentyp entsprechend eingegeben.

- Bei `xs:date` ist das Format `JJJJ-MM-TT[ ±HH:MM]`, wobei die Zeitonenkomponente (Präfix $\pm$) nach der XML-Schemaspezifikation optional ist. Ein Wert für die Zeitonenkomponente kann in der Datumswahl gewählt werden.
- Bei `xs:dateTime` ist das Eingabeformat `JJJJ-MM-TTTHH:MM:SS[ ±HH:MM]`. Die Zeitonenkomponente (Präfix $\pm$) ist gemäß der XML-Schemaspezifikation optional. Ein Wert für die Zeitonenkomponente kann vom Benutzer gewählt werden.

Einfügen einer Datumswahl in das Design und Löschen der Datumswahl

Sie können (i) für jeden Node vom Datentyp `xs:date` oder `xs:dateTime` und (ii) wenn der Node entweder als Inhalt oder als Eingabefeld erstellt wird, eine Datumswahl in das SPS Design einfügen. Eine Datumswahl kann auf eine von zwei Arten eingefügt werden:

- Standardmäßig, wenn im SPS ein Node vom Datentyp `xs:date` oder `xs:dateTime` erstellt wird. Um diese Standardeinstellung zu aktivieren, schalten Sie die Funktion "Datumswahl automatisch hinzufügen" ein. Aktivieren Sie dazu den Befehl **Authentic | Datumswahl automatisch hinzufügen**. Oder klicken Sie alternativ dazu auf das  Symbol in der Symbolleiste. Wenn diese Funktion aktiviert ist, wird die Datumswahl hinzugefügt, wenn ein Element vom Datentyp `xs:date` oder `xs:dateTime` entweder als Inhalt oder als Eingabefeld erstellt wird oder in eine dieser beiden Komponente umgewandelt wird.
- Durch Auswahl des Kontextmenübefehls **Einfügen | Datumswahl**, wenn der Cursor sich im SPS im Node `xs:date` oder `xs:dateTime` an der gewünschten Stelle befindet.

Wenn die Datumswahl eingefügt wird, erscheint an dieser Stelle das Datumswahlsymbol . Verwenden Sie zum Löschen der Datumswahl die **Entf**- oder **Rücktaste**.

Verwendung der Datumswahl in der Authentic-Ansicht

In der Authentic-Ansicht wird die Datumswahl als Symbol angezeigt (*Abbildung unten*).

Organization Chart

Location of logo:

Last Updated: 2003-09-01 

Klicken Sie zum Ändern des Datums auf das Symbol. Daraufhin wird die Datumswahl angezeigt (*Abbildung unten*). Wählen Sie zur Eingabe eines neuen Datums das gewünschte Datum in der Datumswahl aus. Das Datum wird je nach dem Datentyp des Node im korrekten lexikalischen Format eingegeben.

Location of logo:

Last Updated: 2003-09-01

◀ September ▶ 2003 ▶

M	T	W	T	F	S	S	
25	26	27	28	29	30	31	35
1	2	3	4	5	6	7	36
8	9	10	11	12	13	14	37
15	16	17	18	19	20	21	38
22	23	24	25	26	27	28	39
29	30	1	2	3	4	5	40

Today No Timezone

Nanonull, Inc.

Location:

Um eine Zeitzone einzugeben, klicken Sie auf die Schaltfläche "Zeitzone", die standardmäßig auf "Keine Zeitzone" gestellt ist. Die Zeitzone wird in dem lexikalischen Format eingegeben, das für den Datentyp dieses Node erforderlich ist (*Abbildung unten*).

Location of logo:

Last Updated: 2003-09-01+01:00

◀ September ▶ 2003 ▶

M	T	W	T	F	S	S	
25	26	27	28	29	30	31	35
1	2	3	4	5	6	7	36
8	9	10	11	12	13	14	37
15	16	17	18	19	20	21	38
22	23	24	25	26	27	28	39
29	30	1	2	3	4	5	40

Today +01 : 00

Nanonull, Inc.

Location:

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:

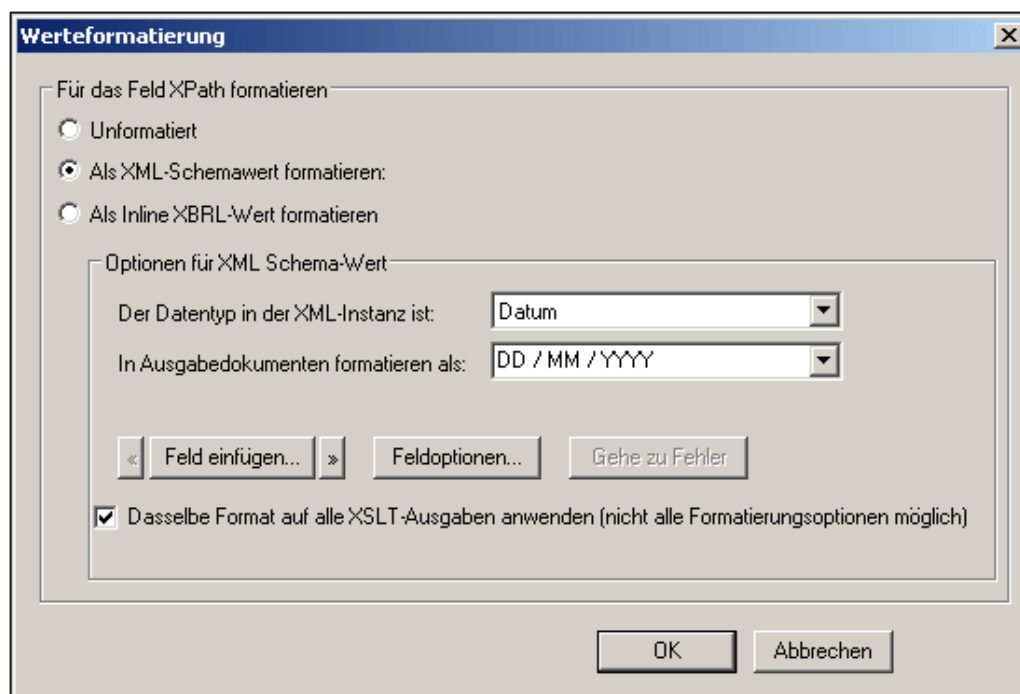
- `xs:date`
- `xs:dateTime`
- `xs:duration`
 - `xs:gYear`
 - `xs:gYearMonth`
 - `xs:gMonth`
 - `xs:gMonthDay`
 - `xs:gDay`

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 `Inhalt` aus und anschließend die Eigenschaftsgruppe `Inhalt`. Klicken Sie nun auf die Bearbeiten

Schaltfläche  der Eigenschaft `Werteformatierung`. Daraufhin erscheint das Dialogfeld "Werteformatierung" (*Abbildung unten*).



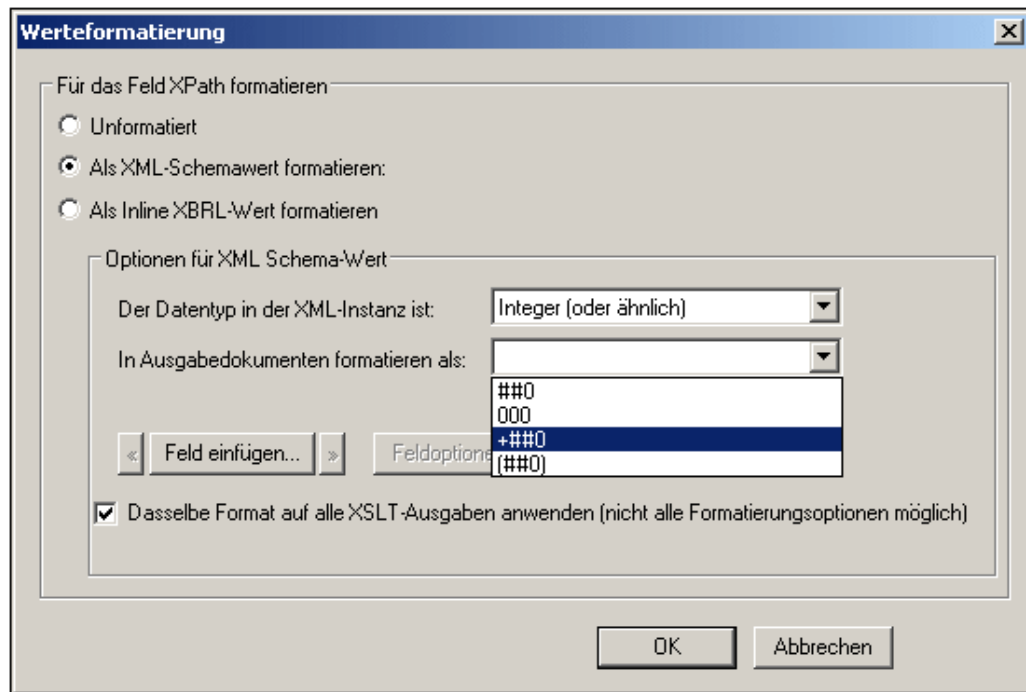
Standardmäßig ist das Optionsfeld "Unformatiert" (das lexikalische Standardformat für den Datentyp des Node) ausgewählt.

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](#).

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](#).

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- und RTF-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.

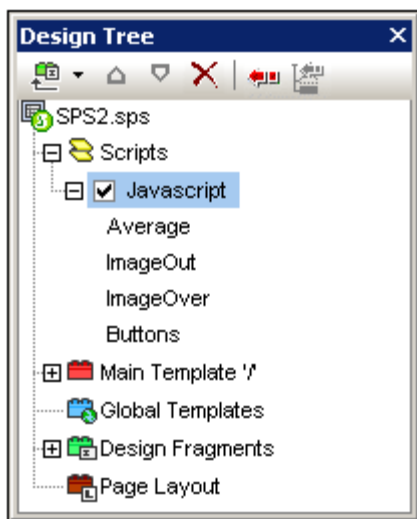
12.8 Verwenden von Scripts

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:

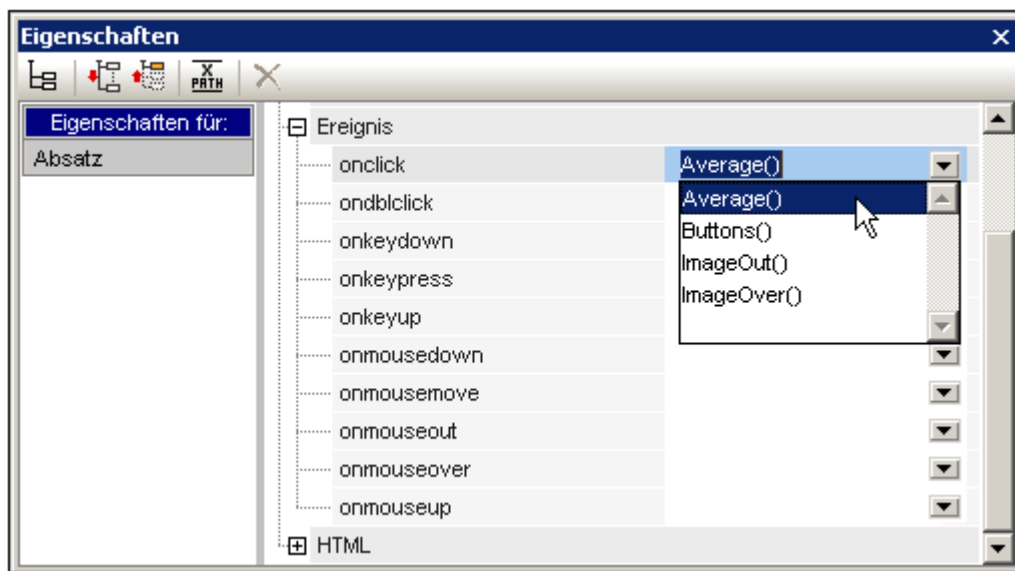
- Um mit Hilfe mehrerer Script-Anweisungen ein komplexes Ergebnis zu erhalten. In diesem Fall sollten alle erforderlichen Scripts an einer Stelle (im Header) als separate Funktionen erstellt und anschließend im Design-Dokument referenziert werden.
- Wenn Sie ein bestimmtes Script an mehreren Stellen im Design-Dokument verwenden möchten.

Die Definition von Funktionen im JavaScript Editor wird im Unterabschnitt [Definition von JavaScript-Funktionen](#) 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 "Scripts" 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" (*Abbildung unten*) zugewiesen werden. Informationen zum Zuweisen einer JavaScript-Funktion zu einem Event Handler finden Sie im Abschnitt [Zuweisen von Funktionen zu Event Handlern](#).

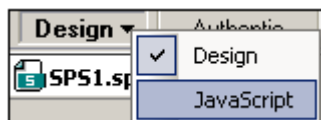


Anmerkungen: Scripts können nur in der HTML-Ausgabe, nicht aber in der Authentic-Ansicht angewendet werden.

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*).



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)
}

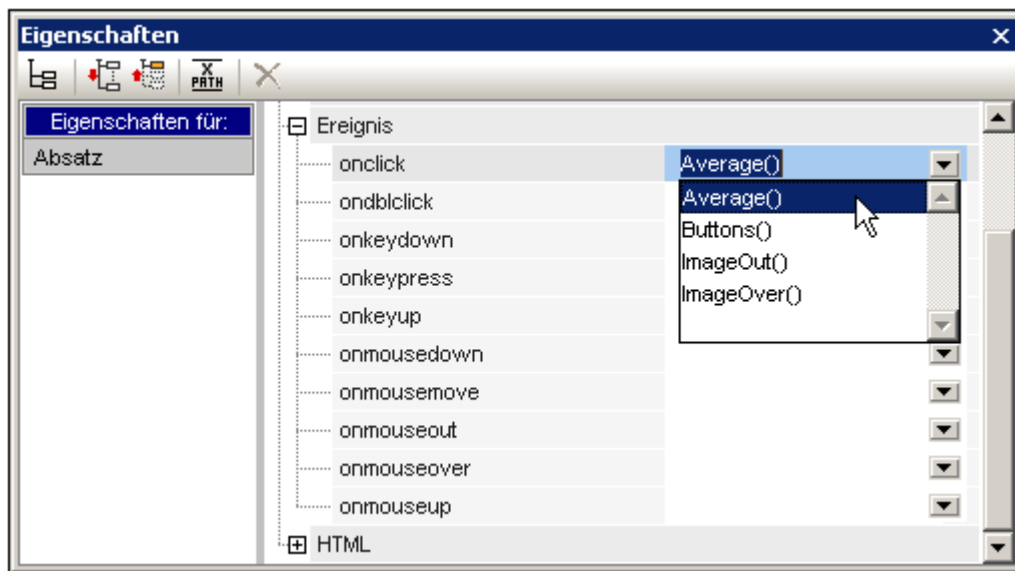
function ClearStatus()
{
    window.status="";
}
-->

</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](#).

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.

Externe JavaScript-Dateien

Ein SPS kann externe JavaScript-Dateien auf zwei Arten aufrufen:

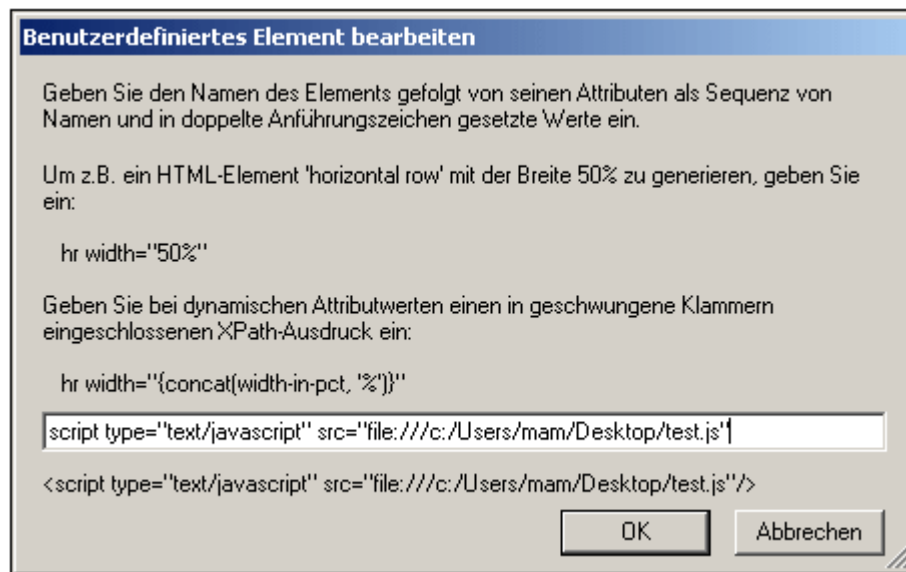
1. [Durch Erstellung eines benutzerdefinierten Elements oder XML-Blocks](#). Diese Design-Objekte können ein `SCRIPT` Element enthalten, das die externe JavaScript-Datei aufruft. Beachten Sie, dass sich das benutzerdefinierte Element bzw. der benutzerdefinierte XML-Block innerhalb des `BODY` Elements 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](#), 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](#) und [benutzerdefinierte XML-Blöcke](#) 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](#) 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](#).



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**.

Sie können dasselbe Ergebnis auch mit einem [benutzerdefinierten XML-Block](#) erzielen. Verwenden Sie dazu dasselbe oben beschriebene Verfahren wie bei benutzerdefinierten Elementen. Der einzige Unterschied ist, (i) dass anstelle eines [benutzerdefinierten Elements](#) ein [benutzerdefinierter XML-Block](#) 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](#) 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.

12.9 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:

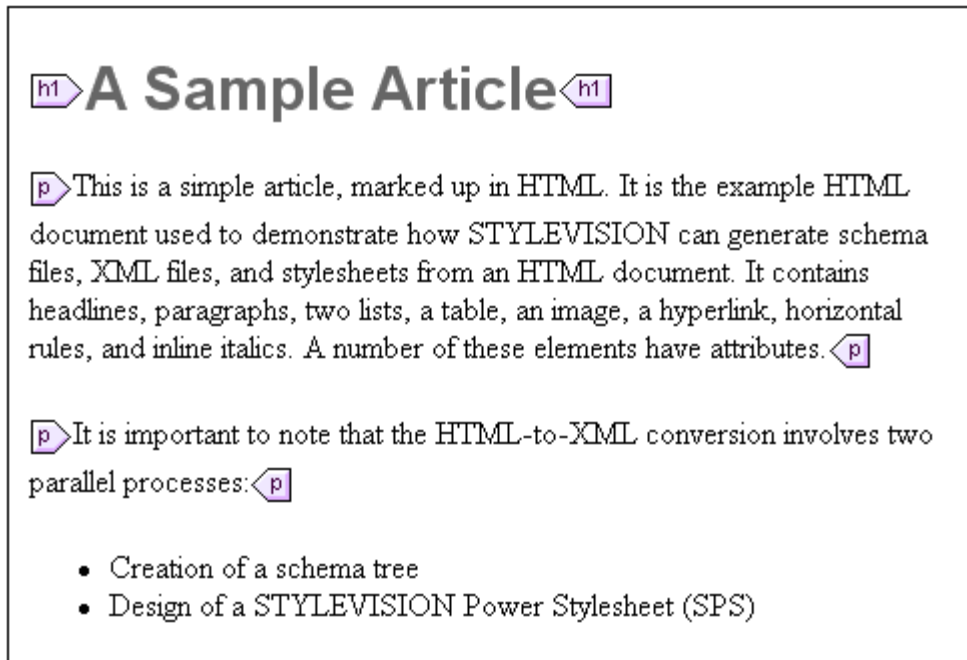
1. [Erstellen eines neuen SPS mittels HTML-Import](#). Wenn eine HTML-Datei in StyleVision importiert wird, wird ein neues SPS-Dokument erstellt. Die HTML-Datei wird in der Design-Dokumentansicht mit Markup-Tags angezeigt. Gleichzeitig wird im Schemafenster eine Schema-Struktur mit einem einzigen Element - einem Dokumentelement mit dem Namen `Root` - 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](#). 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](#):
 - 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.

Erstellen eines neuen SPS anhand einer HTML-Datei

Um anhand einer HTML-Datei ein neues SPS zu erstellen, gehen Sie folgendermaßen vor:

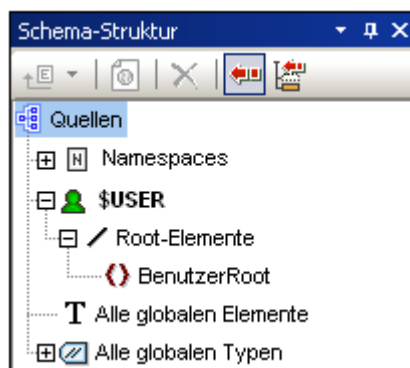
1. Wählen Sie den Menübefehl **Datei | Neu | Neu von HTML-Datei**.
2. Suchen Sie im Dialogfeld "Öffnen" die gewünschte HTML-Datei. Wählen Sie sie aus und klicken Sie auf **Öffnen**.
3. Sie werden gefragt, ob relative Pfade in absolute Pfade konvertiert werden sollen. Wählen Sie die gewünschte Option und klicken Sie auf **OK**.

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.

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](#) 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 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

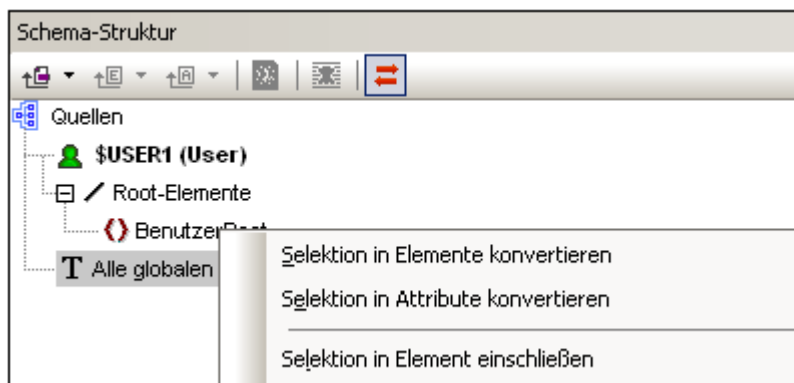
Wenn Sie die Auswahl über den gewünschten Node der Schemastruktur ziehen, erscheint eines der folgenden Symbole sowie die Popup-Meldung: `Create new schema item`.

- 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. In der Schemastruktur (in der Seitenleiste "Schema-Struktur") wird ein solcher XML Node durch Klammern, die drei Punkte enthalten, gekennzeichnet.
- 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](#).

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 Symbolleistenflächen der Seitenleiste "Schema-Struktur" verwenden.

Um einen Node zu löschen, wählen Sie ihn aus und verwenden Sie anschließend entweder das Kontextmenü oder die Symbolleistenfläche. 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 Stilfeunktionen von StyleVision geändert werden.

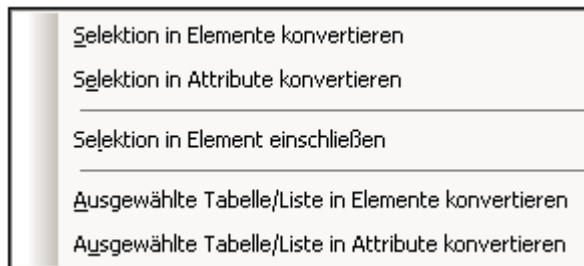
Erstellen von Tabellen und Listen als Elemente/Attribute

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.

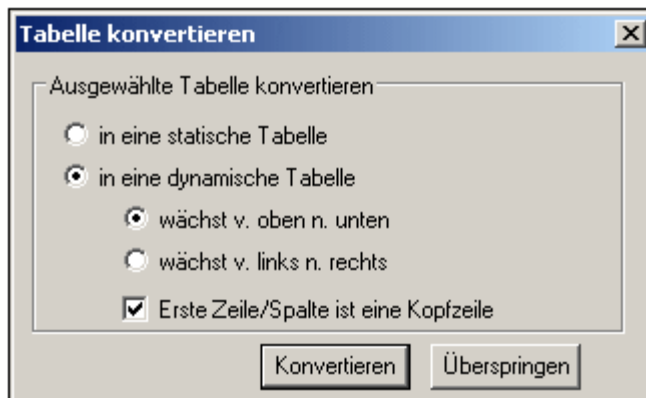
Konvertieren einer Tabelle in Elemente/Attribute

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.



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

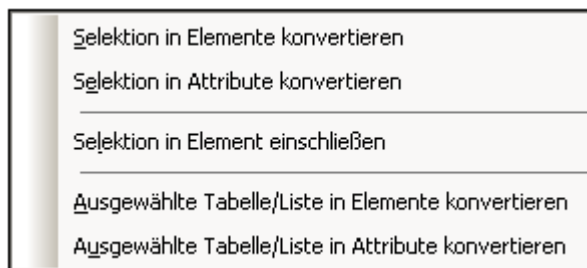
Kopfzeile in das Design eingefügt, die die Spaltenüberschriften als statischen Text enthält; und (ii) die Schemaelement/Attribut-Nodes erhalten den Text der Kopfzeile als Namen. Wenn die erste Zeile/Spalte nicht als Kopfzeile definiert wird, so wird keine Kopfzeile in das Design eingefügt.

6. Klicken Sie nach Auswahl der gewünschten Option(en) zum Fertigstellen auf **Konvertieren**.

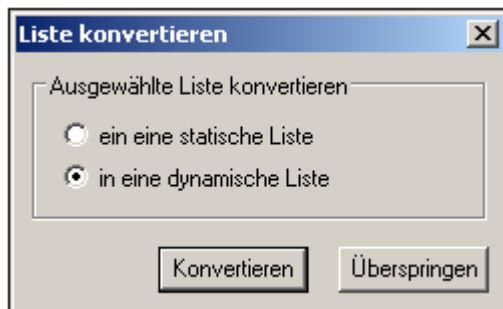
Konvertieren einer Liste in Elemente/Attribute

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 Listeneintragsselement mit dem Platzhalter (*Inhalt*) eingefügt.

4. Klicken Sie nach Auswahl der gewünschten Option(en) zum Fertigstellen auf **Konvertieren**.

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.

12.10 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:

- 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)
- Generieren Sie mit diesem SPS auch die XSLT-für-HTML-Datei.
- Transformieren Sie die XML-Datei mit Hilfe der XSLT-für-HTML-Datei.
- 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 AltovaXML](#) installiert sein.

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 AltovaXML. (AltovaXML ist der XSLT-Transformationsprozessor von Altova. AltovaXML kann von der Altova Website heruntergeladen werden.) AltovaXML 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.

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 AltovaXML

Stellen Sie sicher, dass die neueste Version von AltovaXML installiert ist. AltovaXML ist der Transformationsprozessor von Altova, der zum Transformieren der anhand der Datenbank generierten XML-Datei verwendet wird.

Aktivieren Sie Internet Information Server (Webserver von Microsoft)

Wenn Internet Information Server (IIS) nicht aktiviert ist, 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.
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. Stellen Sie sicher, dass sich die Datenbank und die SPS-Datei im selben Ordner befinden.
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.

Generieren Sie in StyleVision die XSLT-für-HTML-Datei

Generieren Sie die XSLT-für-HTML-Datei durch Aufruf des Befehls *Dateien | Generierte Dateien speichern | Generierte XSLT-HTML-Datei speichern*. Speichern Sie die Datei im selben Ordner, in dem sich auch die ASPX-Applikation befindet.

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* 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).

Führen Sie die Applikation aus

Gehen Sie im Browser zu `localhost/Test1/Test.aspx` (unter der Voraussetzung, dass `Test1` der Name des Ordners ist, in dem die ASPX-Applikation gespeichert wurde).

Die transformierte HTML-Datei sollte nun zu sehen sein. Wenn der Browser hängen bleibt, überprüfen Sie, ob die AltovaXML-Lizenzdatei vorhanden ist. Wenn Sie diese Seite aktualisieren, werden die neuesten Daten aus der Datenbank angezeigt.

12.11 PXF-Datei: Behälter für SPS und dazugehörige Dateien

Ein SPS-Design, in dem XSLT 2.0 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](#)
- [Bereitstellen einer PXF-Datei](#)

Anmerkungen

Beachten Sie bitte die folgenden Punkte zur PXF-Funktion:

- Diese Funktion wird nur bei SPS-Dateien unterstützt, die unter Verwendung von XSLT 2.0 erstellt wurden.
- XBRL und XML-Schemas für Datenbankspalten werden als Quelldateien nicht unterstützt. Das Dialogfeld, in dem Sie gefragt werden, ob die Datei als SPS oder PXF-Datei gespeichert werden soll, wird nicht angezeigt, wenn eine SPS-Datei eine solche Schemaquelle enthält. Wenn Sie eine PXF-Datei bearbeiten, kann eine solche Schemaquelle nicht hinzugefügt werden.

Erstellen einer PXF-Datei

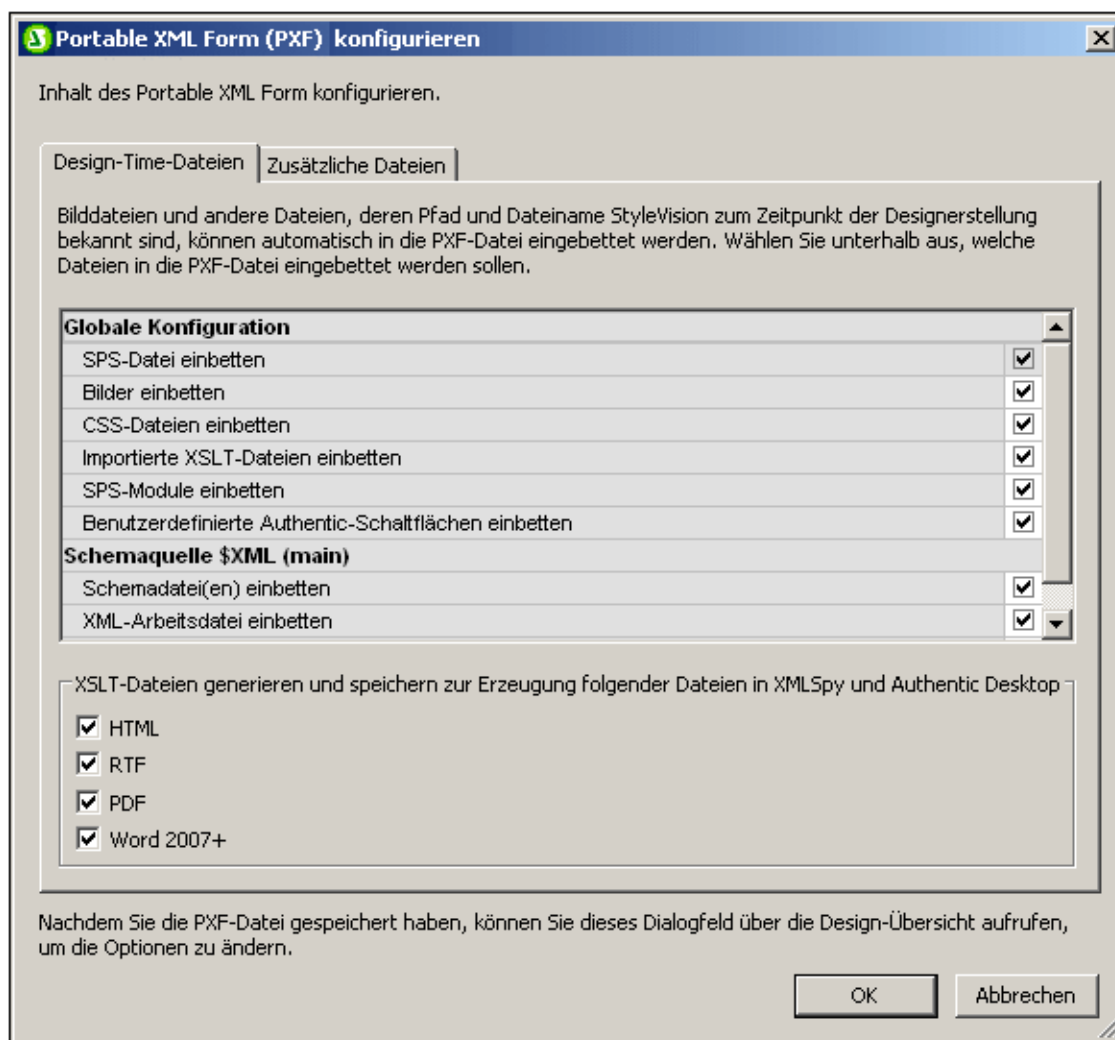
Um eine PXF-Datei zu erstellen, die ein SPS-Design sowie die dazugehörigen Dateien enthält, öffnen Sie das SPS-Design in StyleVision und wählen Sie den Befehl **Datei | Speichern unter**. Daraufhin wird das Dialogfeld "Design speichern" (*Abbildung unten*) geöffnet.



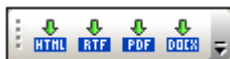
Das **SPS-Format** ist das Altova Standardformat für StyleVision Designs. In diesem Abschnitt wird das PXF-Format beschrieben, daher wird das SPS-Format hier nicht näher behandelt. Das Speichern einer Datei als SPS wird im Abschnitt [Benutzerreferenz](#) im Detail beschrieben.

Als PXF-Datei speichern

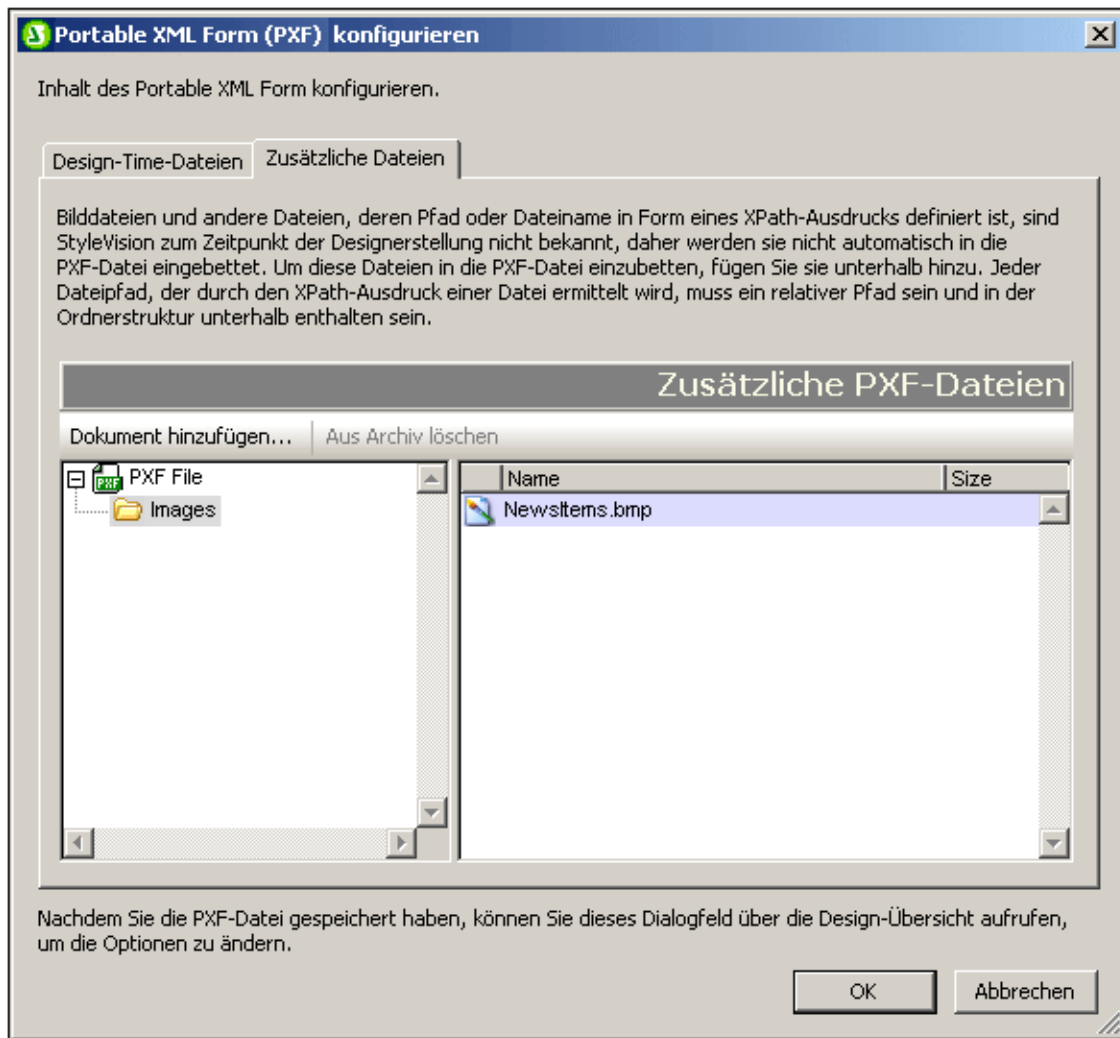
Bei Auswahl der PXF-Option wird das bekannte Windows-Dialogfeld "Speichern unter" angezeigt. Dieser Speicherbefehl funktioniert genau so, wie der [Befehl "Speichern unter"](#), mit dem zusätzlichen Schritt, dass die Dateien, die in die PFX-Datei inkludiert werden sollen, ausgewählt werden müssen. Nachdem 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. 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/NewsItem.bmp` referenziert werden.

Bearbeiten einer PXF-Datei

Eine PXF-Datei kann in StyleVision mit dem Befehl [Datei | Öffnen](#) 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*).

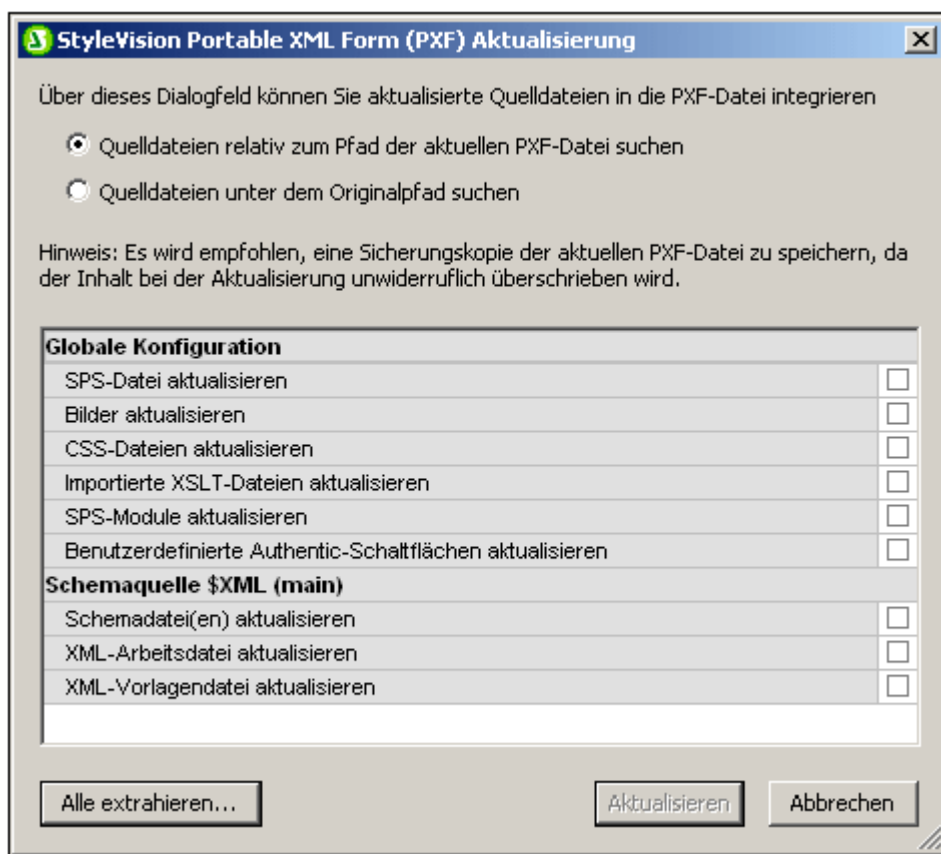


Konfigurieren der eingebetteten Dateien

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](#) 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*).



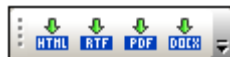
Wählen Sie zuerst aus, ob die Quelldateien relativ zu ihren aktuellen PXF-Ordern oder vom Originalpfad aus aufgerufen werden sollen. Klicken Sie anschließend auf die Dateien, die aktualisiert werden sollen und klicken Sie auf **Aktualisieren**. Daraufhin wird eine neue PXF-Datei erstellt, die die bestehende PXF-Datei überschreibt. Es empfiehlt sich daher vor der Aktualisierung der PXF-Datei eine Sicherungskopie davon anzulegen.

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.

Eine PXF-Datei kann in der [Authentic](#) 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-Ausgabe 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.

Kapitel 13

SPS-Dateien und Datenbanken

13 SPS-Dateien und Datenbanken

Altova Website:  [Datenbankberichte](#)

Wenn als Basis eines SPS - also als Hauptschema eines SPS - eine Datenbank verwendet wird, kann das SPS auf folgende Arten verwendet werden:

- um die Datenbank in der [Authentic-Ansicht](#) zu bearbeiten.
- um ein XML-Schema zu generieren, dessen Struktur auf einer Datenbank basiert (wenn die DB kein Schema enthält; nur XML-Datenbanken wie z.B. IBM DB2 Version 9 und höher enthalten Schemas)
- um eine XML-Datei mit Daten aus der Datenbank zu generieren (wenn die erforderlichen XML-Daten noch nicht im XML-Format gespeichert sind).
- um XSLT Stylesheet für die HTML- und RTF-Ausgabe zu erstellen und zu generieren
- um (auf Basis des SPS Designs) Datenbankberichte in den Formaten HTML und RTF zu generieren. In StyleVision können Sie eine Vorschau der Berichte anzeigen.

Wenn die Quelle eines Zusatzschemas in einem SPS eine Datenbank ist, können die Daten aus der Datenbank in das Design-Dokument inkludiert werden, doch kann die Datenbank selbst nicht in der [Authentic-Ansicht](#) bearbeitet werden. Nur das XML-Dokument oder die mit dem Hauptschema verknüpfte Datenbank kann bearbeitet werden.

Allgemeines Verfahren

In diesem Abschnitt wird die Verfahrensweise beim Arbeiten mit Datenbanken in StyleVision beschrieben. Nach einem kurzen [Einführungskapitel](#) mit einer [Übersicht über die Verwendung von Datenbanken in StyleVision](#) werden in den weiteren Unterabschnitten dieses Abschnitts die verschiedenen Schritte des Arbeitsablaufs beschrieben. Wir unterscheiden zwischen zwei generellen Datenbanktypen: Nicht-XML-Datenbanken und XML-Datenbanken. Der Begriff "Datenbank" wird in zwei Bedeutungen verwendet: Generell werden damit alle Datenbanken bezeichnet; spezifisch bezieht er sich auf nicht-XML-Datenbanken. XML-Datenbanken werden immer als XML-Datenbanken (XML DB) bezeichnet. Diesen Unterschied sollten Sie immer vor Augen haben, da die Methode, wie die Datenbankdaten, die das Schema und die XML-Daten für das SPS bereitstellen, bei diesen beiden Arten von Datenbanken unterschiedlich ist.

- [Herstellen einer Verbindung zu einer Datenbank](#): Hier wird definiert, wie man eine Verbindung zu nicht-XML-Datenbanken einschließlich IBM DB2 Versionen unter Version 9 herstellt.
- [Auswahl von Datenbankdaten](#): Hier wird (separat für nicht-XML- und XML-Datenbanken) beschrieben, wie das Schema und die XML-Daten für das SPS in der Tabellenstruktur der Datenbank ausgewählt werden.
- [Das DB-Schema und die XML-Datenbankdatei](#): Beim Laden von Datenbanktabellen (aus nicht-XML-Datenbanken) generiert StyleVision temporäre auf der DB-Struktur bzw. den Daten basierende XML-Schema- bzw. Datenbankdateien und arbeitet mit diesen. Bei XML-Datenbanken werden das Schema und die XML-Dateien nicht von StyleVision generiert, sondern direkt von der DB - oder im Fall von Schemas - von einem anderen Dateipfad aus referenziert.
- [DB-Filter: Filtern von Datenbankdaten](#): Datenbankdaten, die in die temporäre XML-Datei geladen werden, können gefiltert werden.
- [SPS Design-Funktionen für Datenbanken](#): Im SPS stehen spezielle Datenbankfunktionen wie z.B. Datenbanksteuerelemente und Datenbankabfragen zur Verfügung.
- [Generieren von Ausgabedateien](#): In StyleVision kann eine große Bandbreite an Datenbankberichtsdateien generiert werden.

13.1 Datenbanken und StyleVision

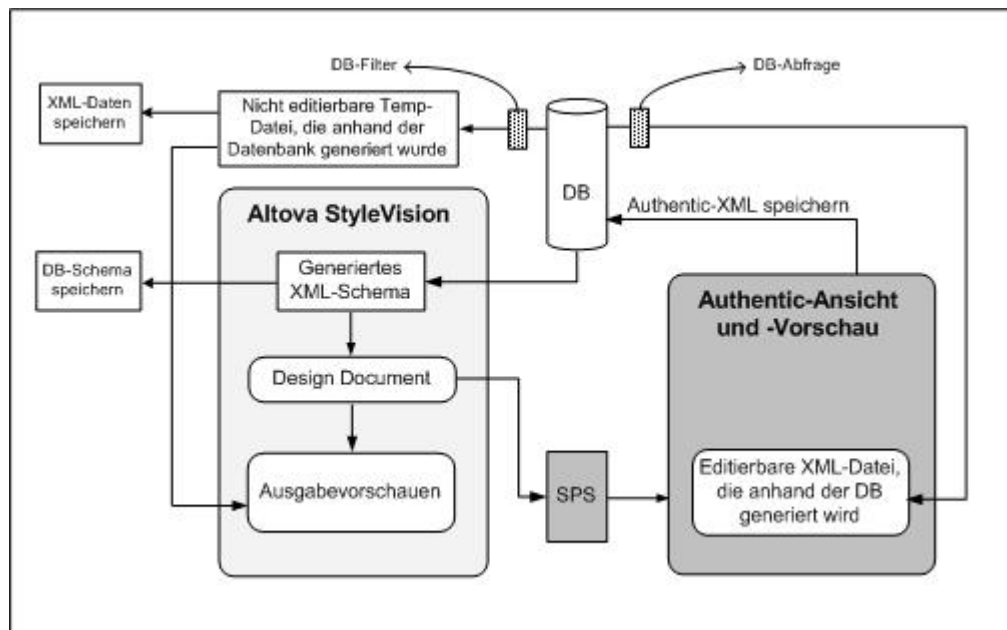
In StyleVision können Sie Datenbank-basierte SPSs erstellen. Mit Hilfe dieser Stylesheets können Sie:

- Datenbanken in der Authentic-Ansicht bearbeiten und
- Berichte anhand von Datenbanken generieren.

Nachdem Sie das SPS erstellt haben, können Sie Berichte in StyleVision anzeigen und Berichte im HTML- und RTF-Format erstellen. Außerdem können Sie die folgenden mit StyleVision generierten XML-Datenbankdateien speichern:

- XML-Schema basierend auf einer Datenbankstruktur (gilt nicht für XML-Datenbanken, bei denen bereits ein Schema zur Verfügung steht)
- XML-Datei, deren Struktur im generierten Schema definiert ist und deren Inhalt aus der Datenbank stammt (gilt nicht für XML-Datenbanken, bei denen Daten bereits im XML-Format zur Verfügung stehen)
- SPS, das Sie erstellt haben und das auf dem generierten Schema basiert
- XSLT Stylesheet für die HTML-Ausgabe (auf Basis des Designs des SPS)
- XSLT Stylesheet für die RTF-Ausgabe (auf Basis des Designs des SPS)
- HTML-Ausgabe
- RTF-Ausgabe

Die gespeicherte XML-Datei kann anschließend mit den erforderlichen XSLT Stylesheets verarbeitet werden. Dies bietet flexiblere Funktionen zur Berichtsgenerierung.



Anmerkung: Das XML-Schema und die XML-Dateien werden von StyleVision anhand von nicht-XML-Datenbanken generiert. Sie können deren Struktur oder Inhalt nicht für die Verwendung in der Authentic-Ansicht ändern, da die Struktur dieser Dateien mit der Struktur der nicht-XML-Datenbank in Zusammenhang steht. Die Bearbeitung der Datenbank und die Erstellung von Berichten anhand der Datenbank ist von der eindeutigen XML-Struktur abhängig, die von StyleVision anhand der Datenbank generiert wurde.

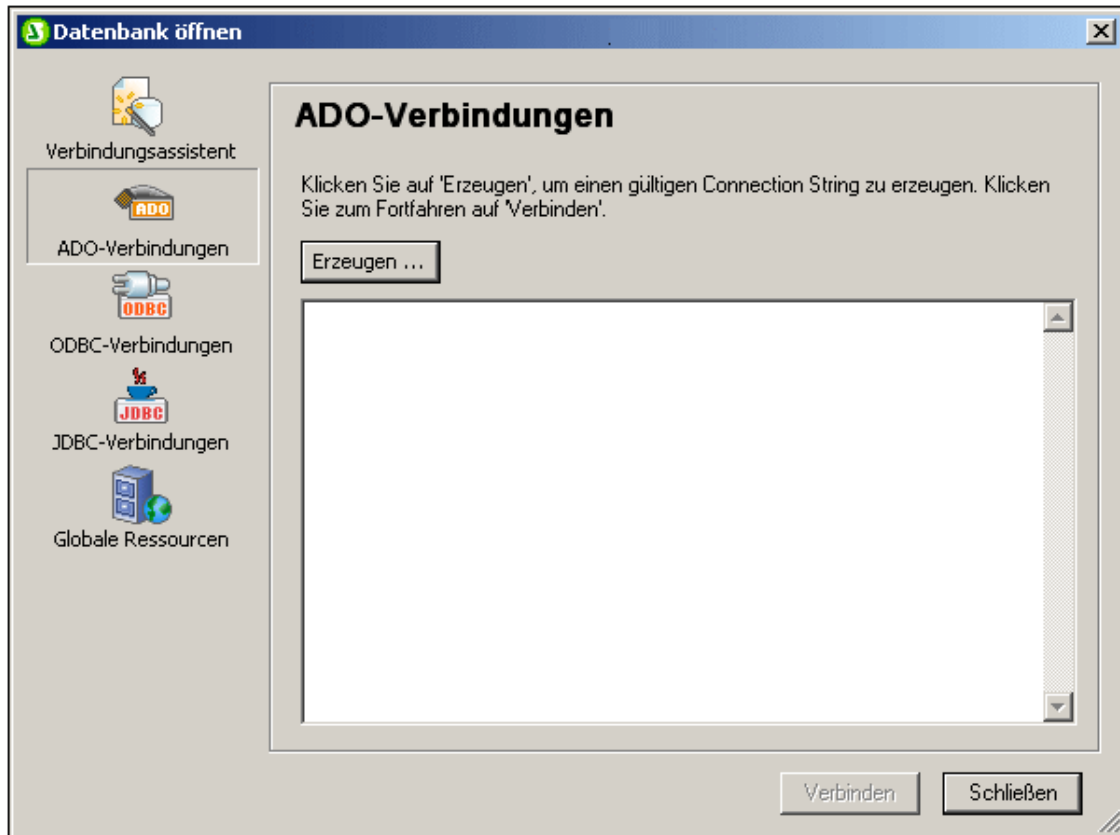
Grundlegende Mechanismen beim Arbeiten mit Datenbank-basierten SPSs

Im Folgenden sind die Schritte beim Erstellen und Verwenden von Datenbank-basierten SPSs beschrieben. Diese Schritte umfassen die beiden Verwendungszwecke von Datenbank-basierten StyleVision Power Stylesheets: Bearbeiten von Datenbanken und Erstellen von HTML- und RTF-Berichten anhand der Datenbank.

- [Herstellen einer Verbindung zur Datenbank](#) mit StyleVision. Während die [Verbindung hergestellt wird](#), können Sie angeben, welche Datenbanktabellen in der Datenbank aus dem XML-Schema herausgefiltert werden sollen.
- Sobald die Verbindung hergestellt ist, wird auf Basis der Struktur der Datenbank ein [temporäres XML-Schema](#) generiert, das dann im Schemafenster von StyleVision in Form einer Baumstruktur angezeigt wird. Im Fall von XML-Datenbanken wird ein bereits vorher (entweder in der DB oder unter einem Dateipfad) vorhandenes Schema referenziert.
- Außerdem werden auch temporäre StyleVision-interne [XML-Dateien erstellt](#). Eine davon ist nicht editierbar (*siehe Abbildung oben*). Sie wird für Vorschauen und als Quelle für die generierte XML-Datendatei verwendet. Die andere ist eine editierbare XML-Datei, welche in der Authentic-Ansicht angezeigt wird (*siehe Abbildung oben*). Wenn Änderungen, die in der Authentic-Ansicht an dieser Datei vorgenommen wurden, gespeichert werden (mit dem Befehl **Datei | XML-Authentic-Daten speichern...**), werden die Änderungen zurück in die Datenbank geschrieben. Die nicht editierbare XML-Datei wird, wenn nötig, jedes Mal beim erneuten Aufrufen der Ausgabevorschau oder beim Speichern der XML-Daten aktualisiert.
- Sie können in StyleVision [Filter auf oberster Ebene](#) definieren, um die in die nicht-editierbare XML-Datei importierten Daten für die Ausgabevorschauen und Berichte einzuschränken.
- Mit einer [DB-Abfrage](#) können Sie in der Authentic-Ansicht die Liste der in der Authentic-Ansicht angezeigten Datensätze einschränken. Sie wird nur beim Editieren verwendet.
- Wurden Änderungen in der Datenbank gespeichert, so wird die nicht-editierbare XML-Datei beim nächsten Aufrufen der Ausgabevorschau mit dem geänderten Inhalt aus der Datenbank aktualisiert und die aktualisierte Datei wird in der Vorschau angezeigt.
- Ein Datenbank-basiertes SPS wird auf dieselbe Art wie das Schema-basierte Standard-SPS erstellt: durch Ziehen der Nodes in das Design-Fenster, Einfügen von statischen Stylesheet-Komponenten, Zuweisen von Anzeigeeigenschaften usw. Diese Mechanismen werden in dieser Dokumentation beschrieben.

13.2 Herstellen einer Verbindung zu einer Datenbank

Wenn Sie entweder den Befehl **Datei | Neu | Neu von DB** oder den Befehl **Datei | Neu | Neu von XML-Spalte in IBM DB2** auswählen, wird das Dialogfeld "Datenbank öffnen" angezeigt (Abbildung unten). (Beachten Sie, dass die Befehle **Neu von DB** und **Neu von XML-Spalte in IBM DB2** auch in der Seitenleiste [Design-Übersicht](#) zur Verfügung stehen).



Die Optionen zum Herstellen einer Verbindung zur Datenbank sind:

- [Über einen Verbindungsassistenten](#), der Sie durch den Vorgang führt.
- [Über eine ADO-Verbindung](#)
- [Über eine ODBC-Verbindung](#)
- [Über eine globale Ressource](#).

Die einzelnen Optionen werden in den Unterabschnitten dieses Abschnitts näher beschrieben.

Anmerkung:

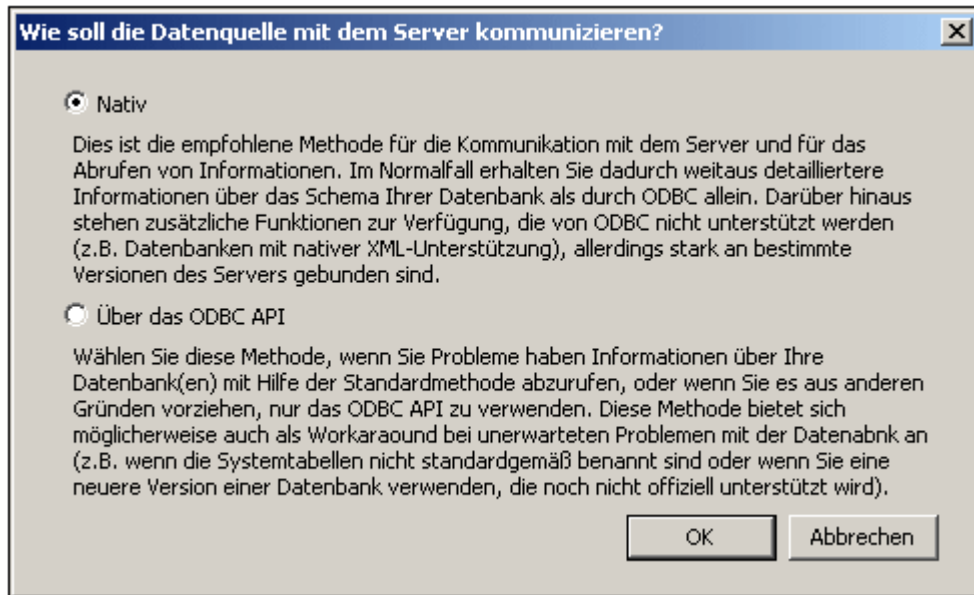
Beim Hinzufügen einer ODBC-Datenquelle für die IBM iSeries (vormals AS/400) wird ein Standard-Flag gesetzt, mit dem Abfrage-Timeouts aktiviert werden. Diese Einstellung muss **deaktiviert** werden, damit das SPS in StyleVision korrekt geladen werden kann. Beim Hinzufügen einer ODBC-Datenquelle (Benutzer/System-DSN) für den iSeries Access ODBC-Treiber, wird das Dialogfeld "iSeries Access for Windows ODBC Setup" geöffnet. Wählen Sie das Register "Performance" aus, klicken Sie auf die Schaltfläche "Advanced" und **deaktivieren** Sie das Kontrollkästchen "Allow query timeout".

Anmerkung: Wenn Sie die 64-Bit-Version von StyleVision verwenden, stellen Sie sicher, dass Sie Zugriff auf die 64-Bit-Datenbanktreiber haben, die für die entsprechenden

Datenbank benötigt werden.

Nativ oder ODBC?

Beim Herstellen der Verbindung werden Sie aufgefordert auszuwählen, ob die Verbindung nativ oder über ODBC hergestellt werden soll (*siehe Abbildung unten*). Aufgrund möglicher Einschränkungen bei einer ODBC-Verbindung wird empfohlen die Standardoption *Nativ* zu wählen. Wenn es Probleme mit der nativen Verbindung gibt, wählen Sie die Option "ODBC".



Auswählen der XML-Daten und des Schemas für das SPS

Nachdem die Verbindung zur Datenbank hergestellt wurde, müssen das Schema und die XML-Daten für das SPS generiert werden. Dies erfolgt auf Basis der Datenbankstruktur und der Daten. Es gibt zwei mögliche Situationen:

- Bei nicht-XML-Datenbanken wählen Sie die Tabelle(n) aus, für die das SPS erstellt wird. StyleVision generiert automatisch: (i) ein Schema, das auf der Struktur der Tabelle(n) basiert und (ii) temporäre XML-Dateien basierend auf diesem Schema, die die Daten in der/den ausgewählten Tabelle(n) enthalten. Informationen zum Auswählen der Tabellen finden Sie im Abschnitt [Auswählen der DB-Daten | Nicht-XML-Datenbanken](#).
- Im Fall von XML-Datenbanken müssen Sie zwei Dinge tun. Erstens [die XML-Zelle der Datenbank auswählen](#), in der die benötigten XML-Daten gespeichert sind. Diese XML-Daten werden als [XML-Arbeitsdatei](#) des SPS geladen. Zweitens müssen Sie [das Schema auswählen](#), auf dem das SPS basiert. Wie man die XML-Daten und das Schema auswählt, wird im Abschnitt, [Auswählen der DB-Daten | XML -Datenbanken](#) beschrieben.

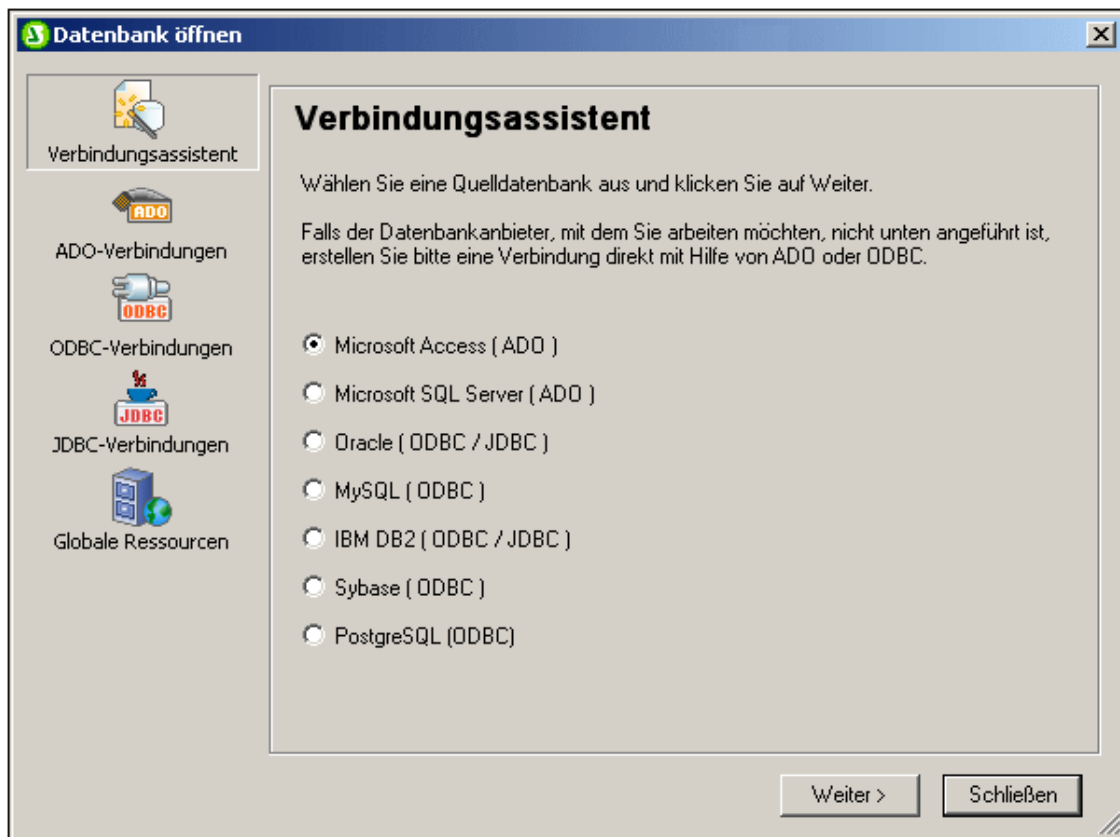
Verbindungsassistent

Bei Auswahl der Schaltfläche **Verbindungsassistent** im Dialogfeld "Datenbank öffnen" wird der Verbindungsassistent (siehe Abbildung unten) aufgerufen. Der Verbindungsassistent hilft Ihnen, eine Verbindung zu den gängigsten Datenbanktypen herzustellen. In diesem Abschnitt wird die Herstellung der Verbindung zu den folgenden Datenbanken beschrieben:

- einer [Microsoft Access-Datenbank](#) (einer nicht-XML-Datenbank). Des Weiteren wird hier beschrieben, wie man eine Verbindung zu anderen nicht-XML-Datenbanken herstellt.
- einer [IBM DB2-Datenbank](#) (einer XML-Datenbank; derzeit sind IBM DB2-Datenbanken die einzigen XML-Datenbanken, die unterstützt werden).

MS Access

Um eine Verbindung zu einer MS Access-Datenbank herzustellen, führen Sie die folgenden Schritte durch. Wählen Sie Microsoft Access (ADO) (Abbildung unten) und klicken Sie auf **Weiter**.



Suchen Sie im Dialogfeld "Verbindung zu MS Access herstellen" die MS Access-Datenbank und klicken Sie auf **Weiter**. Die Verbindung zur Datenquelle wird hergestellt. Informationen über die Auswahl von Tabellen in MS Access-Datenbanken finden Sie unter [Auswahl von Datenbankdaten](#).

Anmerkung: Beachten Sie bitte die folgenden Anmerkungen zu Nicht-XML-Datenbanken:

- Es werden die folgenden Anbieter (Treiber) empfohlen: (i) Für Microsoft SQL-Datenbanken: *Microsoft OLE DB Provider for SQL Server*, (ii) Für Oracle, MySQL, Sybase, und IBM DB2-Datenbanken (Versionen vor Version 9 sind Nicht-XML-DB), *Microsoft OLE DB Provider for ODBC Drivers*.

- Wenn zum Aufrufen der Datenbank ein Passwort benötigt wird, muss es im Connection String gespeichert werden.
- Alle Verbindungsinformationen werden im SPS in Form eines Connection String gespeichert. Es ist daher wichtig, dass auf allen Clients, die dieses SPS verwenden, derselbe Treiber installiert ist, damit sie den Connection String verwenden können.
- Bei Verwendung einer ADO-Verbindung werden hierarchische Beziehungen in Oracle- und Sybase-Datenbanken ignoriert. Anhand dieser Datenbanken werden nur flache Schemas generiert.
- Bei Herstellung einer ADO-Verbindung zu einem MS SQL Server über den Treiber SQL Server Native Client 10.0 müssen auf dem Register *All* des Dialogfelds "Data Link Properties" die folgenden Eigenschaftswerte definiert werden: (i) Setzen Sie als Eigenschaftswert von "Integrated Security" ein Leerzeichen; (ii) Setzen Sie den Eigenschaftswert von "Persist Security Info" auf `true`.

Netzwerkfreigaben und UNC-Pfade

Sie können jeden Ordner auf der Festplatte Ihres Computers "freigeben". Ein freigegebener Ordner kann von jedem Computer im Netzwerk aufgerufen werden und je nach Einstellung der Freigabe können Sie auch Schreibrechte für diesen Ordner haben. Ein freigegebener Ordner kann daher auf einem entfernten Rechner genauso verwendet werden, als handelte es sich dabei um einen lokalen Ordner.

Wenn Sie mit einem SPS arbeiten, das auf einer Datenbank basiert, die sich auf einer Festplatte befindet (z.B. einer MS Access DB), muss der Ordner, in dem sich die Datenbank befindet, als freigegeben gekennzeichnet werden. Für den Connections String, der auf die Datenbank verweist, wird dann ein UNC-Pfad verwendet. Auf diese Art kann das SPS eine Verbindung zur Datenbank herstellen, wenn es von Clients (wie z.B. der Authentic Browser) auf anderen Rechnern verwendet wird.

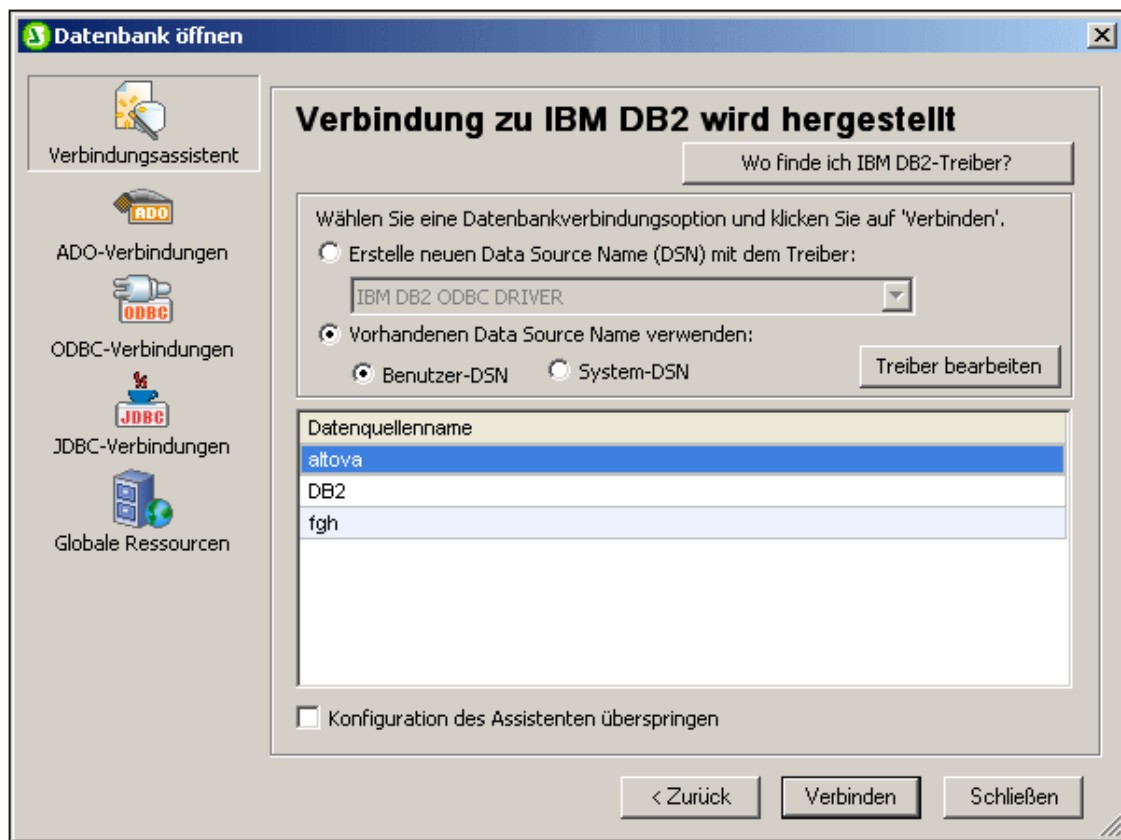
Das Einrichten der Netzwerkfreigabe für ein DB-basiertes SPS erfolgt in zwei Schritten.

- Der Ordner, in dem sich die DB befindet, muss freigegeben werden. (In Windows XP werden die Netzwerkfreigabeeinstellungen durch Rechtsklick auf den Ordner und Auswahl des Befehls **Freigabe und Sicherheit...**) ausgewählt. Außerdem müssen die erweiterten Einstellungen für die Dateifreigabe unter (**Arbeitsplatz | Extras | Ordneroptionen** und Deaktivieren des Kontrollkästchens "Einfache Dateifreigabe verwenden") aktiviert werden.
- Zugriff auf den freigegebenen Ordner und die DB vom entfernten Rechner aus. In einem DB-basierten SPS müssen Sie dazu den Connection String mit einem UNC-Pfad einrichten. Das Format des UNC-Pfads ist:
`\\servername\sharename\path\file.mdb`, wobei `servername` der Name des Servers ist, `sharename` der Name des freigegebenen Ordners, `path` der Pfad zur Datenbank und `file.mdb` der Name der MS Access DB im freigegebenen Ordner oder einem Unterordner des freigegebenen Ordners ist.

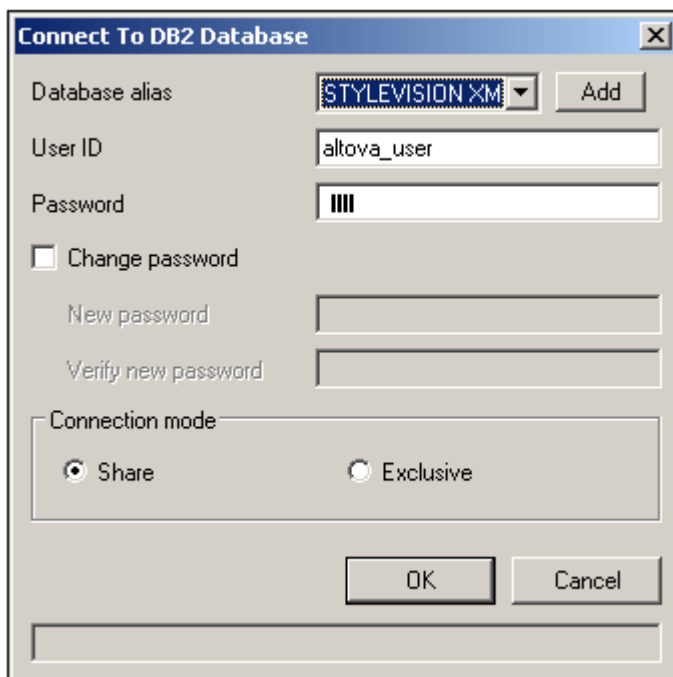
Anmerkung: Netzwerkfreigaben und UNC-Pfade sind für Datenbanken zu verwenden, für die kein Treiber benötigt wird und die sich auf einem Netzwerkserver befinden.

IBM DB2

Wählen Sie im ersten Bildschirm des Verbindungsassistenten (*siehe erste Abbildung in diesem Abschnitt*) den Eintrag IBM DB2 und klicken Sie auf **Weiter**. Daraufhin wird das Konfigurationsdialogfeld (*Abbildung unten*) angezeigt. In diesem Dialogfeld werden die DSNs der verfügbaren IBM DB2-Datenquellen aufgelistet. Wählen Sie den benötigten DSN aus der Liste aus. (Sie können auch einen neuen DSN erstellen, der zur Liste der vorhandenen DSNs hinzugefügt wird und zur Auswahl steht)



Wenn Sie auf **Weiter** klicken, wird das Dialogfeld "Connect to DB2 Database" (Screenshot unten), in das Sie die Benutzerinformationen (ID und Passwort) eingeben müssen.



Wenn Sie auf **OK** klicken wird die Verbindung zur Datenbank hergestellt. Nähere Informationen

finden Sie auch unter [ODBC-Verbindungen](#). Informationen zur Auswahl der erforderlichen Datenbankinformationen finden Sie im Abschnitt [Auswählen der Datenbankdaten](#).

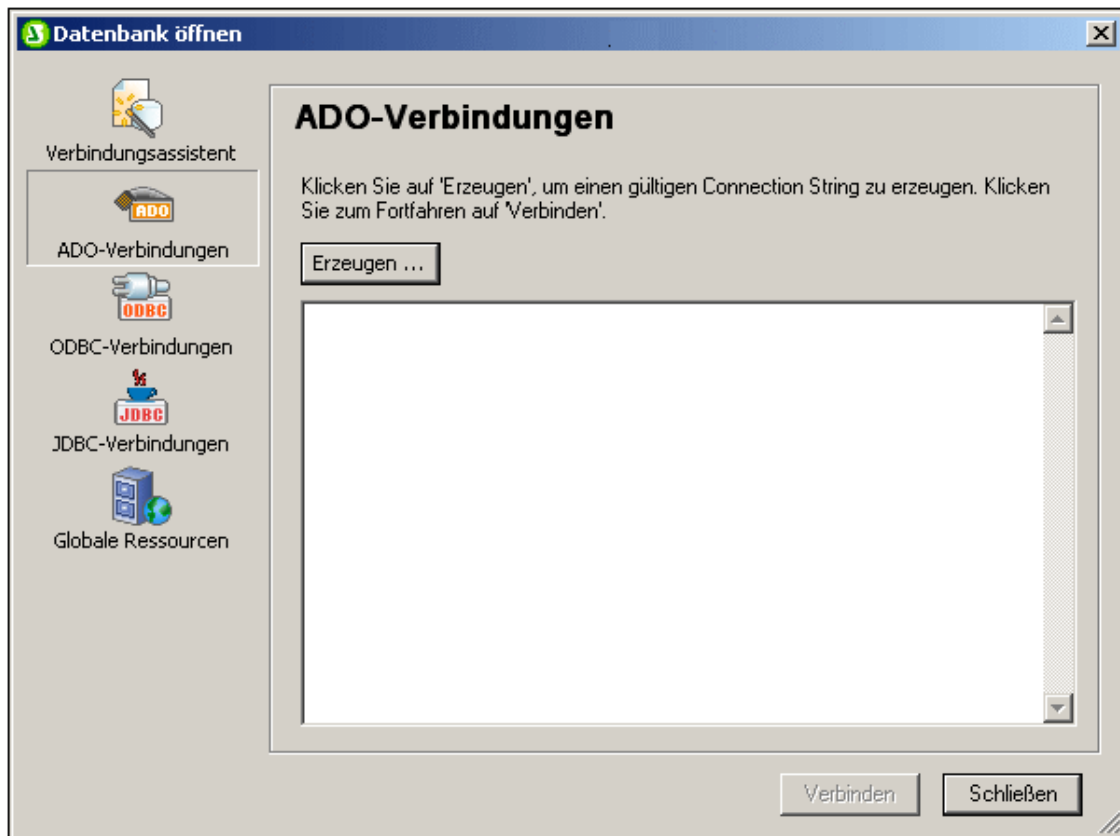
ADO-Verbindungen

Über die Option ADO-Verbindungen können Sie eine Verbindung zu einer Datenbank über ADO herstellen. In diesem Abschnitt finden Sie dazu eine Beschreibung für zwei Fälle:

- eine [Microsoft SQL Server-Datenbank](#) (als Beispiel für eine Nicht-XML-Datenbank).
- eine [IBM DB2-Datenbank](#) (eine XML-Datenbank; derzeit sind IBM DB2-Datenbanken die einzigen XML-Datenbanken, die unterstützt werden).
-

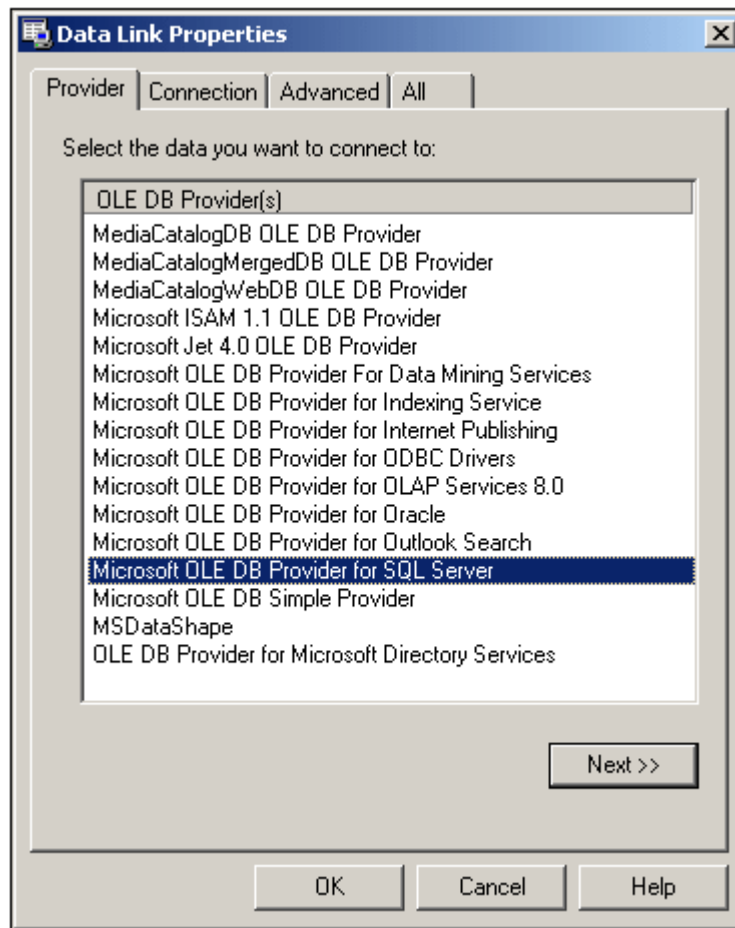
Microsoft SQL Server

Um eine Verbindung über ADO zu einer anderen Datenbank als einer DB vom Typ MS Access herzustellen, müssen Sie einen Connection String zu dieser DB erzeugen. Klicken Sie dazu im Dialogfeld "Datenbank öffnen" (*Abbildung unten*) auf die Schaltfläche **ADO-Verbindungen**.

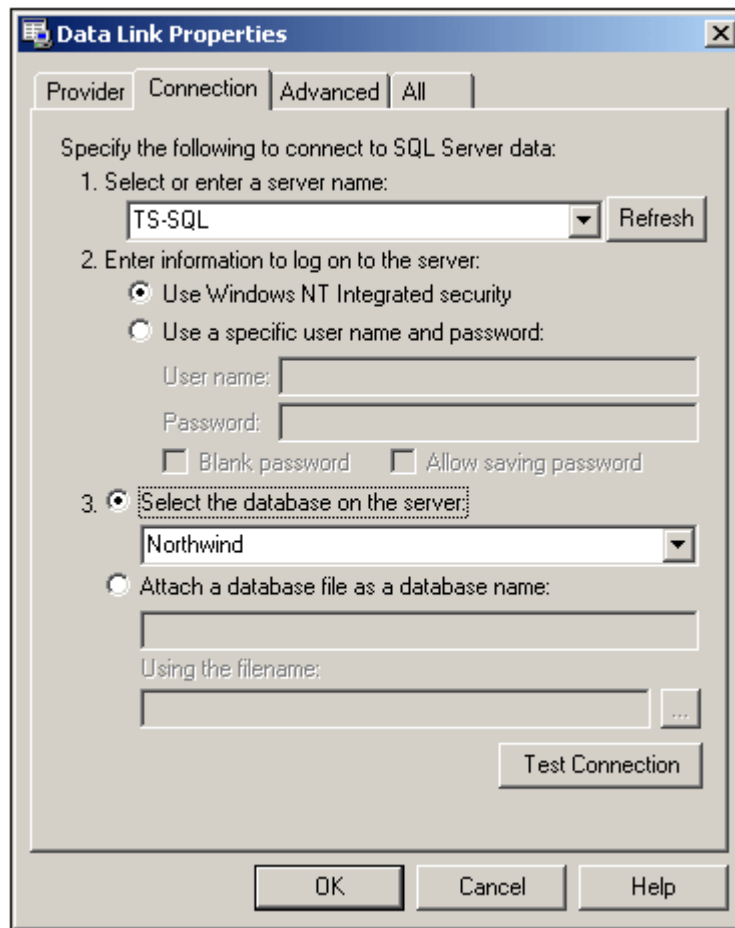


Erstellen Sie den ADO Connection String wie folgt:

1. Klicken Sie im Dialogfeld "ADO Connection String" auf **Erzeugen**. Daraufhin wird das Dialogfeld "Datenverknüpfungseigenschaften" angezeigt.



2. Wählen Sie den Provider (Treiber) für die Verbindung zur Datenbank aus. (Bei SQL Datenbanken empfehlen wir, *Microsoft OLE DB Provider for SQL Server*. Bei Oracle, MySQL, Sybase und IBM DB2-Datenbanken (vor Version 9) empfehlen wir die Verwendung von *Microsoft OLE DB Provider for ODBC Drivers*.) Klicken Sie anschließend auf **Weiter>>**. Daraufhin wird das Register "Verbindung" angezeigt.



3. Geben Sie den Namen des Servers ein, die Logon-Informationen für den Server und den Namen der Datenbank. All diese Informationen sind benutzerspezifisch. Wenn Sie für die Logon-Informationen ein Passwort benötigen, beachten Sie, dass Sie das Kontrollkästchen "Speichern des Kennworts zulassen" aktivieren müssen, um das Passwort im Connection String zu speichern, da die Verbindung sonst nicht zustande kommen kann.
4. Klicken Sie auf **Verbindung testen**, um zu testen, ob die Verbindung mit dem von Ihnen erstellten Connection String hergestellt werden kann. Wenn ein Passwort eingegeben wurde, wird dieses zum Testen der Verbindung verwendet. Das Passwort wird jedoch nur dann im Connection String gespeichert, wenn die Option "Speichern des Kennworts zulassen" aktiviert wurde (*siehe Schritt 3*).
5. Falls der Verbindungstest fehlschlägt, erstellen Sie den Connection String erneut mit den richtigen Einstellungen. Wenn der Verbindungstest klappt, klicken Sie auf **OK**, um den Vorgang fertigzustellen. Der von Ihnen erstellte Connection String wird in den Bereich "ADO-Verbindungen" eingegeben. Ein Passwort wird nur dann in den Connection String eingegeben, wenn Sie die Option "**Speichern des Kennworts zulassen**" aktiviert haben.
6. Klicken Sie auf **Weiter**. Die Verbindung zur Datenbank wird hergestellt und das Dialogfeld "Basistabelleninformationen auswählen" wird angezeigt.

Anmerkung: Beachten Sie bitte die folgenden Anmerkungen zu Nicht-XML-Datenbanken:

- Es werden die folgenden Anbieter (Treiber) empfohlen: (i) Für Microsoft SQL-Datenbanken: *Microsoft OLE DB Provider for SQL Server*; (ii) Für Oracle, MySQL, Sybase, und IBM DB2-Datenbanken (Versionen vor Version 9 sind

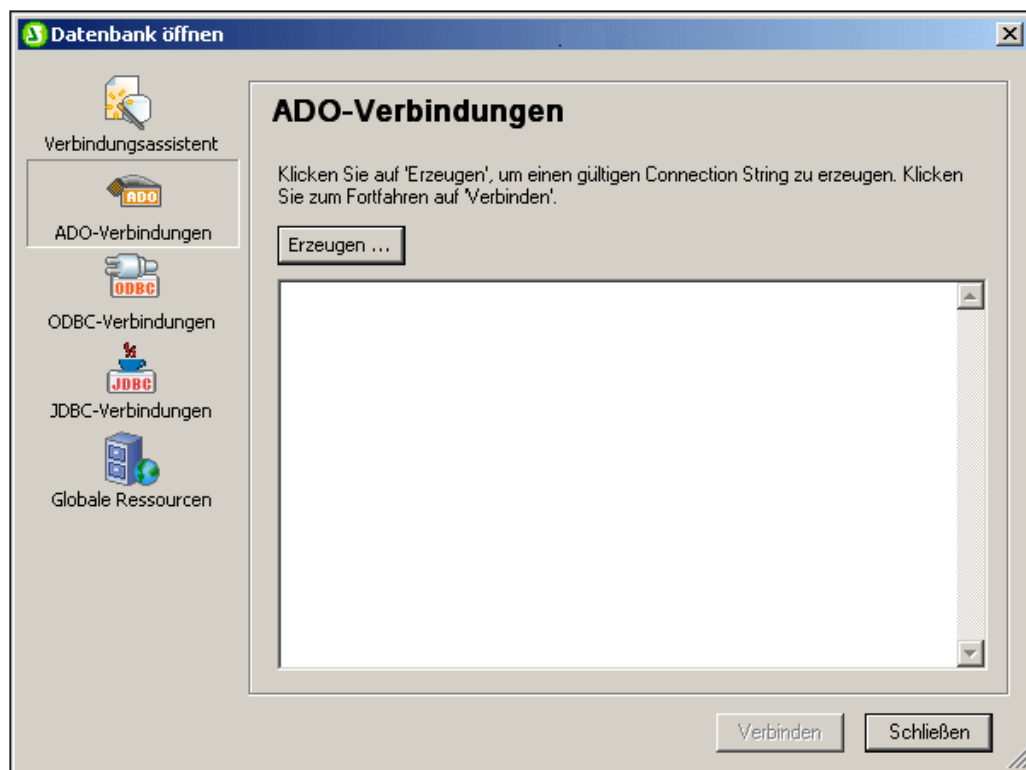
Nicht-XML-DB), *Microsoft OLE DB Provider for ODBC Drivers*.

- Wenn zum Aufrufen der Datenbank ein Passwort benötigt wird, muss es im Connection String gespeichert werden.
- Alle Verbindungsinformationen werden im SPS in Form eines Connection String gespeichert. Es ist daher wichtig, dass auf allen Clients, die dieses SPS verwenden, derselbe Treiber installiert ist, damit sie den Connection String verwenden können.
- Bei Verwendung einer ADO-Verbindung werden hierarchische Beziehungen in Oracle- und Sybase-Datenbanken ignoriert. Anhand dieser Datenbanken werden nur flache Schemas generiert.
- Bei Herstellung einer ADO-Verbindung zu einem MS SQL Server über den Treiber SQL Server Native Client 10.0 müssen auf dem Register *All* des Dialogfelds "Data Link Properties" die folgenden Eigenschaftswerte definiert werden: (i) Setzen Sie als Eigenschaftswert von "Integrated Security" ein Leerzeichen; (ii) Setzen Sie den Eigenschaftswert von "Persist Security Info" auf `true`.

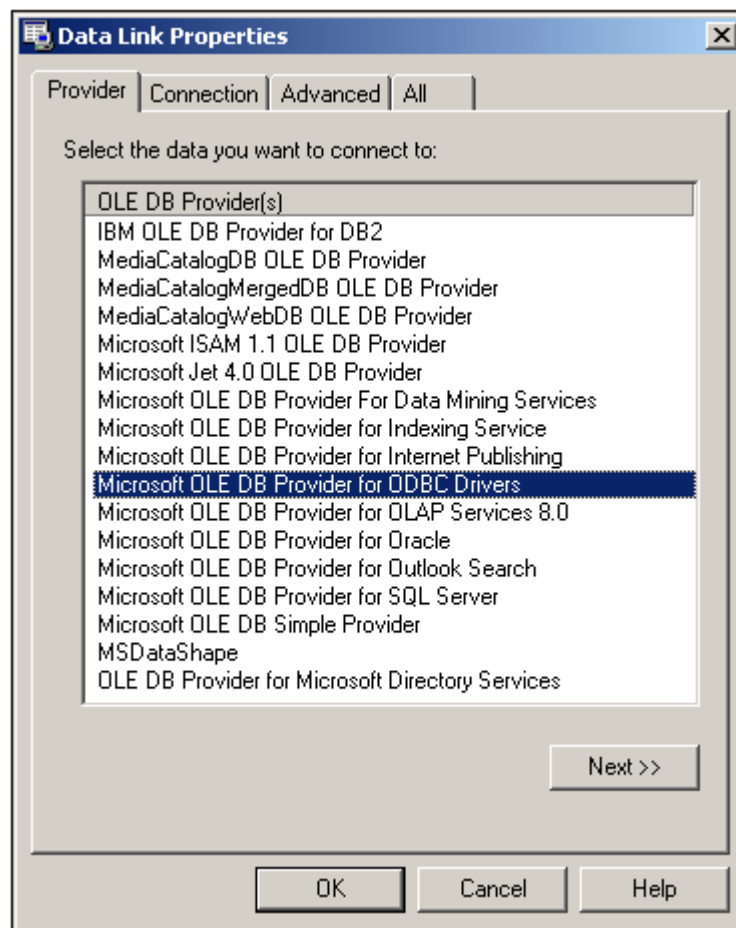
IBM DB2

So stellen Sie über ADO eine Verbindung zu einer IDB DB2-Datenbank her:

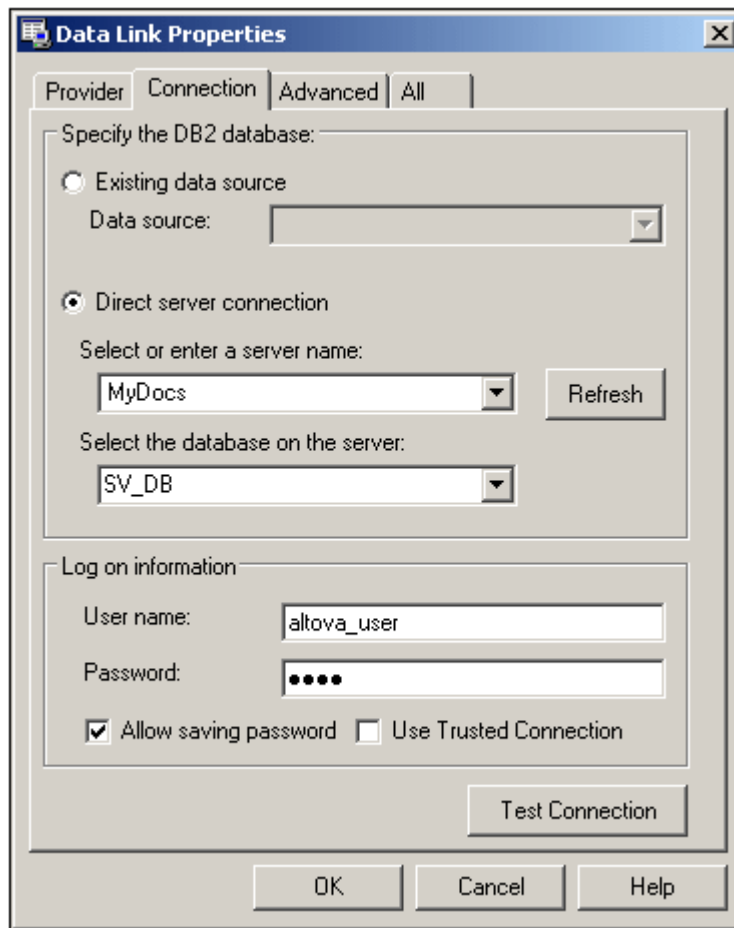
1. Wählen Sie im Dialogfeld "Datenbank öffnen" die Option **ADO-Verbindungen** (Abbildung unten) aus.



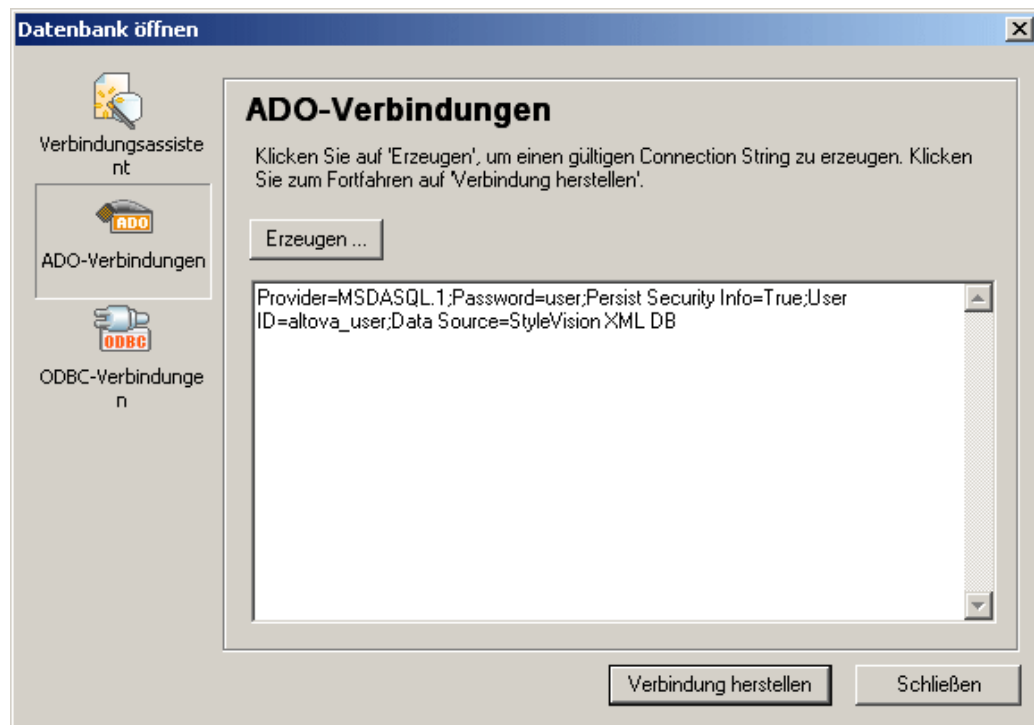
2. Klicken Sie auf die Schaltfläche **Erzeugen**. Das Dialogfeld "Datenverknüpfungseigenschaften" wird geöffnet.
3. Wählen Sie auf dem Register "Provider" "Microsoft OLE DB Provider for ODBC Drivers" aus und klicken Sie auf **Weiter**.



4. Wählen Sie auf dem Register "Verbindung" (*Abbildung unten*) den Eintrag "Direct Server Connection" und in der Auswahlliste für den Server den Server, auf dem die DB gespeichert ist. Wählen Sie anschließend die DB auf diesem Server aus.



5. Geben Sie Ihren Benutzernamen und Ihr Passwort ein.
6. Testen Sie die Verbindung zur DB, indem Sie auf die Schaltfläche **Verbindung testen** klicken. Wenn der Test fehlschlägt, müssen Sie die Verbindungsdaten richtigstellen.
7. Wenn der Test der Verbindung funktioniert hat, aktivieren Sie das Kontrollkästchen "Speichern des Kennworts zulassen", um die Passwortdaten im Connection String zu speichern. Das Register "Verbindung" des Dialogfelds "Datenverknüpfungseigenschaften" sollte anschließend folgendermaßen aussehen.
8. Klicken Sie auf **OK**. Der erzeugte Connection String wird in das Dialogfeld "Datenbank öffnen" eingegeben.

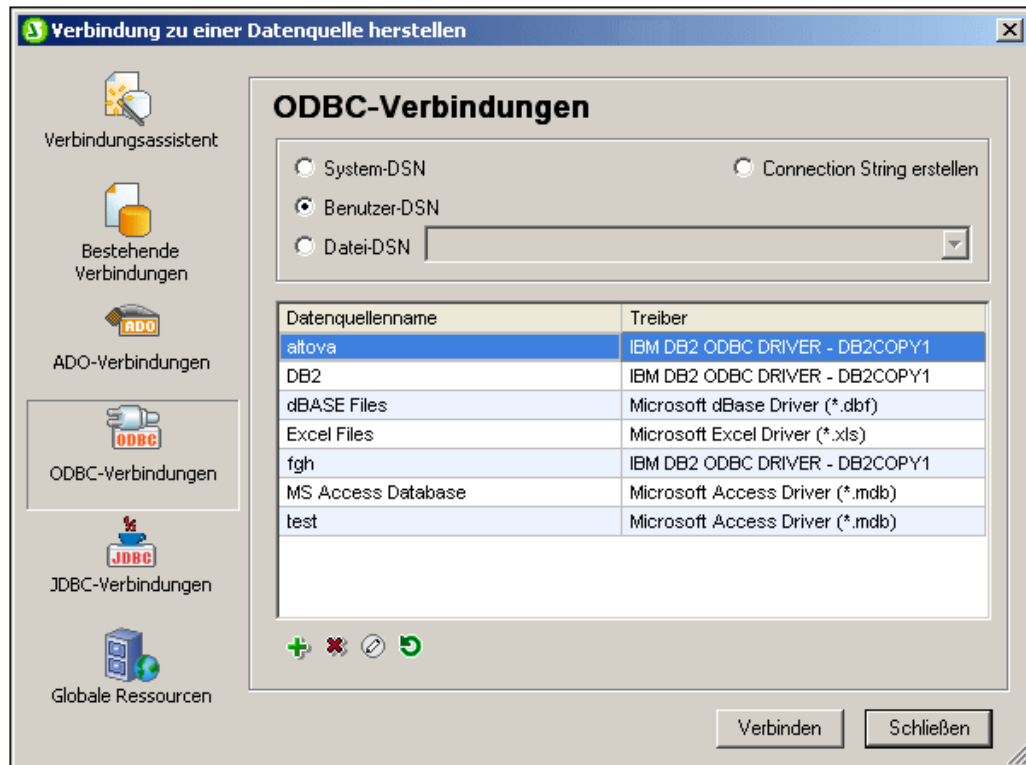


9. Klicken Sie auf die Schaltfläche **Verbindung herstellen**, um die Verbindung zur Datenbank herzustellen. Die Verbindung wird hergestellt und es erscheint ein Dialogfeld, in das Sie die erforderlichen Datenbankinformationen eingeben können. Eine Beschreibung dazu finden Sie in diesem Abschnitt unter [Auswählen der Datenbankdaten](#).

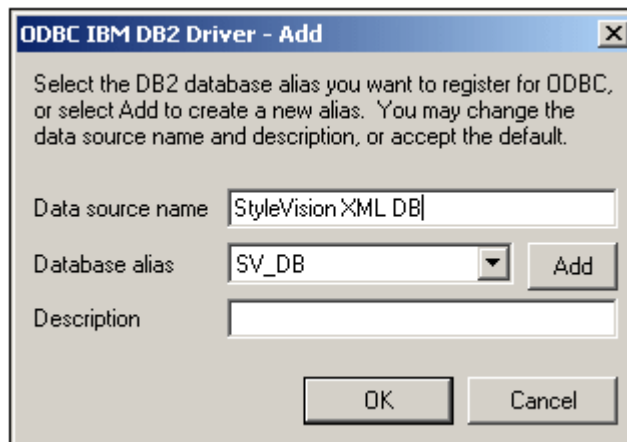
ODBC-Verbindungen

In diesem Abschnitt wird beschrieben, wie Sie über ODBC eine Datenbankverbindung herstellen. In der Anleitung unten wird eine Verbindung zu einer IBM DB2-Datenbank hergestellt. Dieselben Schritte werden aber auch bei anderen Datenbankarten, zu denen über ODBC eine Verbindung hergestellt werden kann, ausgeführt. Um eine Verbindung zu einer IBM DB2-Datenbank über ODBC herzustellen, gehen Sie folgendermaßen vor:

1. Wählen Sie im Dialogfeld "Verbindung zu einer Datenquelle herstellen" die Option **ODBC-Verbindungen** aus.



2. Wählen Sie eine der Optionen zur Definition eines Datenquellennamens (DSN). Wenn Sie "System-DSN" oder "Benutzer-DSN" wählen, werden die verfügbaren DSNs im Fenster "Datenquelle" angezeigt. Bei Auswahl von "Datei-DSN" werden Sie aufgefordert, den Ordner auszuwählen, der die Datei-DSNs enthält. Alternativ dazu können Sie einen Connection String zu DB erzeugen, indem Sie die Option "Connections String erstellen" auswählen.
3. Wenn Sie einen DSN zu den DSNs in der Liste der Datenquellen hinzufügen möchten, klicken Sie auf den Dropdown-Pfeil der Auswahlliste "Neuen DSN erstellen" ;
4. Wählen Sie im Dialogfeld "ODBC DSN erstellen", das nun angezeigt wird, den gewünschten Treiber aus und klicken Sie anschließend auf die Schaltfläche **Benutzer-DSN** oder **System-DSN**.
5. Wählen Sie im Dialogfeld, das daraufhin angezeigt wird (*Abbildung unten*), den DB-Alias aus und geben Sie ihm einen DSN.



6. Klicken Sie zum Fertigstellen auf **OK**. Die DB wird als Datenquelle zur Liste im Bereich "Datenquelle" hinzugefügt.
7. Nachdem Sie die Datenquelle (über die Option System-DSN oder Benutzer-DSN) oder eine DSN-Datei (Option Datei-DSN) ausgewählt oder einen Connection String (Option "Connection String") erstellt haben, klicken Sie auf **Verbindung herstellen**.
8. Wenn Sie nach Ihrer Benutzer-ID und dem Passwort gefragt werden, geben Sie diese ein und klicken Sie auf **OK**. Daraufhin wird die Verbindung hergestellt und es erscheint ein Dialogfeld, in dem Sie die benötigten Datenbankinformationen auswählen können. Eine Anleitung dazu finden Sie im Abschnitt [Auswählen der Datenbankdaten](#).

Anmerkung: Sie können die Liste im Bereich "Datenquelle" über die Schaltflächen *Edit Selected DSN*, *Remove Selected DSN* und *Refresh Listed DSNs* am unteren Rand des Bereichs "ODBC-Verbindungen" bearbeiten (wenn die Option System-DSN oder Benutzer-DSN ausgewählt ist).



Ausgewählten DSN bearbeiten.



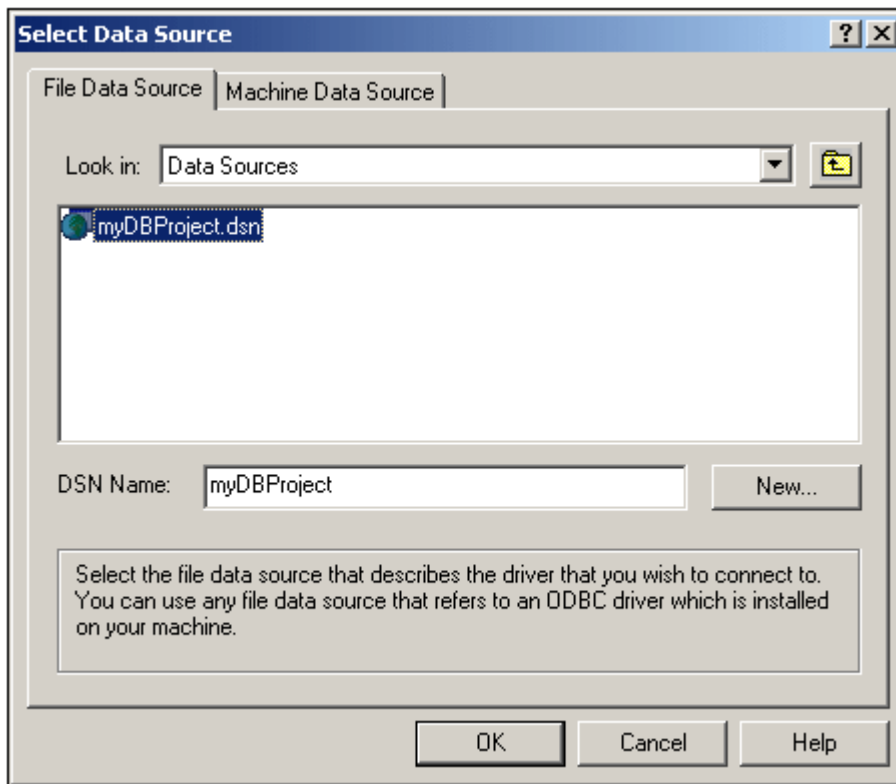
Ausgewählten DSN entfernen.



Aufgelistete DSNs aktualisieren.

Erstellung eines Connection String

Wenn Sie im Bereich "ODBC-Verbindungen" auf das Optionsfeld "Connection String erstellen" klicken, erscheint das Dialogfeld "Select Data Source" (*Abbildung unten*). Sie können entweder einen Datei-DSN (auf dem Register "File Data Source") oder eine Datenquelle auswählen, die auf Ihrem Rechner zur Verfügung steht (wird auf dem Register "Machine Data Source" aufgelistet).



Wenn Sie auf **OK** klicken, erscheint ein Dialogfeld, in dem Sie nach Benutzerdaten wie u.a. Benutzer-ID und Passwort gefragt werden. Wenn Sie in diesem Dialogfeld auf **OK** klicken, wird der Connection String in den Bereich "ODBC-Verbindungen" eingegeben.

Anmerkung: Wenn zum Herstellen einer Verbindung zu einer MS Access-Datenbank eine ODBC-Verbindung verwendet wird, können die Daten in der Datenbank nicht bearbeitet werden.

JDBC-Verbindungen

In diesem Abschnitt wird beschrieben, wie Sie über JDBC eine Verbindung zu einer Datenbank herstellen. In der Anleitung unten, wird beschrieben, wie Sie eine Verbindung zu einer IBM DB2-Datenbank herstellen. Die Anleitung gilt aber auch für andere Arten von Datenbanken, die über JDBC verbunden werden können.

Schritte vor dem Herstellen der Verbindung

Bevor Sie über JDBC eine Verbindung zu einer Datenbank herstellen, gehen Sie folgendermaßen vor:

1. Vergewissern Sie sich, dass das Java Runtime Environment (JRE) installiert ist. Ist dies nicht der Fall, dann installieren Sie es. Verwenden Sie für einen 32-Bit-Rechner ein 32-Bit JRE und für einen 64-Bit-Rechner ein 64-Bit JRE.
2. Installieren Sie einen JDBC-Treiber. Dazu ist keine spezielle Installation vonnöten. Kopieren Sie einfach den Treiber in ein lokales Verzeichnis, z.B. `c:\jdbc`. Beachten Sie, dass JDBC-Treiber (d.h. Jar-Dateien) plattformunabhängig sind.
3. Richten Sie den `CLASSPATH` ein, sodass er den Pfad enthält, unter dem der JDBC-Treiber gespeichert ist. (Die Applikation liest die `CLASSPATH`-Umgebungsvariable um den JDBC-Treiber zu finden.) Um den `CLASSPATH`, aufzurufen und zu bearbeiten, gehen Sie folgendermaßen vor: Klicken Sie auf **Start | Systemsteuerung | System | Erweitert | Umgebungsvariablen**. Wählen Sie im Dialogfeld "Umgebungsvariablen", das daraufhin angezeigt wird, entweder die Benutzerumgebungsvariable `CLASSPATH` oder die Systemumgebungsvariable `CLASSPATH` aus und klicken Sie auf die Schaltfläche **Bearbeiten**. Fügen Sie den Pfad zum JDBC-Treiber zum `CLASSPATH` hinzu. Beispiel: `CLASSPATH=C:\jdbc\sqljdbc.jar; C:\jdbc\db2jcc.jar; .`
4. Melden Sie sich beim System ab und dann wieder an, damit die `CLASSPATH` Änderungen aktiv werden.

Anmerkung: Wenn Sie den kompletten IBM DB2- oder Oracle Client installieren, wird der `CLASSPATH` automatisch mit den JDBC-Treibern der entsprechenden Pakete befüllt. Nähere Informationen finden Sie in den Beschreibungen weiter unten.

Herstellen einer Verbindung über JDBC

Um über JDBC eine Verbindung herzustellen, gehen Sie folgendermaßen vor:

1. Wählen Sie im Dialogfeld "Verbindung zu einer Datenquelle herstellen" die Option **JDBC-Verbindungen** aus. Daraufhin wird das Fenster "JDBC-Verbindungen" angezeigt (*Abbildung unten*).



2. Wählen Sie aus der Dropdown-Liste *Treiber* einen JDBC-Treiber aus (alle gefundenen Treiber, d.h. alle, die sich in der `CLASSPATH` Umgebungsvariablen befinden, werden aufgelistet). Geben Sie in das Textfeld *Datenbank-URL* einen Connection String ein und geben Sie gegebenenfalls einen Benutzernamen und ein Passwort ein. Im Folgenden finden Sie die Connection String-Syntax für gebräuchliche Datenbanken. Zu jeder Datenbank ist ein Beispiel-Connection String aufgelistet.

Oracle

```
jdbc:oracle:thin:[ user/password]@//[ host][: port]/SID
jdbc:oracle:thin:@//abcd234/ORA11
```

IBM DB2

```
jdbc:db2://host_name:port/dbname
jdbc:db2://MyDB2:50000/boz
```

MySQL

```
jdbc:mysql://host_name:port/dbname
jdbc:mysql://MyDB2:3306/moz
```

MSSQL

```
jdbc:sqlserver://host:port;datasenname=name;user=name;password=Pwd
jdbc:sqlserver://abcd38:1433;datasenname=coz
jdbc:sqlserver://Q5;DatabaseName=coz;SelectMethod=Cursor
```

PostgreSQL jdbc:postgresql://host:port/database
L

jdbc:postgresql://abc993:5432/qanoz

Sybase jdbc:sybase:Tds:host:port/dbname

jdbc:sybase:Tds:abc12:2048/QUE

3. Klicken Sie auf die Schaltfläche **Verbinden**. Daraufhin wird die Verbindung zur Datenquelle hergestellt. Informationen über die nachfolgenden Dialogfelder finden Sie unter [Auswählen der Datenbankdaten](#).

Schritt für Schritt: MSSQL, MySQL, PostGre und andere Nicht-XML-Datenbanken

Unten finden Sie eine schrittweise Anleitung für die Herstellung einer Verbindung zu einer Nicht-XML-Datenbank über JDBC.

1. JDBC JAR-Dateien (Treiberdateien): Kopieren Sie die Dateien in einen lokalen Ordner auf Ihrem Rechner. Fügen Sie anschließend zur Windows `CLASSPATH` Variablen den vollständigen Pfad und Dateinamen hinzu, z.B.: `C:\jdbc\sqljdbc.jar`;
`C:\jdbc\db2jcc.jar` .
2. Melden Sie sich ab und wieder an, um die Änderungen am `CLASSPATH` aktiv zu machen.
3. Starten Sie StyleVision und rufen Sie das Dialogfeld "Verbindung zu einer Datenbank herstellen" auf.
4. Im Fenster "JDBC-Verbindungen" werden die gefundenen JDBC-Treiber in der Dropdown-Liste *Treiber* aufgelistet. Wenn die Dropdown-Liste leer ist, vergewissern Sie sich, dass die Datei `altovadb.jar` im Ordner `C:\Program Files\Altova\Common 2012\jar` vorhanden ist.
5. Stellen Sie wie im Abschnitt oben unter *Herstellen einer Verbindung über JDBC* beschrieben, eine Verbindung zu einer Datenbank her.

Schritt für Schritt: Oracle

Unten finden Sie eine schrittweise Anleitung für die Herstellung einer Verbindung zu einer Oracle-Datenbank über JDBC. Wenn Sie die XML- und Datenbankfunktionen der Oracle-Datenbank nicht benötigen, befolgen Sie die Anleitung für Nicht-XML-Datenbanken. Der Installationsordner des Oracle Client wird in der Anleitung unten durch den Platzhalter `%ORACLE_HOME%` angegeben.

1. Installieren Sie Oracle Client Software, wobei die OCI- und ODBC-Funktionen aktiviert sind. Wenn bereits ein Oracle Client installiert ist, überprüfen Sie, ob die zwei unten aufgelisteten jar-Dateien vorhanden sind:
`%ORACLE_HOME%\LIB\xmlparserv2.jar`
`%ORACLE_HOME%\RDBMS\jlib\xdb.jar`
2. Fügen Sie die folgenden Dateien zur Windows `CLASSPATH` Umgebungsvariablen hinzu:
`%ORACLE_HOME%\jdbc\lib\ojdbc6.jar`
`%ORACLE_HOME%\LIB\xmlparserv2.jar`
`%ORACLE_HOME%\RDBMS\jlib\xdb.jar`
3. Melden Sie sich ab und wieder an, um die `CLASSPATH` Änderungen aktiv zu machen.
4. Starten Sie StyleVision und rufen Sie das Dialogfeld "Verbindung zu einer Datenbank

- herstellen" auf.
5. Im Fenster "JDBC-Verbindungen" werden die gefundenen JDBC-Treiber in der Dropdown-Liste *Treiber* aufgelistet. Wenn die Dropdown-Liste leer ist, vergewissern Sie sich, dass die Datei `altovadb.jar` im Ordner `C:\Program Files\Altova\Common 2012\jar` vorhanden ist.
 6. Stellen Sie wie im Abschnitt oben unter *Herstellen einer Verbindung über JDBC* beschrieben, eine Verbindung zu einer Datenbank her.

Schritt für Schritt: IBM DB2

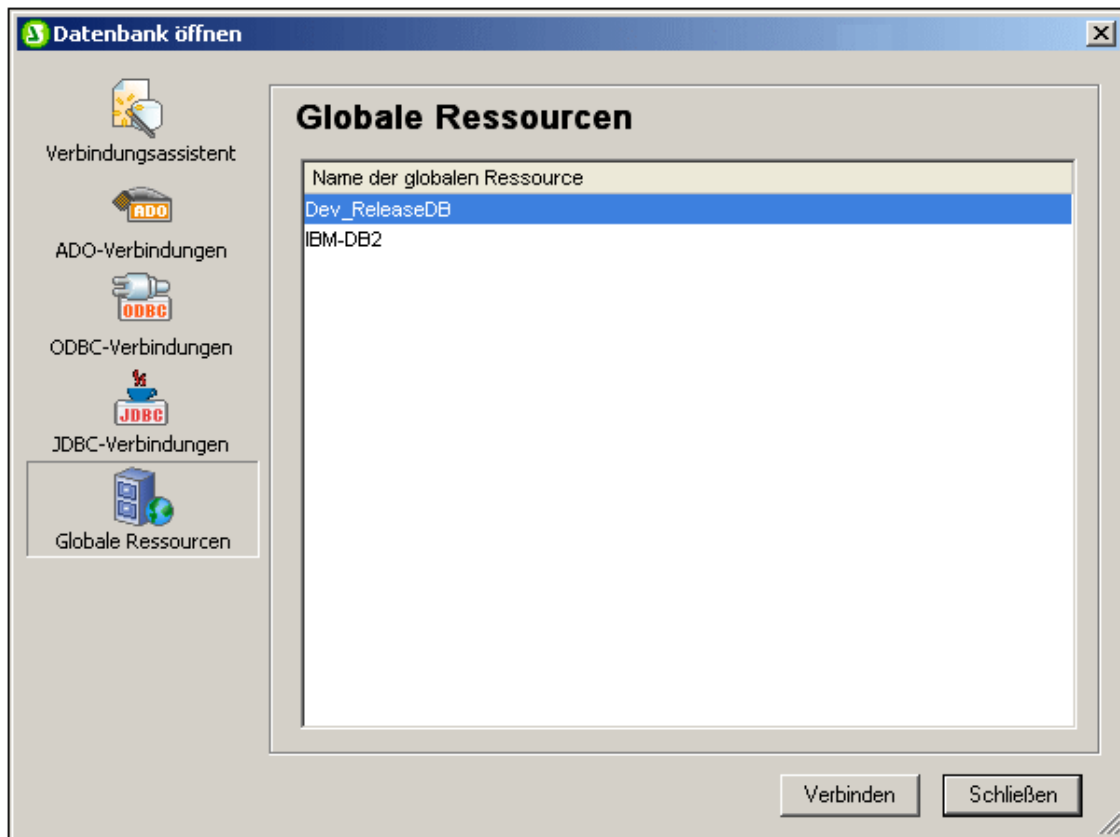
Unten finden Sie eine schrittweise Anleitung für die Herstellung einer Verbindung zu einer IBM DB2-Datenbank über JDBC.

1. Wenn bereits ein IBM DB2 Client installiert wurde, müssen Sie nichts unternehmen, da der `CLASSPATH` bereits während der Installation eingerichtet wurde.
2. Wenn kein IBM DB2 Client installiert wurde, fügen Sie die IBM DB2 JDBC jar-Treiberdateien `db2jcc.jar` und `db2jcc_license_cu.jar` zum Windows `CLASSPATH` hinzu. Melden Sie sich ab und wieder an, um die `CLASSPATH` Änderungen aktiv zu machen.
3. Starten Sie StyleVision und rufen Sie das Dialogfeld "Verbindung zu einer Datenbank herstellen" auf.
4. Im Fenster "JDBC-Verbindungen" werden die gefundenen JDBC-Treiber in der Dropdown-Liste *Treiber* aufgelistet. Wenn die Dropdown-Liste leer ist, vergewissern Sie sich, dass die Datei `altovadb.jar` im Ordner `C:\Program Files\Altova\Common 2012\jar` vorhanden ist.
5. Stellen Sie wie im Abschnitt oben unter *Herstellen einer Verbindung über JDBC* beschrieben, eine Verbindung zu einer Datenbank her.

Anmerkung: Wenn eine Datenbankverbindung über JDBC erfolgt, ist aufgrund mangelnder vom Treiber zurückgegebenen Informationen (i) die Bearbeitung von Daten in Tabellen ohne Primärschlüssel nicht möglich; (ii) und die Ausführung einer Datenbearbeitung im SQL-Editor funktioniert nicht.

Globale Ressourcen

Über die Option "Globale Ressourcen" können Sie eine Verbindung zu einer Datenbank herstellen, die als [globale Ressource definiert wurde](#). Der Vorteil dabei ist, dass Sie die Datenbankquelle für Ihr SPS mit Hilfe einer globalen Ressource schnell und einfach durch [Wechseln der aktiven Konfiguration](#) wechseln können. Bevor Sie versuchen über eine globale Ressource eine Verbindung zu einer Datenbank herzustellen, stellen Sie sicher, dass für die gewünschte Datenbank eine globale Ressource erstellt wurde.



Wenn Sie im Dialogfeld "Datenbank öffnen" auf die Schaltfläche "Globale Ressourcen" klicken, werden alle globalen Ressourcen vom Typ Datenbank, die in der derzeit aktiven [XML-Datei für globale Ressourcen](#) definiert wurden, angezeigt. Wählen Sie die gewünschte globale Ressource aus und klicken Sie auf **Verbindung herstellen**. Wenn die ausgewählte globale Ressource mehr als eine Konfiguration hat, wird die Datenbankressource für die gerade aktive Konfiguration (überprüfen Sie diese unter **Extras | Aktive Konfiguration** oder über die Symbolleiste "Globale Ressourcen") verwendet und die Verbindung wird hergestellt. Sie müssen nun auswählen, welche Datenstrukturen und Daten verwendet werden sollen (Beschreibung siehe [Auswählen der Datenbankdaten](#)).

13.3 Auswählen der Datenbankdaten

Je nachdem, ob es sich bei der Verwendung der Datenbank um eine Nicht-XML-Datenbank (wie z.B. MS Access) oder eine XML-Datenbank handelt (IBM DB2 Version 9.0 usw.), müssen bei der Auswahl des Schemas und der XML-Daten, die im SPS verwendet werden sollen, eine oder mehrere Tabellen, eine Zelle oder ein bestimmtes Schema ausgewählt werden. Die Auswahl des Schemas und der XML-Daten bezeichnen wir als Auswahl der Datenbankdaten. Diese Auswahl erfolgt sofort, nachdem die Verbindung zur Datenbank hergestellt wurde.

Wie die Datenbankdaten ausgewählt werden, hängt von der Art der Datenbank ab, zu der die Verbindung hergestellt wird:

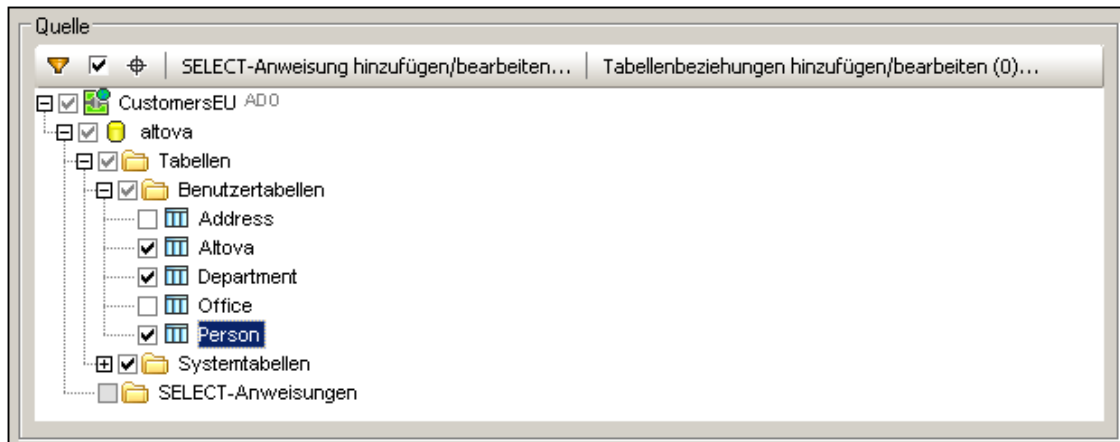
- Im Fall von [Nicht-XML-Datenbanken](#) wird/werden die Tabelle(n) ausgewählt, für die das SPS erstellt wird. StyleVision generiert automatisch: (i) ein Schema, das auf der Struktur der Tabelle(n) basiert, sowie (ii) temporäre XML-Dateien, die auf diesem Schema basieren und die die Daten in der/den ausgewählten Tabelle(n) enthalten. Eine Beschreibung zur Auswahl der Tabellen finden Sie im Abschnitt [Auswählen der Datenbankdaten | Nicht-XML-Datenbanken](#).
- Im Fall von [XML-Datenbanken](#) müssen Sie zwei Dinge tun. Erstens müssen Sie [die XML-Zelle der Datenbank, in der die benötigten XML-Daten gespeichert sind, auswählen](#). Diese XML-Daten werden als die [XML-Arbeitsdatei](#) des SPS geladen. Zweitens müssen Sie [das Schema auswählen](#), auf dem das SPS basieren soll. Eine Beschreibung zur Auswahl der XML-Daten und des Schemas finden Sie im Abschnitt [Auswählen der Datenbankdaten | XML-Datenbanken](#).

Nicht-XML-Datenbanken

Nachdem Sie eine Verbindung zu einer Nicht-XML-Datenbank hergestellt haben, wird das Dialogfeld "Select Tables" angezeigt. Das Dialogfeld besteht aus zwei Bereichen. Im oberen Bereich, der eine grafische Darstellung der Tabellen in der Datenbank enthält, wählen Sie die für das SPS benötigten Tabellen aus. StyleVision generiert auf Basis der ausgewählten Tabellen ein XML-Schema und XML-Datendateien. Im unteren Vorschaubereich des Dialogfelds "Select Tables" sehen Sie eine Vorschau auf den Inhalt der ausgewählten Tabelle.

Der Bereich "Quelle"

Im Bereich "Quelle" sehen Sie eine grafische Darstellung der Datenbanktabellen (*siehe Abbildung unten*). Wählen Sie die für das SPS benötigten Tabellen durch Anklicken der jeweiligen Kontrollkästchen aus.



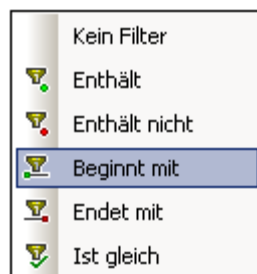
Sie Symbolleiste des Bereichs "Quelle" (*Abbildung unten*) enthält drei Schaltflächen, nämlich von links nach rechts: **Ordnerinhalt filtern**, **Nur ausgewählte Objekte anzeigen** und **Objektsuche**. Mit der Schaltfläche **Nur ausgewählte Objekte anzeigen** können Sie entweder alle Tabellen oder alle ausgewählten Tabellen einblenden.



Filtern von Ordnerinhalt

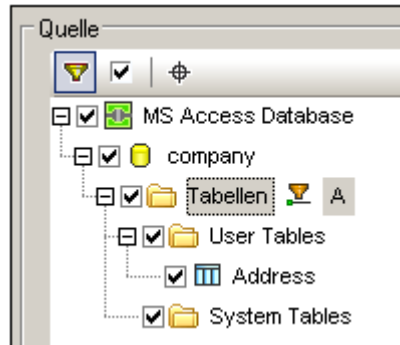
So filtern Sie Objekte im Bereich "Quelle":

1. Klicken Sie in der Symbolleiste des Bereichs "Quelle" auf die Schaltfläche **Ordnerinhalt filtern**. Neben dem Ordner "Tabellen" erscheint das Filtersymbol.
2. Klicken Sie auf das Filtersymbol neben dem Ordner "Tabellen" und wählen Sie die Filteroptionen aus dem Popup-Menü aus (*Abbildung unten*), z.B. *Beginnt mit*.



3. Geben Sie im Eingabefeld, das angezeigt wird, den Filterstring ein (in der Abbildung

unten ist der Filterstring für den Ordner `Tabellen A`). Der Filter wird während der Eingabe angewendet.



Die Objektsuche

Um ein bestimmtes Datenbankobjekt nach seinem Namen zu suchen, können Sie die Objektsuche des Browser-Fensters verwenden. Dies funktioniert folgendermaßen:

1. Klicken Sie in der Symbolleiste des Bereichs "Quelle" auf die Schaltfläche "Objektsuche". Eine Auswahlliste erscheint am unteren Rand des Browsers.
2. Geben Sie den Suchstring in das Eingabefeld dieser Liste ein, z.B. `Altova` (*Abbildung unten*). Wenn Sie auf den Dropdown-Pfeil klicken, werden alle Objekte, die den Suchstring enthalten, angezeigt.



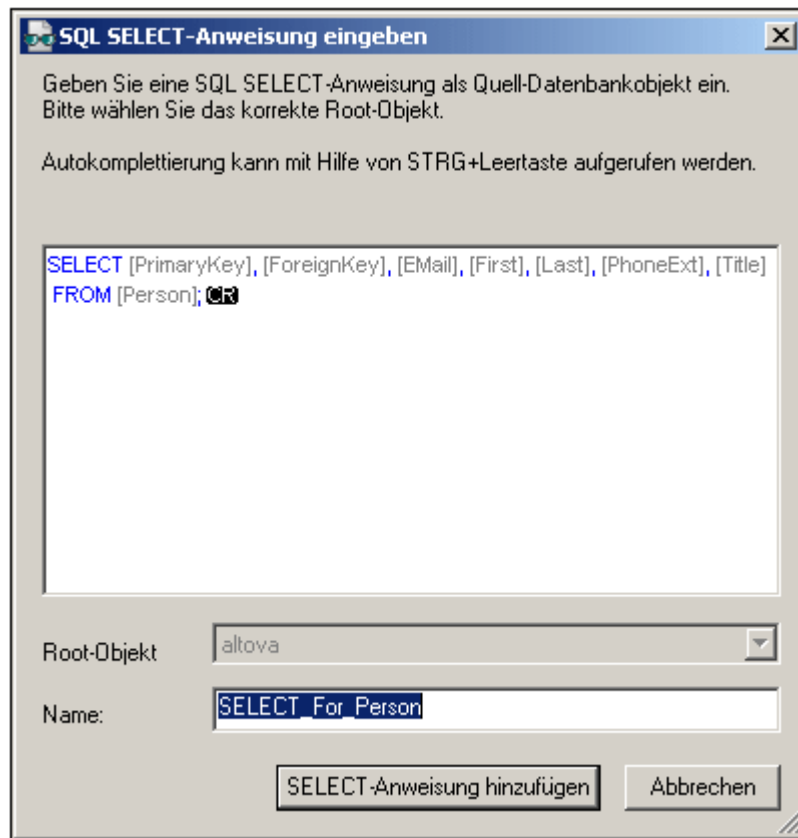
3. Klicken Sie auf das Objekt in der Liste, um es im Browser zu sehen.

Hinzufügen und Bearbeiten von SELECT-Anweisungen für lokale Ansichten

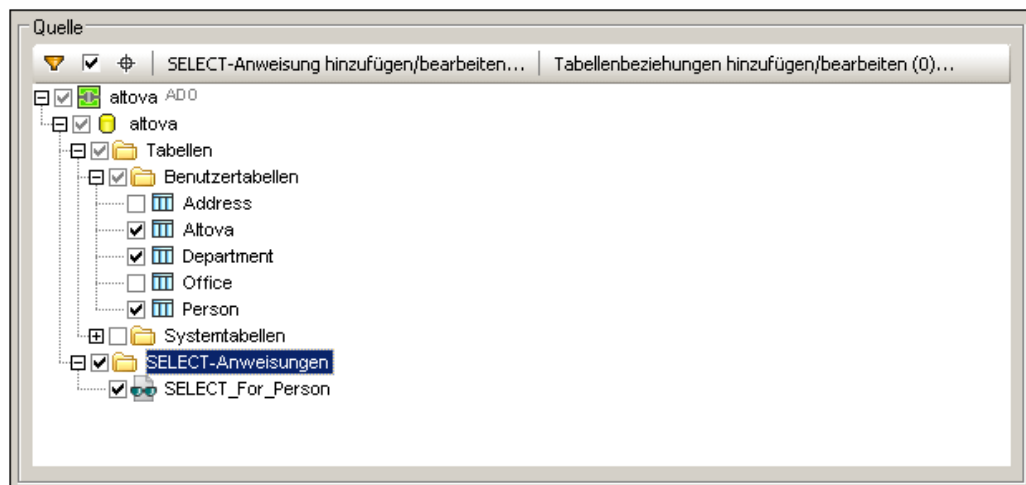
Sie können zur Erstellung lokaler Ansichten `SELECT`-Anweisungen in SQL erstellen. Wenn das Schema anhand einer Datenbankverbindung generiert wird, für die lokale Ansichten (oder `SELECT`-Anweisungen) definiert sind, enthält das Schema, das für die Datenbank generiert wird, eine Tabelle für jede `SELECT`-Anweisung.

Um eine `SELECT`-Anweisung zu erstellen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf das Register **SELECT-Anweisung hinzufügen/bearbeiten**. Daraufhin wird das Dialogfeld "SQL Select-Anweisung eingeben" angezeigt (*Abbildung unten*).



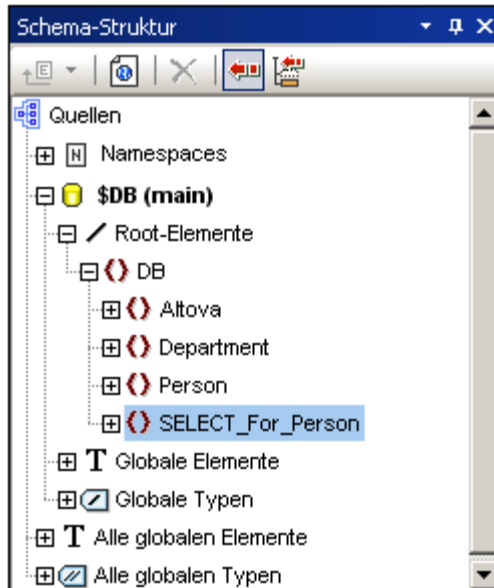
2. Geben Sie die `SELECT`-Anweisung ein. Wenn Sie eine Verbindung zu einer Oracle oder IBM DB2-Datenbank über JDBC herstellen möchten, darf die `SELECT`-Anweisung kein Semikolon am Schluss enthalten. Wenn Sie eine `SELECT`-Anweisung für eine ganze Tabelle erstellen wollen, klicken Sie mit der rechten Maustaste im Dialogfeld "Datenbankobjekte einfügen" auf die Tabelle und wählen Sie den Menübefehl **SELECT-Anweisung erzeugen und hinzufügen**.
3. Klicken Sie auf **Select-Anweisung hinzufügen**. Die `SELECT`-Anweisung wird zur Liste der `SELECT`-Anweisungen im Dialogfeld "Datenbankobjekte einfügen" hinzugefügt (Abbildung unten).



Anmerkung: Wenn Sie eine Verbindung zu einer Oracle oder IBM DB2-Datenbank über

JDBC herstellen möchten und für den Datenabruf eine `SELECT`-Anweisung mit dem Befehl "Tabelle hinzufügen/entfernen" verwenden möchten, darf die `SELECT`-Anweisung kein Semikolon am Schluss enthalten.

Wenn Sie in diesem Dialogfeld auf **Fertig stellen** klicken, wird für jede `SELECT`-Anweisung eine Tabelle erstellt (*Abbildung unten*).

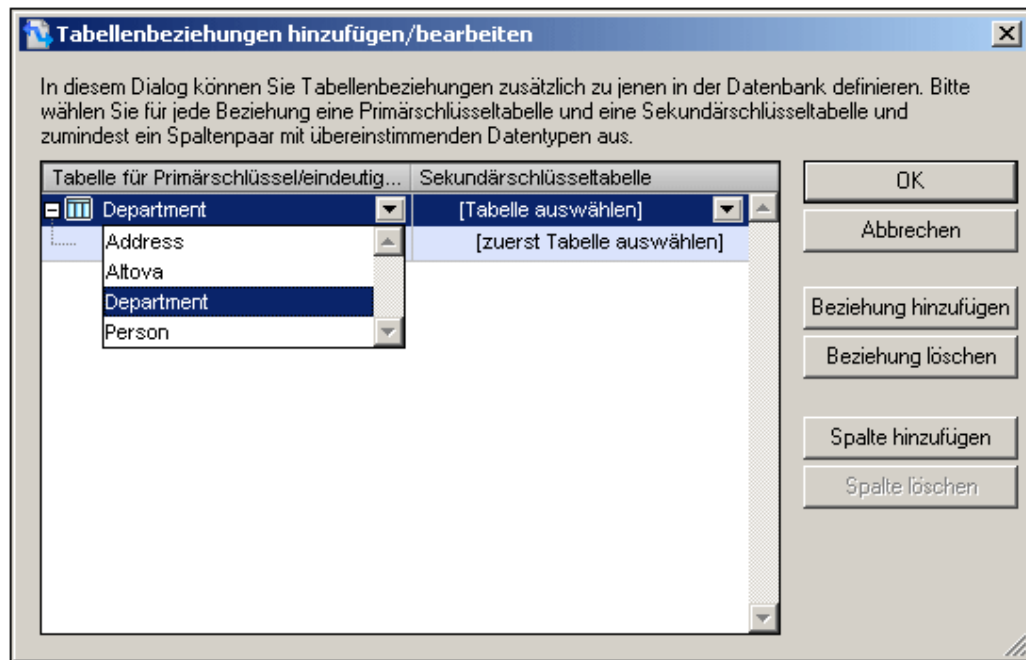


Lokale Beziehungen zwischen Tabellen

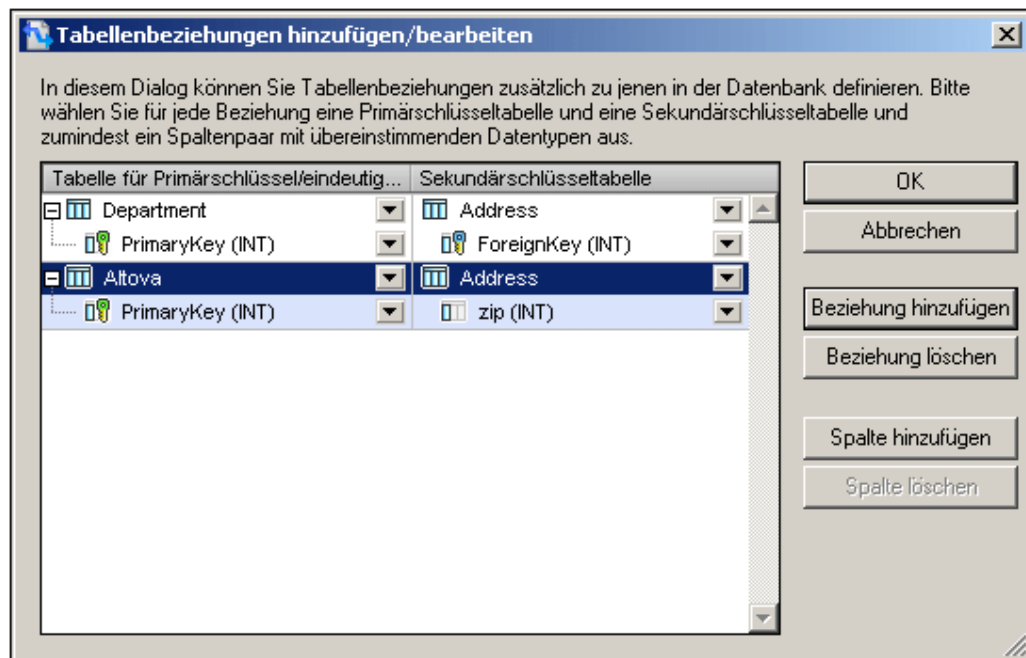
Ähnlich wie bei Primär- und Sekundärschlüsselbeziehungen können Sie lokale Beziehungen erstellen. Die Beziehung ist in StyleVision lokal, d.h. die Datenbank selbst muss nicht geändert werden. Die lokale Beziehung wird im generierten Schema dargestellt.

Um eine lokale Beziehung zu erstellen, gehen Sie folgendermaßen vor:

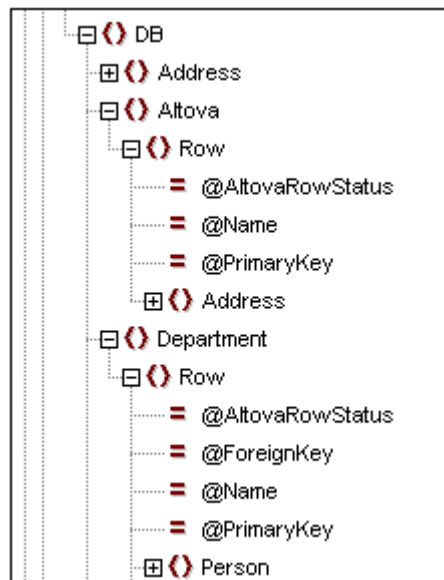
1. Klicken Sie im Dialogfeld "Datenbankobjekte einfügen" auf das Register **Lokale Beziehungen**. Daraufhin wird das Dialogfeld "Tabellenbeziehungen hinzufügen" angezeigt (*siehe Abbildung unten*).
2. Klicken Sie auf die Schaltfläche **Beziehung hinzufügen**, und klicken Sie in der Spalte "Master" auf die Dropdown-Schaltfläche der Auswahlliste "Select target table" (*Abbildung unten*). Wählen Sie auch für die Spalte "Child" eine Tabelle aus. Die Beziehung, die schließlich generiert wird, wählt Master-Zeilen aus, in denen die ausgewählte Master-Spalte mit der ausgewählten Child-Spalte übereinstimmt.



3. Wählen Sie die Master-Tabellenspalte aus, die mit der Child-Tabellenspalte übereinstimmen muss und wählen Sie anschließend die Child-Spalte aus. Verwenden Sie auch hier wieder die Auswahllisten in den entsprechenden Spalten (*siehe Abbildung unten*). Beachten Sie: Wenn die Typen nicht übereinstimmen, wird eine Fehlermeldung angezeigt.
4. Fügen Sie ggf. weitere lokale Beziehungen hinzu, indem Sie Schritt 2 und 3 wiederholen.



5. Klicken Sie auf **OK**, um die Beziehung fertig zu stellen. Wenn das Schema generiert wird, werden darin die neu erstellten Beziehungen berücksichtigt.



Sehen Sie sich die beiden Abbildungen oben an. Sie sehen darin, wie im generierten Schema innerhalb von `Person`, `Department` und `Altova` angelegt wurden. Diejenigen `Department` Zeilen, in denen der Wert von `ForeignKey` gleich dem Wert von `PrimaryKey` der übergeordneten Zeile `Person` ist, werden ausgegeben. Und die `Altova`-Zeilen, deren `Name` Werte mit dem `Last` Wert der übergeordneten Zeile `Person` übereinstimmen, werden ausgegeben.

Anmerkung: Lokale Beziehungen zwischen SQL `SELECT` Anweisungen werden nicht unterstützt.

Vorschauen und Vorschauenfenster

Um die Struktur und den Inhalt der Tabelle in einer Vorschau zu sehen, wählen Sie die Tabellen im Bereich "Quelle" aus und klicken Sie anschließend im Vorschaubereich auf die Schaltfläche **Neu laden**. (Abbildung unten). Der Inhalt der Tabelle wird im Tabellenformat im Vorschaubereich angezeigt (Abbildung unten).

Vorschau

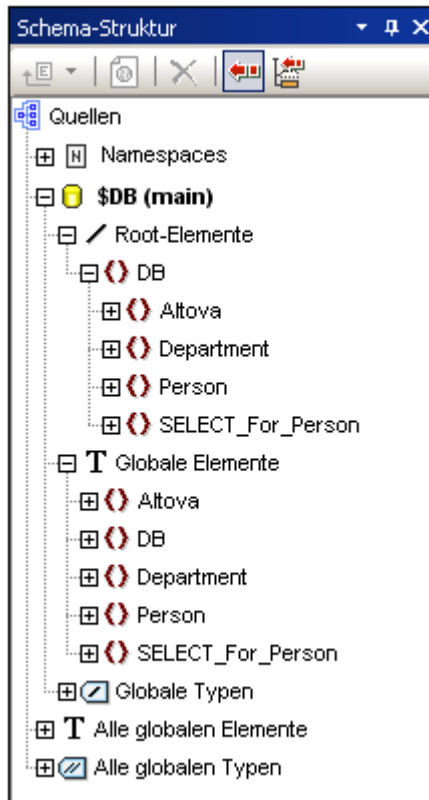
Neu laden Table: Person

PrimaryKey	ForeignKey	E-Mail	First	Last
1	1	v.callaby@nanonull.com	Vernon	Callaby
2	1	f.further@nanonull.com	Frank	Further
3	1	l.matise@nanonull.com	Loby	Matise
4	2	j.firstbread@nanonull.com	Joe	Firstbread
5	2	s.sanna@nanonull.com	Susi	Sanna
6	3	f.landis@nanonull.com	Fred	Landis

Generieren des XML-Schemas und der XML-Arbeitsdatei anhand der Datenbank

- Klicken Sie nach Auswahl der Tabellen für das SPS auf **Fertig stellen**, um das XML-Schema zu generieren und zu laden. Im Schemafenster wird ein XML-Schema mit einer Struktur angezeigt, die der Datenbank mit den ausgewählten Tabellen entspricht. Außerdem wird eine XML-Arbeitsdatei erstellt und in der Ausgabevorschau

angezeigt, deren Struktur der des generierten Schemas entspricht und die Daten aus den ausgewählten Tabellen enthält.



Beachten Sie, dass alle ausgewählten Datenbanktabellen im XML-Schema als Children des Dokumentelements `DB` erstellt und als Einträge in der Liste der globalen Elemente angezeigt werden. Eine ausführliche Beschreibung der Struktur des generierten XML-Schemas finden Sie unter [Das DB-Schema und DB-XML-Dateien](#). Beachten Sie, dass das anhand der Datenbank generierte XML-Schema von keinem später erstellten DB-Filter geändert wird.

Nachdem Sie die Verbindung zur Datenbank hergestellt und das XML-Schema generiert haben, steht Ihnen die ganze Palette der StyleVision Funktionen zum Erstellen eines SPS für die Datenbank zur Verfügung.

XML-Datenbanken

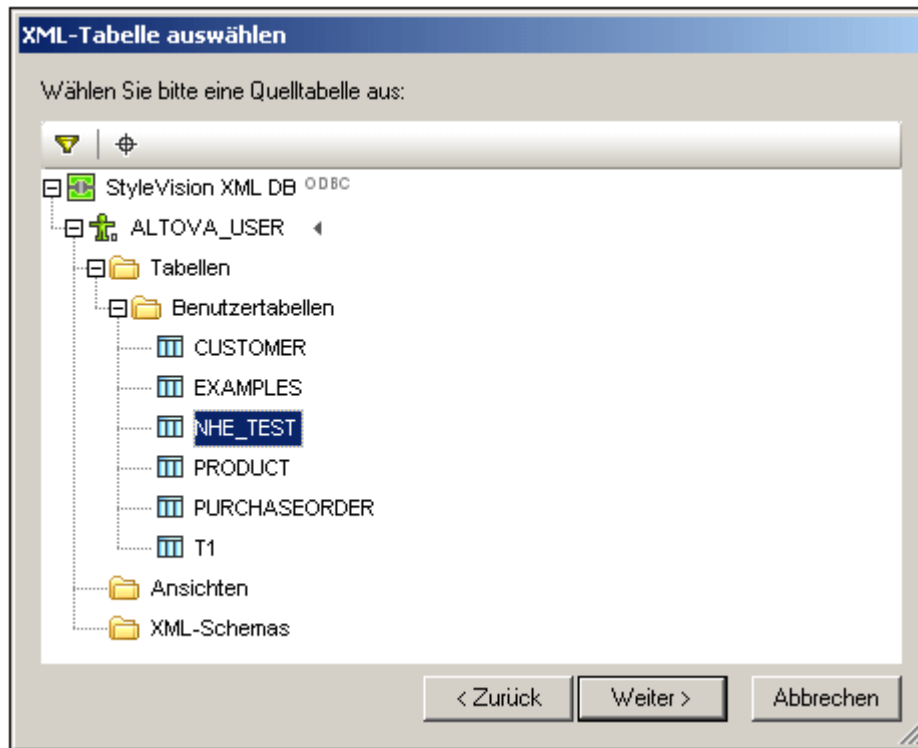
Nachdem Sie über das Dialogfeld "Datenbank öffnen" (siehe vorhergehende Abschnitte) eine Verbindung zur XML-Datenbank (nur IBM DB2-XML-Datenbanken werden unterstützt) hergestellt haben, müssen Sie zwei Dinge tun:

- Auswahl der Zelle in der DB, die das gewünschte XML-Dokument enthält. Das XML-Dokument wird automatisch als XML-Arbeitsdatei geladen.
- Auswahl des XML-Schemas, das als Basis für das SPS generiert werden soll.

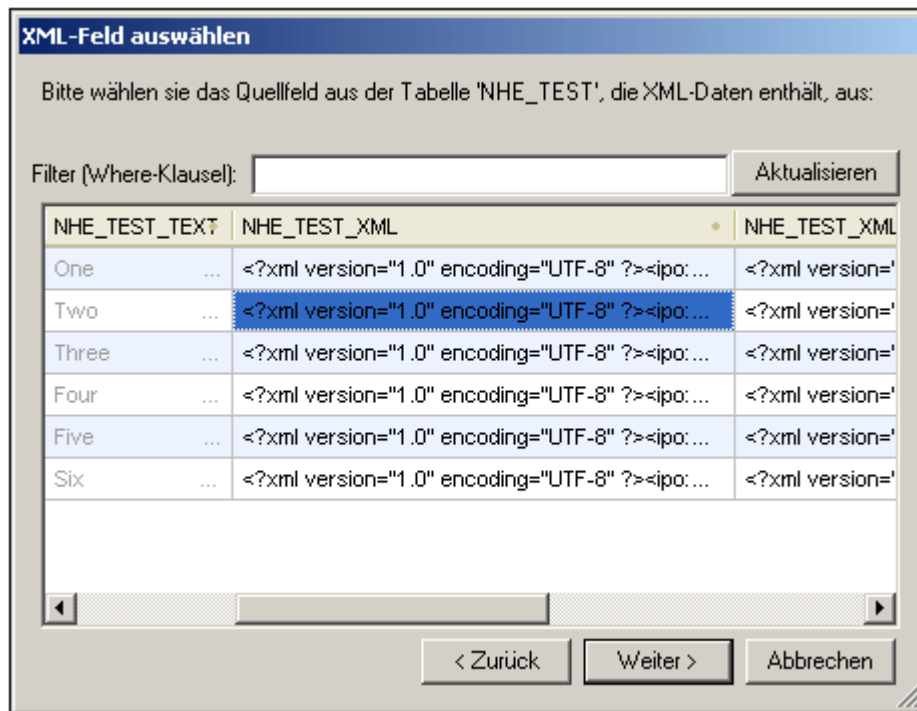
Auswahl der XML-Zelle und der XML-Arbeitsdatei

Nach Herstellung der Verbindung zur IBM DB2-Datenbank wird das Dialogfeld "XML-Tabelle auswählen" (*Abbildung unten*) angezeigt.

1. Wählen Sie im Dialogfeld "XML-Tabelle auswählen" die Tabelle aus, die die XML-Daten enthält, die als XML-Arbeitsdatei erstellt werden sollen. In der Abbildung unten wurde die Tabelle `NHE_TEST` ausgewählt.



2. Klicken Sie auf **Weiter**. Daraufhin erscheint das Dialogfeld "XML-Zelle auswählen" (*Abbildung unten*). Wenn Sie die im Fenster angezeigte Auswahl filtern möchten, geben Sie eine SQL `WHERE`-Klausel ein und klicken Sie auf **Aktualisieren**. Beachten Sie: Die `WHERE`-Klausel sollte nur die Bedingung sein (ohne das Schlüsselwort `WHERE`, z.B.: `NHE_TEST_TEXT= 'Two'`)



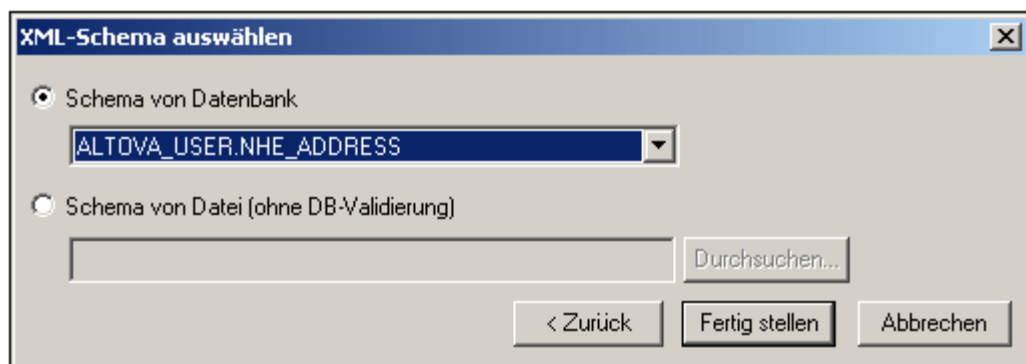
3. Wählen Sie die Zelle aus, die die XML-Daten enthält, die als XML-Arbeitsdatei erstellt werden sollen. In der Abbildung oben ist die ausgewählte Zelle blau markiert.
4. Klicken Sie auf **Weiter**. Daraufhin erscheint das Dialogfeld "XML-Schema auswählen". Hier können Sie das Schema für das SPS auswählen. *Siehe nächster Abschnitt.*

Anmerkung: Wenn Sie eine Verbindung zu einer Oracle oder IBM DB2-Datenbank über JDBC herstellen möchten und für den Datenabruf eine `SELECT`-Anweisung mit dem Befehl "Tabelle hinzufügen/entfernen" verwenden möchten, darf die `SELECT`-Anweisung kein Semikolon am Schluss enthalten.

Auswahl des XML-Schemas für das SPS

Das für das SPS verwendete Schema kann entweder ein in der DB enthaltenes XML-Schema sein oder ein Schema unter einem Dateipfad, auf den StyleVision Zugriff hat. So wählen Sie das Schema aus:

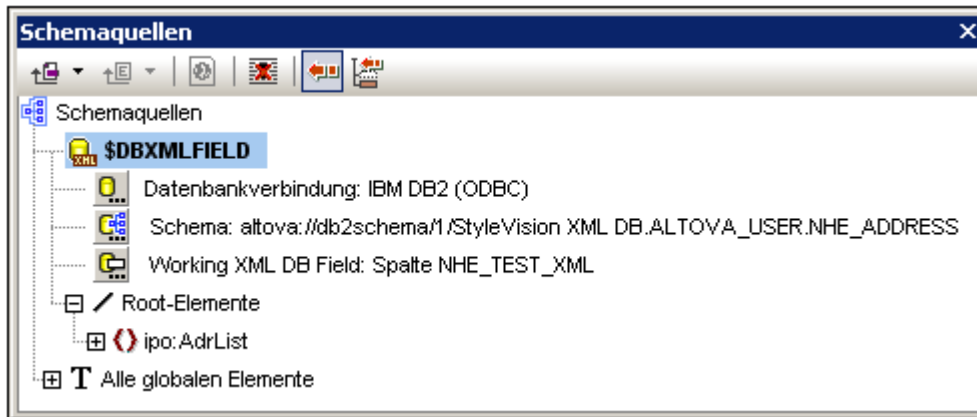
1. Aktivieren Sie im Dialogfeld "XML-Schema auswählen" (Abbildung unten) das entsprechende Optionsfeld, je nachdem, ob Sie das Schema aus den in der DB gespeicherten Schemas oder aus einem Ordner auswählen möchten. Beachten Sie: Bei Auswahl eines nicht-DB-Schemas - d.h. wenn das Schema aus einer externen Datei stammt - wird keine DB-Validierung durchgeführt.



2. Wählen Sie das Schema aus. In der Datenbank gespeicherte Schemas werden in der Schema-Dropdown-Liste der Auswahlliste "Datenbank" aufgelistet und können hier ausgewählt werden. Ein externes Schema wird mit Hilfe der Schaltfläche "Durchsuchen" ausgewählt.
3. Klicken Sie auf **Fertig stellen**.

Die Schema-Struktur

Nach Auswahl der XML-Daten und des Schemas werden die ausgewählten XML-Daten als XML-Arbeitsdatei erstellt und das Schema wird in das SPS geladen. Beide werden im Fenster "Schema-Struktur" angezeigt (*Abbildung unten*).



Das SPS kann nun mit Hilfe der üblichen StyleVision Funktionen erstellt werden. Beachten Sie: Die Daten in der XML-Arbeitsdatei können in der Authentic-Ansicht bearbeitet und in der DB gespeichert werden.

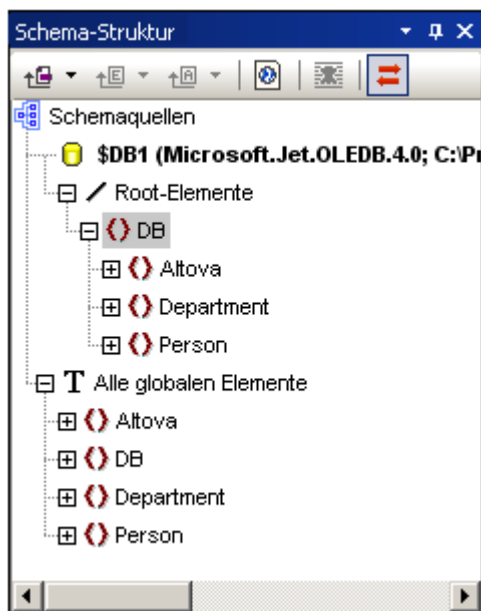
Anmerkung: Die XML-Arbeitsdatei sollte gemäß dem für das SPS ausgewählten Schema gültig sein. Stellen Sie außerdem sicher, dass das [Root-Element \(Dokumentelement\)](#) des Schemas mit dem Root-Element des XML-Dokuments übereinstimmt.

13.4 Das DB-Schema und DB-XML-Dateien

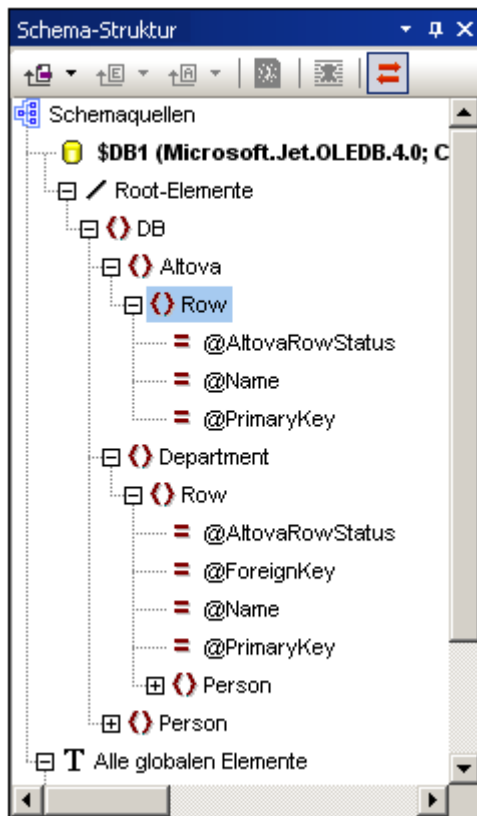
Das XML-DB-Schema

Beim Laden einer nicht-XML-Datenbank (nicht-XML-DB) in StyleVision wird ein XML-Schema mit einer Struktur geladen, die auf der Datenbank basiert, und im Fenster "Schema-Struktur" angezeigt. (Im Fall von XML-Datenbanken wird ein vorhandenes Schema (entweder ein in der DB oder ein unter einem Dateipfad gespeichertes Schema) als das Schema definiert, das im SPS verwendet werden soll.) Der Abschnitt über Schemas bezieht sich daher nur auf nicht-XML-Datenbanken.

Das XML-Schema wird mit einem Dokumentelement namens `DB` erstellt. Das Element `DB` enthält Child-Elemente, die der obersten Ebene der Tabellen in der Datenbank entsprechen. Diese Tabellenelemente der obersten Ebene werden im Schema-Fenster auch als Einträge in der Liste der globalen Vorlagen erstellt. Die Elemente der obersten Ebene in der Abbildung unten sind: `Altova`, `Department` und `Person`; Sie entsprechen Tabellen in der Datenbank.



Jedes Tabellenelement der obersten Ebene hat eine unbegrenzte Anzahl an Zeilen (*siehe Abbildung unten*). Jede Zeile entspricht einem Datensatz in der Datenbank. In der Schemastruktur werden die Zeilen durch ein einziges `Row`-Element dargestellt. Jedes `Row`-Element hat Attribute, die den Feldern der Tabelle entsprechen. Eines dieser Attribute wird von StyleVision für jede einzelne Zeile jeder Tabelle erstellt: `AltovaRowStatus`, welches den aktuellen Status enthält: hinzugefügt, aktualisiert und/oder gelöscht. Die restliche Attribute sind die Felder der jeweiligen DB-Tabelle.



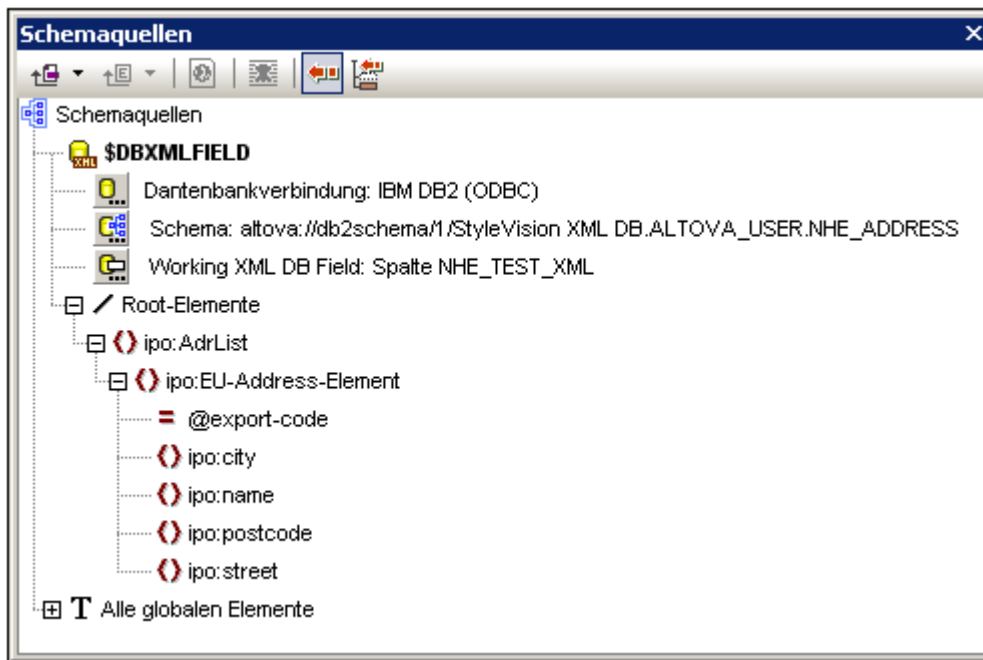
Anmerkung: Die Struktur des generierten XML-Schemas wird oben gezeigt. Alle Tabellen, die beim Herstellen der Verbindung mit der DB ausgewählt wurden, sind in der Struktur enthalten. Die Erstellung eines DB-Filters wirkt sich nicht auf die Struktur des XML-Schemas aus.

Neue DB-Schema-Struktur

Die Struktur des XML-Schemas, das mit StyleVision ab Version 2005 anhand von Datenbanken erstellt wird unterscheidet sich von der Struktur, die mit früheren Versionen von StyleVision erstellt wurde. Mit der neuen Struktur wird die Bearbeitung von Datenbanken in der Authentic-Ansicht von Altova-Produkten möglich, ein Feature, das in früheren Versionen nicht zur Verfügung stand. Aus diesem Grund erzeugt jedes SPS, das mit früheren Versionen von StyleVision generiert wurde, einen Fehler, wenn es in einer StyleVision Version ab Version 2005 geöffnet wird. Um die Datenbankbearbeitungs- und -berichtsfunktionen von StyleVision verwenden zu können, sollten Sie das SPS in der aktuellen Version von StyleVision neu erstellen.

XML-DB-Datendateien

Nachdem Sie eine Verbindung zur XML-Datenbank hergestellt und das XML-Schema und die Spalte mit den XML-Daten ausgewählt haben, werden das ausgewählte Schema und die für die XML-Arbeitsdatei verwendete Spalte im Fenster "Schema-Struktur" (*Abbildung unten*) angezeigt.



Anhand der Datenbank werden zwei **temporäre** XML-Dateien generiert (genauer hierzu siehe [Datenbanken und StyleVision](#)):

- eine temporäre editierbare XML-Datei, die in der Authentic-Ansicht bearbeitet werden kann.
- eine temporäre nicht editierbare XML-Datei, die als XML-Arbeitsdatei verwendet wird (für Vorschauen und die Generierung der Ausgabedatei)

Die temporäre **editierbare XML-Datei** wird beim Laden der DB in StyleVision generiert. Sie kann nach der Erstellung des StyleVision Power Stylesheet in der Authentic-Ansicht bearbeitet werden. Die Anzeige in der Authentic-Ansicht kann durch Verwendung von Abfragemechanismen in der Authentic-Ansicht gefiltert werden. Jede Änderung, die an den editierbaren Daten in der Authentic-Ansicht vorgenommen wird, wird in die temporäre XML-Datei geschrieben. Wenn Sie auf **Datei | Authentic XML-Daten speichern...** klicken, werden die Informationen in der temporären editierbaren XML-Datei in der Datenbank gespeichert.

Die temporäre **nicht editierbare XML-Datei** wird beim Laden der Datenbank in StyleVision generiert. Sie wird als XML-Arbeitsdatei und zum Generieren der HTML- und RTF-Ausgabe verwendet. Die editierbare XML-Datei muss gespeichert werden, bevor Änderungen, die in der Authentic-Ansicht gemacht wurden, in einer Vorschau angezeigt werden können.

Anmerkung:

- In der Authentic-Ansicht anderer [Authentic-Ansicht](#) Produkte wird beim Öffnen eines DB-basierten SPS nur eine temporäre (editierbare) XML-Datei erstellt. Änderungen, die an der Authentic-Ansicht vorgenommen wurden, werden in diese Datei geschrieben. Beim Speichern der Datei werden die Informationen in der XML-Datei in die Datenbank geschrieben.
- Sie können die Daten, die in die nicht editierbare temporäre XML-Datei übernommen werden, zur Berichtserstellung filtern. (Nähere Informationen siehe [Bearbeiten des DB-Filters](#)).
- Sie müssen keine XML-Arbeitsdatei zuweisen, um die HTML- und RTF-Vorschau zu sehen. Zu diesem Zweck wird die automatisch generierte (nicht editierbare) temporäre XML-Datei verwendet.

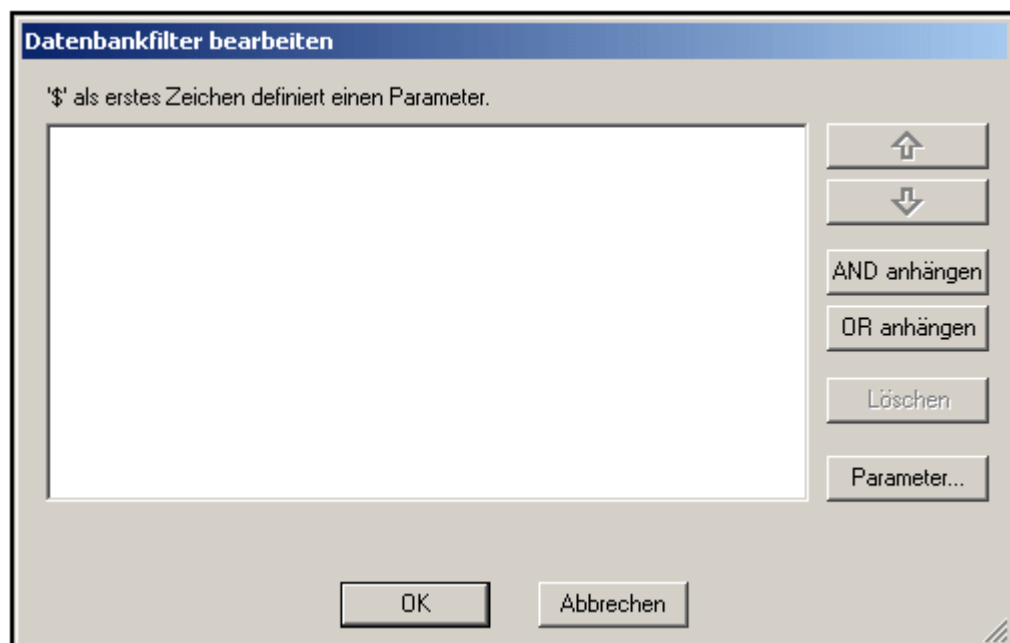
13.5 DB-Filter: Filtern von Datenbankdaten

Die Daten, die aus der Datenbank (DB) in die nicht editierbare XML-Datei importiert werden, können gefiltert werden. (Beachten Sie, dass die nicht editierbare XML-Datei für die Berichtsgenerierung verwendet wird und dass sich die Auswirkungen eines DB-Filters daher nur auf die HTML- und RTF-Vorschau auswirken, nicht aber auf die Authentic-Vorschau, in welcher die temporäre **editierbare** XML-Datei angezeigt wird, und nicht in der Authentic-Ansicht). Der DB-Filter kann entweder innerhalb der Datenbank selbst (falls dies von Ihrer DB-Applikation unterstützt wird) oder im SPS (SPS-Datei) erstellt werden. Im SPS kann für jede Datentabelle auf oberster Ebene im XML-Schema (d.h. für die Datentabellen, die die Child-Elemente des DB-Elements bilden) ein Filter erstellt werden. Jedes Mal, wenn ein DB-Filter erstellt oder geändert wird, werden die Daten aus der DB in die temporäre nicht editierbare XML-Datei, die für die DB generiert wird, neu geladen. Auf diese Art können Sie die XML-Datei mit Hilfe von DB-Filtern möglichst klein halten und die Berichtsgenerierung möglichst effizient gestalten.

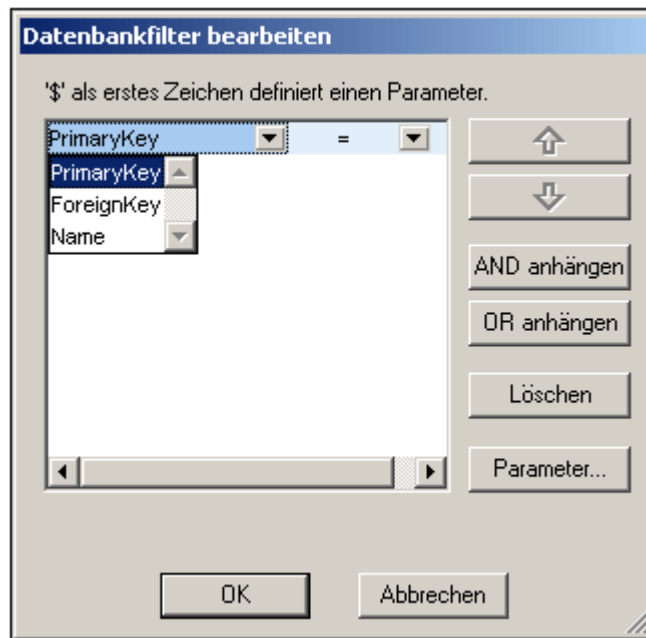
Anmerkung: Bei Verwendung eines DB-Filters werden die Daten, die in die nicht editierbare XML-Datei importiert werden, geändert. Wenn Sie ein SPS mit einem DB-Filter speichern und dann eine XML-Datei anhand des SPS generieren, wird die generierte XML-Datei entsprechend den Kriterien des DB-Filters gefiltert.

Erstellen eines DB-Filters

1. Wählen Sie im Design-Dokument oder in der Schemastruktur das Element der Datentabelle aus (entweder durch Klicken auf den Start- oder End-Tag des Elements oder durch Auswahl des Elements in der Schemastruktur), für das Sie einen DB-Filter erstellen möchten.
2. Wählen Sie [Datenbank | DB-Filter bearbeiten](#) oder klicken Sie in der Symbolleiste auf das Symbol . Daraufhin wird das folgende Dialogfeld angezeigt:



3. Klicken Sie auf eine der Schaltflächen **AND anhängen** oder **OR anhängen**. Daraufhin wird ein leeres Kriterium für den Filter angehängt (siehe Abbildung unten).



4. Geben Sie den Ausdruck für das Kriterium ein. Ein Ausdruck besteht aus: (i) einem Feldnamen (steht über die dazugehörige Auswahlliste zur Verfügung); (ii) einem Operator (verfügbar über die dazugehörige Auswahlliste); und (iii) einem Wert (der direkt eingegeben wird). Nähere Informationen dazu, wie Ausdrücke korrekt erstellt werden, finden Sie weiter unten unter [Ausdrücke in Kriterien](#).
5. Wenn Sie ein neues Kriterium hinzufügen möchten, klicken Sie auf die Schaltfläche **AND anhängen** oder **OR anhängen**, je nachdem, welchen logischen Operator (AND oder OR) Sie für die beiden Kriterien verwenden möchten. Fügen Sie anschließend das neue Kriterium hinzu. Nähere Informationen über logische Operatoren finden Sie im Abschnitt [Neuordnen von Kriterien in DB-Filtern](#).

Ausdrücke in Kriterien

Ausdrücke in DB-Filterkriterien bestehen aus einem Feldnamen, einem Operator und einem Wert. Die **verfügbaren Feldnamen** sind die Child-Elemente der auf der obersten Ebene ausgewählten Datentabelle; die Namen dieser Felder sind in einer Auswahlliste aufgelistet (siehe Abbildung oben). Die verfügbaren **Operatoren** sind unten aufgelistet:

=	ist gleich
<>	ist nicht gleich
<	kleiner als
<=	kleiner oder gleich
>	größer als
>=	größer oder gleich
LIKE	phonetisch gleich
NOT LIKE	phonetisch nicht gleich
IS NULL	ist leer
NOT NULL	ist nicht leer

Bei Auswahl von IS NULL oder NOT NULL ist das Wertefeld deaktiviert. Eingegebene **Werte** dürfen nicht in Anführungszeichen (oder andere Trennzeichen) gesetzt werden. Außerdem muss die Formatierung der Werte der im entsprechenden Feld der Datenbank entsprechen, da das Ergebnis des Ausdrucks andernfalls FALSE ergibt. Wenn z.B. ein Kriterium für ein Feld des Datentyps date in einer MS Access Datenbank den Ausdruck `StartDate=25/05/2004` hat, ist das Ergebnis des Ausdrucks FALSE, da der Datentyp date in einer MS Access-Datenbank

das Format `JJJJ-MM-TT` hat.

Verwendung von Parametern mit DB-Filtern

Sie können als Wert eines Ausdrucks auch den Namen eines Parameters eingeben. In diesem Fall wird der Parameter aufgerufen und sein Wert wird als Wert dieses Ausdrucks verwendet. Beim hier eingegebenen Parameter kann es sich um einen Parameter handeln, der bereits für das Stylesheet deklariert wurde, oder um einen Parameter, den Sie zu einem späteren Zeitpunkt deklarieren.

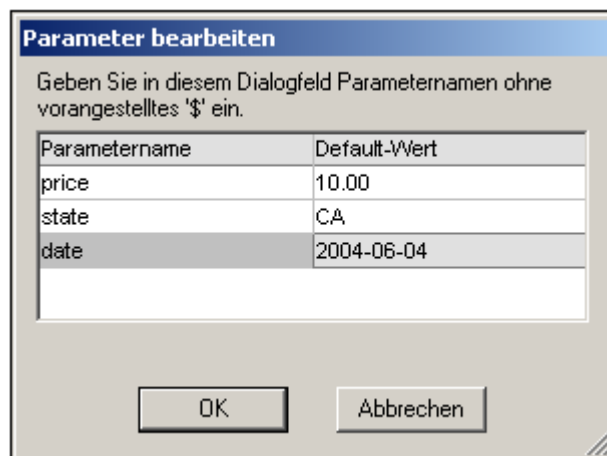
Parameter eignen sich, wenn Sie einen einzigen Wert in mehreren Ausdrücken verwenden möchten oder wenn Sie einen Wert über die Befehlszeile an einen Parameter übergeben möchten (nähere Informationen siehe [StyleVision über die Befehlszeile](#)).

Um den Namen eines Parameters als den Wert eines Ausdrucks einzugeben, geben Sie in das Werteingabefeld `$` ein, gefolgt vom Namen des Parameters (ohne Leerzeichen dazwischen). Wenn der Parameter bereits deklariert wurde (siehe [Parameter](#)), erscheint der Eintrag in Grün. Wurde der Parameter noch nicht deklariert, erscheint der Eintrag rot und Sie müssen den Parameter deklarieren.

Deklarierten von Parametern über das Dialogfeld "DB-Filter bearbeiten"

Um das Dialogfeld "Parameter bearbeiten" aufzurufen (um Parameter zu deklarieren), gehen Sie folgendermaßen vor:

1. Klicken Sie im Dialogfeld "DB-Filter bearbeiten" auf die Schaltfläche **Parameter...**. Daraufhin wird das unten gezeigte Dialogfeld "Parameter bearbeiten" aufgerufen.

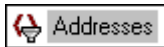


2. Geben Sie den Namen und Wert des Parameters in die entsprechenden Felder ein.

Alternativ dazu können Sie das Dialogfeld "Parameter bearbeiten" aufrufen und einen DB-Parameter durch Auswahl von **Bearbeiten | Stylesheet-Parameter bearbeiten** deklarieren oder bearbeiten.

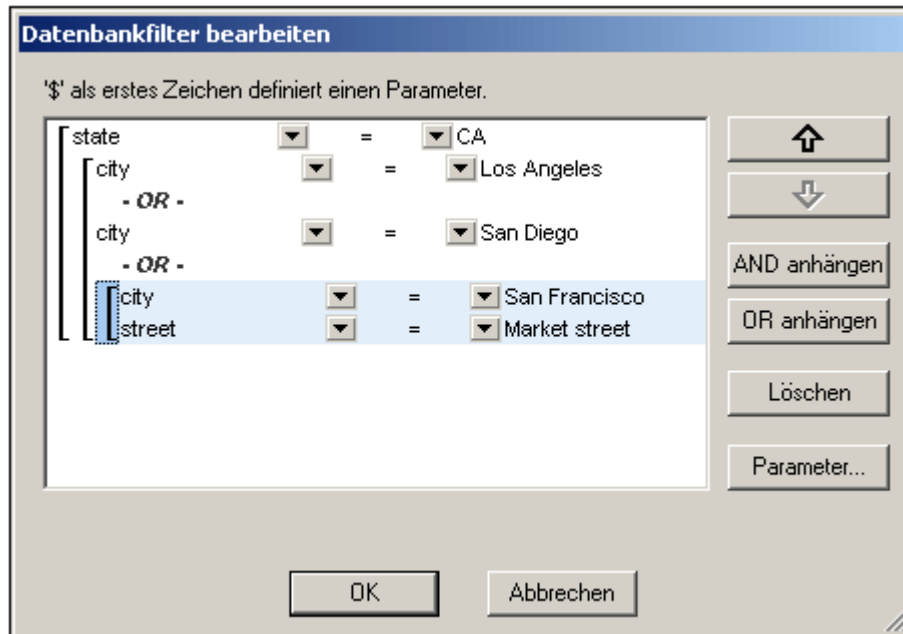
Anmerkung: Das Dialogfeld "Parameter bearbeiten" enthält alle Parameter, die für das Stylesheet definiert wurden. Wenn Sie einen undeklarierten Parameter im SPS verwenden, ergibt dies einen Fehler. Wenn Sie einen Parameter jedoch deklarieren und nicht verwenden, ist dies kein Fehler.

Nach Erstellung eines DB-Filters für ein Datentablenelement, wird dieses Element in der Schema-Struktur mit einem Filtersymbol angezeigt, wie das Element `Addresses` in der Abbildung unten.



Neureihung der Kriterien in DB-Filtern

Die logische Struktur des DB-Filters und die Beziehung zwischen zwei beliebigen Kriterien oder Gruppen von Kriterien wird grafisch angezeigt. Jede Ebene der logischen Struktur wird durch eine eckige Klammer gekennzeichnet. Wenn zwei Kriterien nebeneinander stehen, so wurde dafür der AND Operator verwendet. Wenn zwei Kriterien durch das Wort **OR** getrennt sind, so wird der OR Operator angezeigt. Die Kriterien sind außerdem eingerückt, um die Gliederung der logischen Struktur des DB-Filters übersichtlich darzustellen.



Der DB-Filter in der Abbildung oben kann in Textform folgendermaßen dargestellt werden:

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

Sie können den DB-Filter neu ordnen, indem Sie ein Kriterium oder eine Gruppe von Kriterien relativ zu den anderen Kriterien im DB-Filter nach oben oder nach unten verschieben. Um ein Kriterium oder eine Gruppe von Kriterien zu verschieben, gehen Sie folgendermaßen vor:

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

Beachten Sie bitte die folgenden Punkte:

- Wenn sich das danebenliegende Kriterium in der Richtung, in der das Kriterium verschoben wird, auf derselben Ebene befindet, tauschen die beiden Kriterien den Platz.
- Eine Gruppe von Kriterien (d.h. ein Kriterium innerhalb einer Klammer) ändert seine Position innerhalb derselben Ebene und wird nicht auf eine andere Ebene verschoben.
- Ein einzelnes Kriterium ändert seine Position innerhalb derselben Ebene. Wenn sich das danebenliegende Kriterium weiter außen/innen befindet (d.h. nicht auf derselben Ebene), wird das ausgewählte Kriterium **um immer jeweils eine Ebene** nach außen/innen verschoben.

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

Ändern eines DB-Filters

Um einen DB-Filter zu ändern, klicken Sie auf [Datenbank | DB-Filter bearbeiten](#). Daraufhin wird das Dialogfeld "DB-Filter bearbeiten" angezeigt. Sie können nun die Ausdrücke in jedem der aufgelisteten Kriterien bearbeiten, neue Kriterien hinzufügen, Kriterien neu ordnen oder Kriterien im DB-Filter löschen. Nachdem Sie die Änderungen fertiggestellt haben, klicken Sie auf OK. Die Daten aus der DB werden automatisch in StyleVision neu geladen, um den Änderungen im DB-Filter Rechnung zu tragen.

Löschen eines DB-Filters

Um einen DB-Filter zu löschen, wählen Sie das Element, für das der DB-Filter gelöscht werden soll, im Design-Fenster oder in der Schema-Struktur aus. (Es gibt für jedes Datentabellenelement auf oberster Ebene einen DB-Filter.) Klicken Sie anschließend auf [Datenbank | DB-Filter löschen](#). Der Filter wird daraufhin gelöscht und das Filtersymbol wird nicht mehr neben dem Namen des Elements in der Schema-Struktur angezeigt.

13.6 SPS Design Features für Datenbanken

Sie können ein DB-basiertes SPS genau wie ein Schema-basiertes SPS erstellen, d.h. durch Ziehen von Schema-Nodes aus dem Schemafenster in das Design-Dokument; durch Einfügen von statischem Inhalt direkt in das Design-Dokument; und durch Anwenden einer geeigneten Formatierung auf die verschiedenen Design-Komponenten. Die folgenden Punkte beziehen sich speziell auf DB-basierte SPSs.

Erstellen einer dynamischen Tabelle für eine DB-Tabelle

Um eine dynamische Tabelle für eine DB-Tabelle zu erstellen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Schema-Struktur die DB-Tabelle der obersten Ebene aus, die als dynamische Tabelle erstellt werden soll und ziehen Sie sie in das Design.
2. Wenn Sie die Maustaste loslassen, erstellen Sie die Tabelle als `Inhalt` und löschen Sie den Platzhalter `Inhalt`. Wenn die Funktion "DB-Funktion automatisch hinzufügen" aktiv ist, sieht das Design in etwa folgendermaßen aus:



3. Wählen Sie in der Schema-Struktur das Element `Row` der DB-Tabelle, die als dynamische Tabelle erstellt werden soll, aus.
4. Ziehen Sie es an eine Stelle innerhalb des Elements "Addresses".
5. Wählen Sie beim Loslassen der Maustaste den Befehl "Tabelle erstellen" aus dem Popup-Menü aus und wählen Sie die DB-Felder aus, die als Spalten der dynamischen Tabelle erstellt werden sollen. Die DB-Tabelle wird als dynamische Tabelle erstellt.

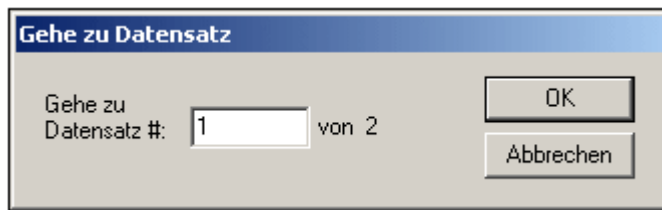
Anmerkung: Sie können eine DB-Tabelle auch in einem anderen Format, z.B. als (Inhalt) erstellen.

Automatisches Hinzufügen von DB-Steuerelementen

Mit dem Menübefehl **Authentic | DB-Steuerelemente automatisch hinzufügen** oder der Symbolleistenschaltfläche  können Sie die automatische Einfügung von Navigationselementen für DB-Tabellen ein- oder ausschalten. Wenn die Funktion aktiviert ist, hat diese Symbolleistenschaltfläche eine schwarze Umrandung. Wenn Sie deaktiviert ist, hat sie keine Umrandung. Wenn die Funktion aktiv ist, werden DB-Steuerelemente (*siehe Abbildung unten*) automatisch eingefügt, wenn eine DB-Tabelle in das Design-Dokument gezogen wird. Das Element wird standardmäßig vor dem Start-Tag `Row` eingefügt.



Mit Hilfe dieser Steuerelemente kann der Benutzer der Authentic-Ansicht zwischen den Datensätzen der DB-Tabelle in der Authentic-Ansicht navigieren. Über die erste (ganz linke) Schaltfläche gelangen Sie zum ersten Datensatz; über die zweite Schaltfläche gelangen Sie zum vorhergehenden Datensatz; die dritte Schaltfläche ist die "Gehe zu Datensatz" Schaltfläche. Wenn Sie darauf klicken, erscheint das Dialogfeld "Gehe zu Datensatz" (*Abbildung unten*). In diesem Dialogfeld werden Sie aufgefordert, die Nummer des gewünschten Datensatzes einzugeben. Über die vierte Schaltfläche gelangen Sie zum nächsten Datensatz und über die fünfte zum letzten Datensatz.



Um die Navigationselemente manuell in das Design-Dokument einzufügen - z.B. wenn Sie die Navigationselemente an einer anderen Stelle als der vorgegebenen einfügen möchten - gehen Sie folgendermaßen vor:

1. Deaktivieren Sie die Funktion "DB-Steuerelemente automatisch hinzufügen".
2. Platzieren Sie den Cursor an die Stelle, an der die Navigationselemente angezeigt werden sollen (jedoch innerhalb des Start- und End-Tags der DB-Tabelle).
3. Klicken Sie auf die rechte Maustaste und wählen Sie im Popup-Menü den Befehl **Einfügen | DB-Steuerelement | Navigation** oder **Einfügen | DB-Steuerelement | Navigation+Gehe zu**.

Einfügen einer DB-Abfrage-Schaltfläche für die Authentic-Ansicht

Mit Hilfe der Schaltfläche "DB-Abfrage"  kann der Benutzer der Authentic-Ansicht eine DB-Abfrage absenden. Er kann damit Bedingungen dafür erstellen, welche Datensätze in der Authentic-Ansicht angezeigt werden sollen. Sie können Abfrage-Schaltflächen für einzelne DB-Tabellen zwischen folgenden Tags einfügen:

- dem Start-Tag des DB-Tabellenelements und dem Start-Tag des Child-(Row)-Elements des DB-Tabellenelements.
- dem End-Tag des DB-Tabellenelements und dem End-Tag, des Child (Row)-Elements dieses Tabellenelements.

Um eine DB-Abfrage-Schaltfläche in Ihr Design-Dokument einzufügen, gehen Sie folgendermaßen vor:

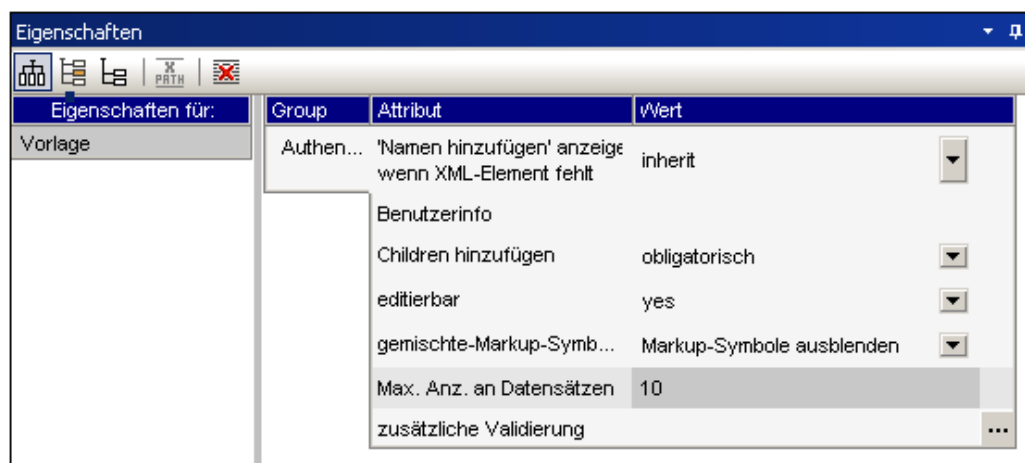
1. Platzieren Sie den Cursor an die Stelle (*siehe oben*), an der die Abfrage-Schaltfläche angezeigt werden soll. (in der Abbildung unten wird der Cursor zwischen den End-Row-Tag und den End-Customer-Tag platziert.
2. Klicken Sie auf die rechte Maustaste und wählen Sie im Kontextmenü (oder dem Menü **Einfügen) Einfügen | DB-Steuerelement | Abfrageschaltfläche**. Die Abfrageschaltfläche wird an dieser Stelle in das Design-Dokument eingefügt.

Wenn Sie in der Authentic-Ansicht darauf klicken, erscheint daraufhin das Dialogfeld "Datenbankabfrage bearbeiten", das in der Dokumentation zur [Authentic-Ansicht](#) beschrieben ist.

Festlegen der Anzahl der angezeigten Datensätze

Über die Einstellung "Max. Anz. an Datensätzen" in der Vorlage der Tabelle der obersten Ebene können Sie festlegen, wie viele Datensätze in der Authentic-Ansicht angezeigt werden sollen. Gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht das Element `Row` aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" in der Spalte "Eigenschaften für" `Vorlage` aus und die Eigenschaftsgruppe des Elements.



3. Geben Sie im Feld "Max. Anz. an Datensätzen" die maximale Anzahl an anzuzeigenden Datensätze an.

Anmerkung:

- Die Anzahl der anzuzeigenden Datensätze wird auf die Authentic-Ansicht angewendet.
- Wenn in der Authentic-Ansicht die maximale Anzahl an anzuzeigenden Datensätzen kleiner ist als die Gesamtanzahl der Datensätze in der DB, so wird die maximale Anzahl gleich zu Beginn in der Authentic-Ansicht geladen und angezeigt. Zusätzliche Datensätze werden über die Navigationsschaltflächen für die Datensätze in der Authentic-Ansicht angezeigt.
- Wenn in der Authentic-Ansicht eine große Anzahl an Tabellen geöffnet ist, wird eine Fehlermeldung angezeigt, dass zu viele Tabellen geöffnet sind. Sie können im Design die Anzahl der offenen Tabellen reduzieren, indem Sie die Anzahl der angezeigten Datensätze verringern (da ein Datensatz Tabellen enthalten kann). Auf der anderen Seite kann der Benutzer die Anzahl der in die Authentic-Ansicht geladenen Datensätze mit Hilfe von Abfragen einschränken.

AltovaRowStatus

In dem anhand der DB erstellten XML-Schema hat jede Tabelle ein **AltovaRowStatus**-Attribut. Der Wert dieses Attributs wird automatisch von StyleVision festgelegt und besteht aus drei Zeichen, die mit --- initialisiert sind. Wenn eine Zeile geändert oder eine neue Zeile hinzugefügt wird, wird der Wert mittels der folgenden Zeichen geändert.

A	Die Zeile wurde hinzugefügt (aber noch nicht in der Datenbank gespeichert).
U, u	Die Zeile wurde aktualisiert (aber noch nicht in der Datenbank gespeichert).
D, d, X	Die Reihe wurde gelöscht (aber noch nicht in der Datenbank gespeichert).

Diese Werte enthalte also Informationen für den Benutzer, welche Zeilen bearbeitet wurden. Die Statusinformationen bestehen bis zu dem Zeitpunkt, zu dem die Datei gespeichert wird. Danach werden die Statusinformationen wieder initialisiert (Anzeige von ---).

Formatieren von Komponenten des Design-Dokuments

Wenn Datensätze hinzugefügt, geändert oder gelöscht werden, formatiert StyleVision die hinzugefügten/geänderten/gelöschten Datensätze auf bestimmte Art, sodass sie Benutzer von anderen Datensätze unterscheiden können. Auch Datentyp-Fehler werden gekennzeichnet, indem sie rot angezeigt werden. Wenn Sie diese Markierungen beibehalten möchten, stellen Sie sicher, dass die Formatierung, die Sie Zeilen in einer Tabelle zuweisen, nicht dieselben Eigenschaften hat, wie die von StyleVision zugewiesenen. In StyleVision werden die folgende Formatierungen zugewiesen:

Hinzugefügt	A	Fett, unterstrichen
Geändert (aktualisiert)	U, u	Unterstrichen
Gelöscht	D, d, X	Durchgestrichen
Datentypfehler		Roter Text

13.7 Generieren von Ausgabedateien

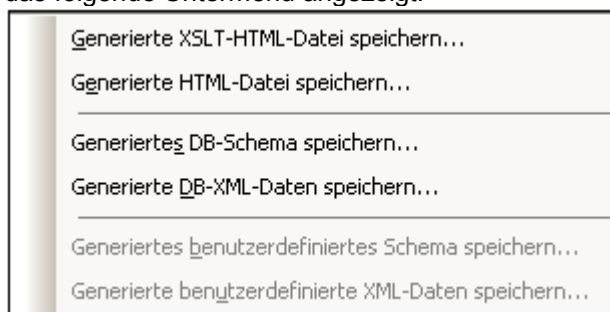
Nachdem Sie ein Datenbank-basiertes SPS generiert haben, können Sie damit in Zusammenhang stehende Dateien generieren und speichern. Es können die folgenden Dateien generiert und gespeichert werden:

- das XML-Schema auf Basis der DB-Struktur
- die XML-Datei, die die Struktur des generierten XML-Schemas und den Inhalt aus der Datenbank hat
- die XSLT-Datei für die HTML-Ausgabe
- die HTML-Ausgabedatei
- die XSLT-Datei für die RTF-Ausgabe
- die RTF-Ausgabedatei

Die Dateien können über die Benutzeroberfläche oder die Befehlszeile generiert und gespeichert werden.

Generierung über die Benutzeroberfläche von StyleVision

1. Wählen Sie im Menü **Datei** den Befehl **Generierte Dateien speichern**. Daraufhin wird das folgende Untermenü angezeigt.



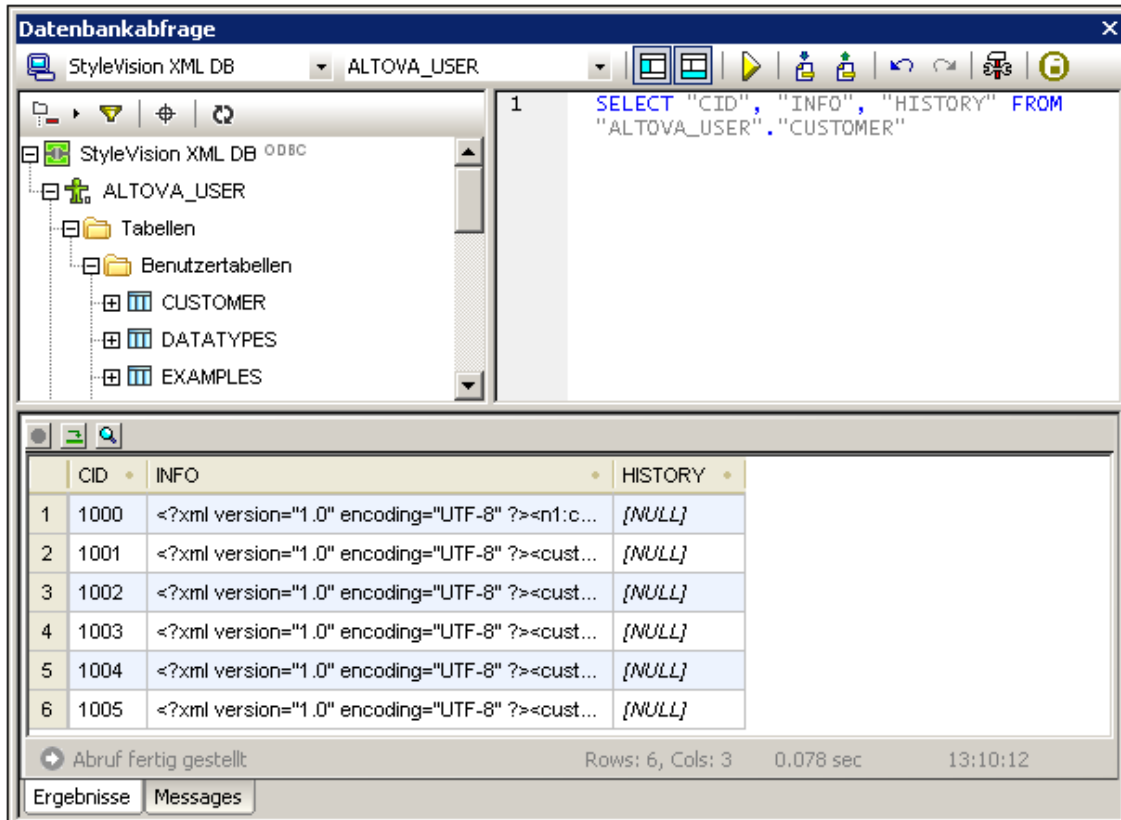
2. Wählen Sie die gewünschte Datei aus. Daraufhin erscheint das Dialogfeld "Speichern unter".
3. Suchen Sie den gewünschten Ordner, geben Sie den gewünschten Dateinamen ein und klicken Sie auf **OK**.

Generierung über die Befehlszeile

Sie können StyleVision über die Befehlszeile aufrufen und damit die Dateien generieren und speichern, die mit einem DB-basierten SPS verknüpft sind. Sie können nicht nur das XML-Schema und die XSLT-Datei speichern, sondern auch eine XML-Datei mit Daten aus der DB sowie eine HTML- und RTF-Ausgabedatei, die auf dem Design im SPS basieren. Eine Beschreibung zur Verwendung der Befehlszeile finden Sie unter [StyleVision über die Befehlszeile](#).

13.8 Datenbank abfragen

Mit dem Befehl **Datenbank abfragen** im Menü **Datenbank** wird das Fenster "Datenbankabfrage" (Abbildung unten) geöffnet. Sobald dieses Fenster geöffnet wurde, können Sie es durch Auswahl des Befehls **Datenbank | Datenbank abfragen** bzw. durch Klicken auf die Symbolleiste-Schaltfläche "Datenbank abfragen"  ein- und ausblenden.



Übersicht über das Fenster "Datenbankabfrage"

Das Fenster "Datenbankabfrage" besteht aus drei Teilen:

- einem [Browser-Fenster](#) links oben, in dem Verbindungsinformationen und Datenbanktabellen angezeigt werden.
- einem [Abfragefenster](#) rechts oben, in das die Abfrage eingegeben wird.
- einem mit Registern versehenen [Ergebnis-/Meldungsfenster](#). Im Fenster "Ergebnisse" werden die Ergebnisabfrage im sogenannten Ergebnisraster angezeigt. Im Fenster "Meldungen" werden die Meldungen (einschl. Fehler- und Warnmeldungen), die bei der Ausführung der DB-Abfrage generiert werden, angezeigt.

Das Fenster "Datenbankabfrage" verfügt im oberen Bereich über eine Symbolleiste. Zwei der Schaltflächen in dieser Symbolleiste sind im Folgenden beschrieben. Die restlichen Schaltflächen werden im Abschnitt [Fenster "Abfrage": Beschreibung und Funktionen](#) beschrieben.



Blendet das Browser-Fenster ein bzw. aus.



Blendet das Ergebnis-/Meldungsfenster ein bzw. aus.

Übersicht über die Mechanismen zur Abfrage von Datenbanken

Die Vorgangsweise bei der Abfrage von Datenbanken ist die folgende. Sie wird in den Unterabschnitten dieses Abschnitts näher beschrieben.

1. Über das Fenster "Datenbankabfrage" wird eine [Verbindung zur Datenbank hergestellt](#). Unterstützt werden die Datenbanken: MS Access 2000 und 2003; Microsoft SQL Server; Oracle; MySQL; Sybase; und IBM DB2.
2. Die Datenbank, zu der eine Verbindung hergestellt wurde, bzw. Teile dieser Datenbank werden im [Browser-Fenster](#) angezeigt. Dieses Fenster kann je nach gewünschter Ansicht angepasst werden.
3. Im [Abfragefenster](#) wird eine [Abfrage](#) gemäß den für die gewünschte Datenbank benötigten Syntaxregeln eingegeben und ausgeführt.
4. Die [Ergebnisse der Abfrage](#) können mit verschiedenen Filtern gefiltert, bearbeitet und wieder in der Datenbank gespeichert werden.

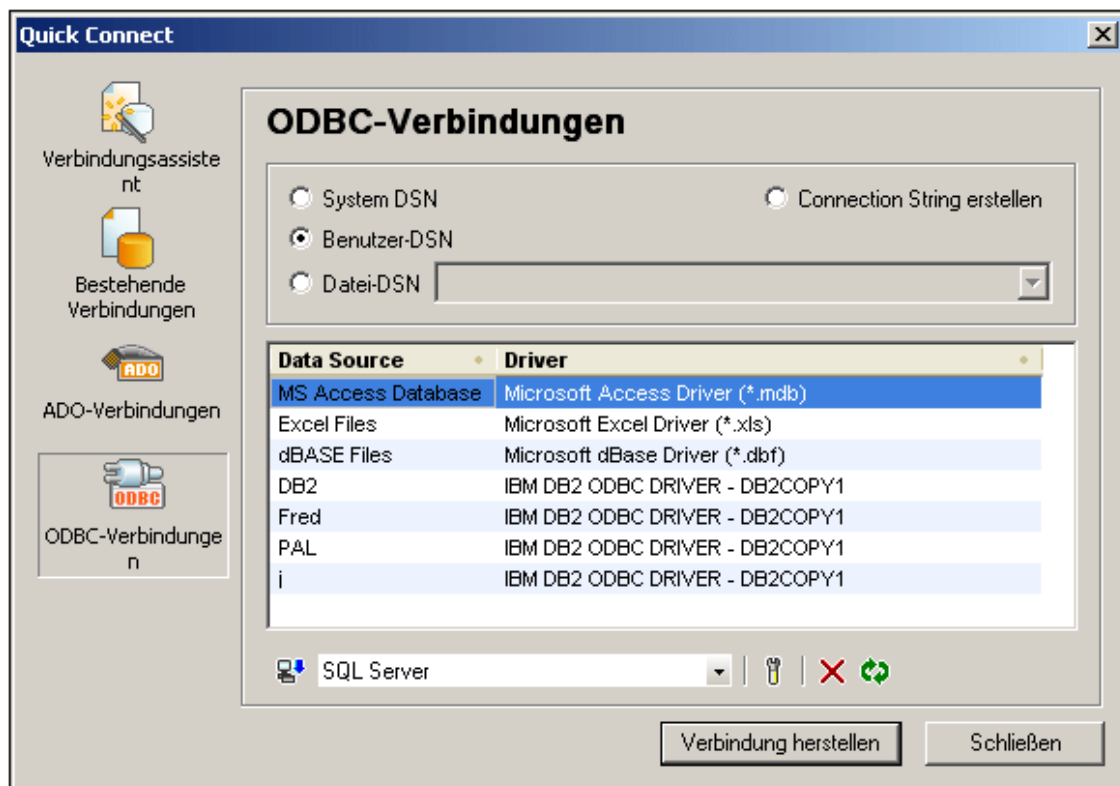
Datenquellen

Zur Ausführung einer Datenbankabfrage müssen Sie zuerst eine Verbindung zur gewünschten Datenbank herstellen. In diesem Abschnitt wird beschrieben, wie Sie:

- eine Verbindung zu einer Datenbank herstellen und
- die gewünschte Datenquelle und das Root-Objekt aus mehreren bestehenden Verbindungen auswählen.

Herstellen einer Verbindung zu einer Datenbank

Wenn Sie zum ersten Mal in einer Sitzung (oder wenn keine Datenbankverbindung besteht) im Menü **Ansicht** auf den Befehl **Datenbank abfragen** klicken, wird das Dialogfeld "Verbindungsassistent" (*Abbildung unten*) aufgerufen, über das Sie eine Verbindung zu einer Datenbank herstellen können. Um später weitere Verbindungen herzustellen, klicken Sie im Fenster "Datenbankabfrage" auf die Schaltfläche "Verbindungsassistent" . Wenn bereits Verbindungen bestehen, können Sie die gewünschte aus der Liste [auswählen](#).



Eine Beschreibung zum Herstellen einer Verbindung zu einer Datenbank über das Dialogfeld "Verbindungsassistent" finden Sie im Abschnitt [Herstellen einer Verbindung zu einer Datenquelle](#).

Altova StyleVision bietet vollständige Unterstützung für die unten aufgelisteten Datenbanken. Altova ist bestrebt, auch Unterstützung für andere ODBC/ADO-Datenbanken zu gewährleisten, doch wurde die erfolgreiche Herstellung der Verbindung und Verarbeitung der Daten nur bei den aufgelisteten Datenbanken getestet.

Auch das jeweils verfügbare Root-Objekt für die einzelnen unterstützten Datenbanken wird

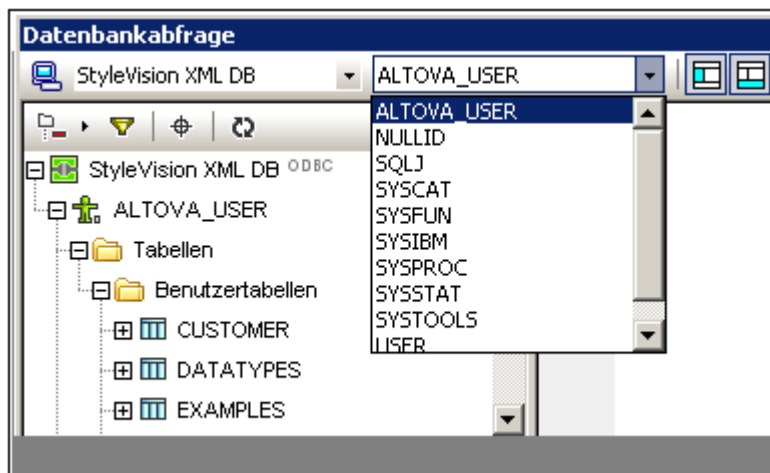
aufgelistet:

Datenbank (nativ unterstützt)	Root-Objekt
MS SQL Server 2000, 2005 und 2008	database
MS SQL Server 2005 und 2008	schema
Oracle 9i, 10g und 11g	schema
MS Access 2003 und 2007	database
MySQL 4.x und 5.x	database
IBM DB2 8.x und 9	schema
Sybase 12	database
IBM DB2 für i 5.4 und i 6.1	schema
PostgreSQL 8.0, 8.1, 8.2, 8.3	database

Anmerkung: Wenn Sie die 64-Bit-Version von StyleVision verwenden, stellen Sie sicher, dass Sie Zugriff auf die 64-Bit-Datenbanktreiber haben, die für die entsprechenden Datenbank benötigt werden.

Auswahl der erforderlichen Datenquelle

Alle bestehenden Verbindungen und ihre Root-Objekte werden in zwei Auswahllisten in der Symbolleiste des Fensters "Datenbankabfrage" (*Abbildung unten*) aufgelistet. Nach Auswahl der gewünschten Datenquelle in der Auswahlliste auf der linken Seite können Sie in der Auswahlliste auf der rechten Seite das gewünschte Root-Objekt auswählen.



In der Abbildung oben wurde die Datenbank mit dem Namen `StyleVision DB` ausgewählt. Von den verfügbaren Root-Objekten für diese Datenbank wurde das Root-Objekt `ALTOVA_USER` ausgewählt. Die Datenbank und das Root-Objekt werden anschließend im [Browser-Fenster](#) angezeigt.

Browser-Fenster: Anzeigen der DB-Objekte

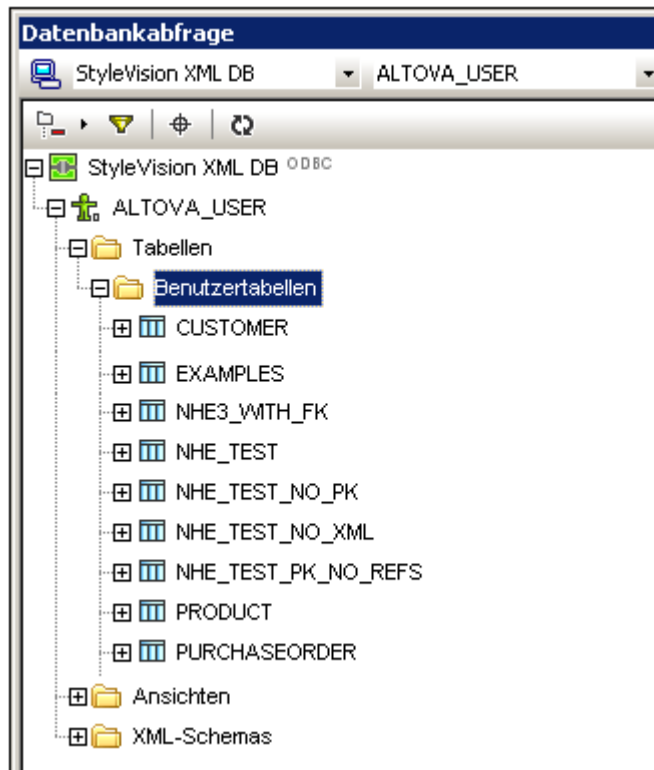
Im Browser-Fenster sehen Sie eine Übersicht über die Objekte in der ausgewählten Datenbank. Diese enthält Datenbank-Constraint-Informationen, z.B. ob es sich bei einer Spalte um einen Primär- oder Sekundärschlüssel handelt. In Datenbanken vom Typ IBM DB2 Version 9 sehen Sie im Browser zusätzlich registrierte XML-Schemas in einem separaten Ordner (*siehe Abbildung unten*).

In diesem Abschnitt wird Folgendes beschrieben:

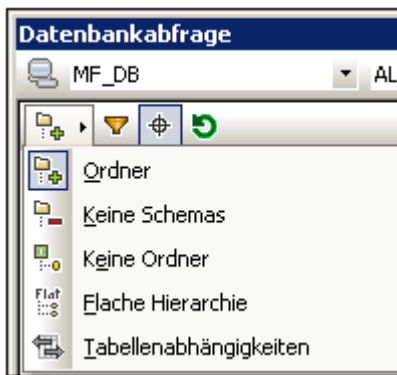
- Die im Browser-Fenster verfügbaren [Layouts](#).
- [Wie Datenbankobjekte gefiltert werden](#).
- [Wie Datenbankobjekte gesucht werden](#).
- Wie das Root-Objekt der aktiven Datenquelle [aktualisiert wird](#).

Browser-Fenster Layouts

Im Standard-Ordner-Layout werden Datenbankobjekte hierarchisch angezeigt. Je nachdem, welches Objekt ausgewählt ist, stehen unterschiedliche Kontextmenüoptionen zur Verfügung, wenn Sie mit der rechten Maustaste auf ein Objekt klicken.



Um ein Layout für den Browser auszuwählen, klicken Sie in der Symbolleiste des Browser-Fensters auf die Schaltfläche "Layout" und wählen Sie das Layout aus der Dropdown-Liste aus (*Abbildung unten*). Beachten Sie, dass sich die Schaltfläche ändert, je nachdem, welches Layout Sie ausgewählt haben.



Die verfügbaren Layouts sind:

- *Ordner*: Gliedert die Datenbankobjekte auf Basis des Objekttyps in hierarchisch angelegte Ordner. Dies ist die Standardeinstellung.
- *Keine Schemas*: Ähnlich dem Ordner-Layout, mit der Ausnahme, dass es keine Ordner für Datenbank-Schemas gibt; Tabellen werden daher nicht nach dem Datenbankschema gegliedert.
- *Keine Ordner*: Zeigt die Datenbankobjekte in einer Hierarchie an, ohne Ordner zu verwenden.
- *Flach*: Gliedert die Datenbankobjekte auf der ersten hierarchischen Ebene nach Typ. Anstatt dass Spalten z.B. in der entsprechenden Tabelle angezeigt werden, werden alle Spalten in einem separaten Spaltenordner angezeigt.
- *Tabellenabhängigkeiten*: Kategorisiert die Tabellen nach ihrer Beziehung zu anderen Tabellen. Es gibt Kategorien für Tabellen mit Sekundärschlüsseln, Tabellen, die durch Sekundärschlüssel referenziert werden und Tabellen, die keine Beziehung zu anderen Tabellen haben.

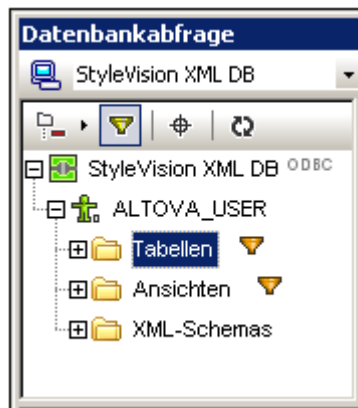
Um Tabellen nach Benutzer- und Systemtabellen zu sortieren, wechseln Sie ins Layout "Ordner", "Keine Schemas" oder "Flach", rechtsklicken Sie auf den Ordner "Tables" und wählen Sie den Befehl **Nach Benutzer- und Systemtabellen sortieren**. Die Tabellen werden in den Ordner "Benutzertabellen" und "Systemtabellen" in alphabetischer Reihenfolge sortiert.

Filtern von Datenbankobjekten

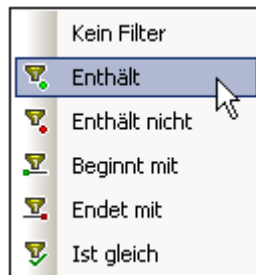
Im Browser-Fenster können Schemas, Tabellen und Ansichten (in allen Layouts mit Ausnahme des Layouts "Keine Ordner" und "Tabellenabhängigkeiten") nach Namen oder dem Teil eines Namens gefiltert werden. Die Objekte werden während der Eingabe der Zeichen gefiltert, dabei wird die Groß- und Kleinschreibung standardmäßig nicht beachtet.

So filtern Sie Objekte im Browser:

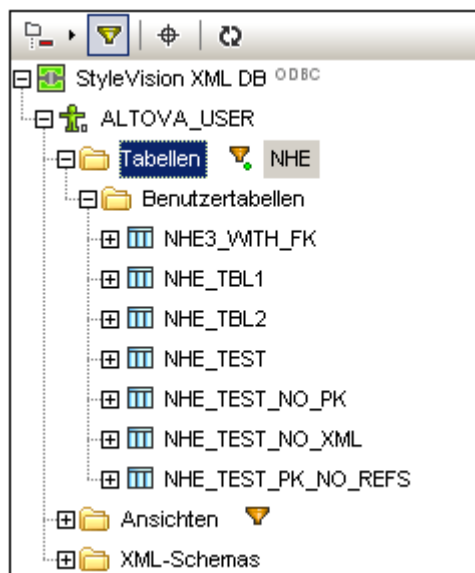
1. Klicken Sie in der Symbolleiste des Browser-Fensters auf die Schaltfläche "Ordnerinhalt filtern". Neben den Ordnern "Tables" und "Views" im ausgewählten Layout erscheinen Filtersymbole (*Abbildung unten*).



2. Klicken Sie auf das Filtersymbol neben dem gewünschten Ordner und wählen Sie aus dem Popup-Menü die Filteroption aus, z.B. *Enthält*.



3. Geben Sie im Eingabefeld, das nun angezeigt wird, den Filterstring ein (in der Abbildung unten ist der Filterstring für den Ordner *Tables* der String *NHE*). Der Filter wird während der Eingabe angewendet.

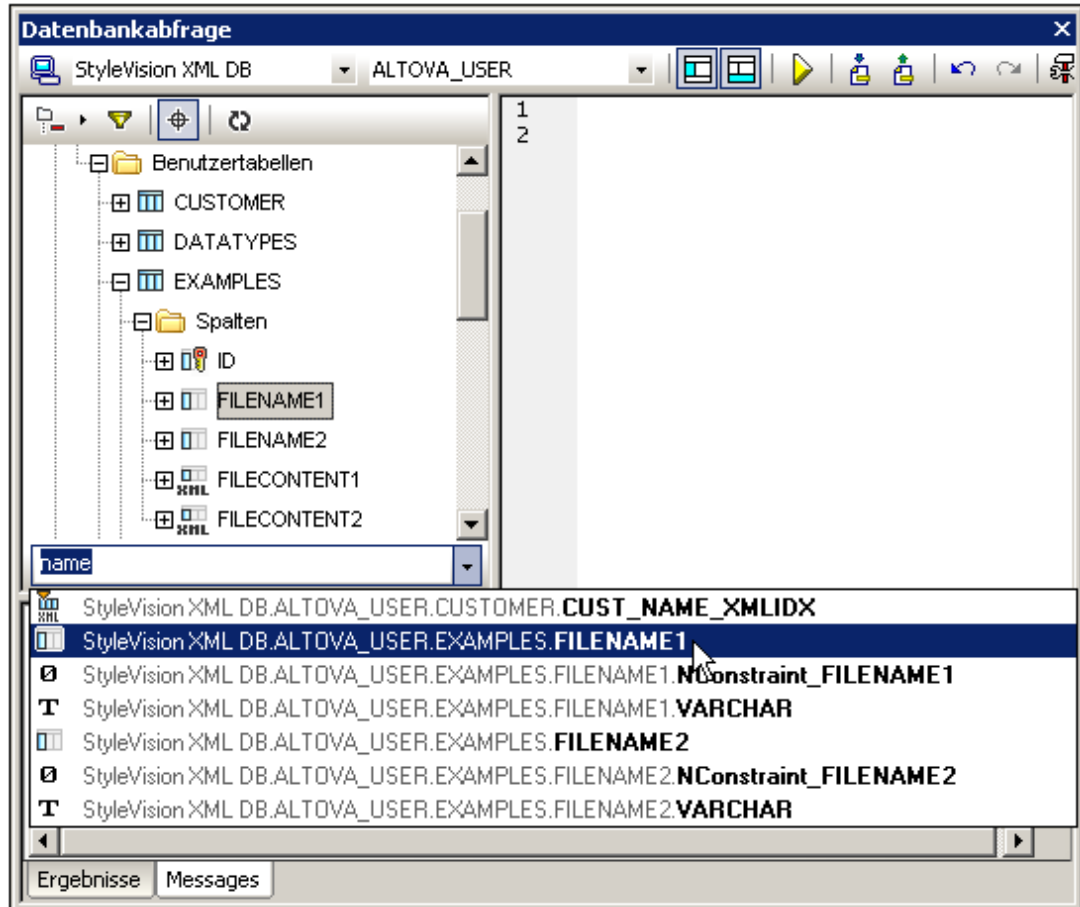


Suchen von Datenbankobjekten

Um ein bestimmtes Datenbankobjekt nach seinem Namen zu suchen, können Sie die Objektsuche des Browser-Fensters verwenden. Dies funktioniert folgendermaßen:

1. Klicken Sie in der Symbolleiste des Browser-Fensters auf die Schaltfläche

- "Objektsuche". Eine Dropdown-Liste erscheint am unteren Rand des Browsers.
2. Geben Sie den Suchstring, z.B. `name` in das Eingabefeld dieser Liste ein (*Abbildung unten*). Wenn Sie auf den Dropdown-Pfeil klicken, werden alle Objekte, die den Suchstring enthalten, angezeigt.



3. Klicken Sie auf das Objekt in der Liste, um es im Browser zu sehen.

Aktualisieren des Root-Objekts

Das Root-Objekt der aktiven Datenquellen kann durch Klicken auf die Schaltfläche "Aktualisieren" in der Symbolleiste des Browser-Fensters aktualisiert werden.

Abfragefenster: Beschreibung und Funktionen

Das Abfragefenster ist ein intelligenter SQL Editor zur Eingabe von Abfragen für die ausgewählte Datenbank. Klicken Sie nach Eingabe der Abfrage im Fenster "Datenbankabfrage" auf die Schaltfläche "[Ausführen](#)", um die Abfrage auszuführen und die Ergebnisse und Ausführungsmeldungen im [Fenster "Ergebnisse/Meldungen"](#) anzuzeigen. Eine Anleitung zum Arbeiten mit Abfragen finden Sie im nächsten Abschnitt, [Abfragefenster: Arbeiten mit Abfragen](#). In diesem Abschnitt werden die Hauptfunktionen des Abfragefensters beschrieben.

- [SQL Editor-Schaltflächen](#) in der Datenbankabfrage-Symbolleiste
- [SQL Editor-Optionen](#)
- [Autokomplettierung von SQL-Anweisungen](#)
- [Definition von Regionen in einem SQL Script](#)
- [Einfügen von Kommentaren in ein SQL-Script](#)
- [Verwendung von Lesezeichen](#)

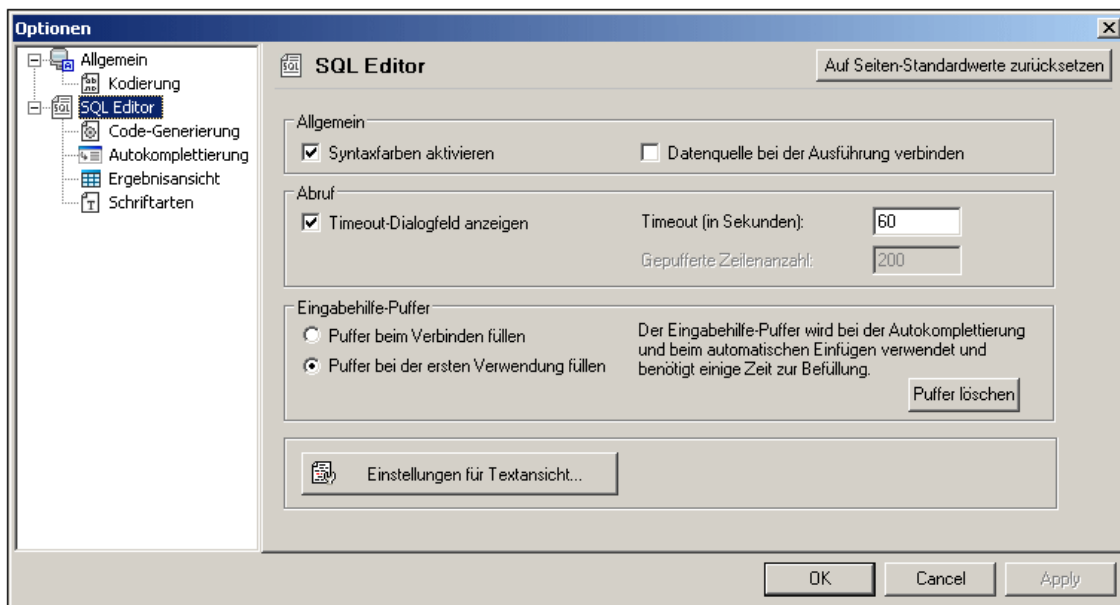
SQL Editor-Schaltflächen in der Datenbankabfrage-Symbolleiste

Für die Arbeit mit dem SQL Editor werden die folgenden Schaltflächen in der Symbolleiste des Fensters "Datenbankabfrage" verwendet:

	Ausführen	Führt die aktuell ausgewählte SQL-Anweisung aus. Wenn das Script mehrere Anweisungen enthält und keine ausgewählt wird, werden alle ausgeführt.
	SQL-Datei importieren	Öffnet eine SQL-Datei im SQL Editor.
	SQL-Datei exportieren	Speichert SQL-Abfragen in einer SQL-Datei.
	Rückgängig	Macht unbegrenzt viele Bearbeitungen im SQL Editor rückgängig.
	Wiederherstellen	Stellt unbegrenzt viele Bearbeitungen im SQL Editor wieder her.
	Optionen	Öffnet das Dialogfeld "Optionen" im SQL Editor.
	SQL-Script in DatabaseSpy öffnen	Öffnet das SQL Script in Altova DatabaseSpy.

SQL Editor Optionen

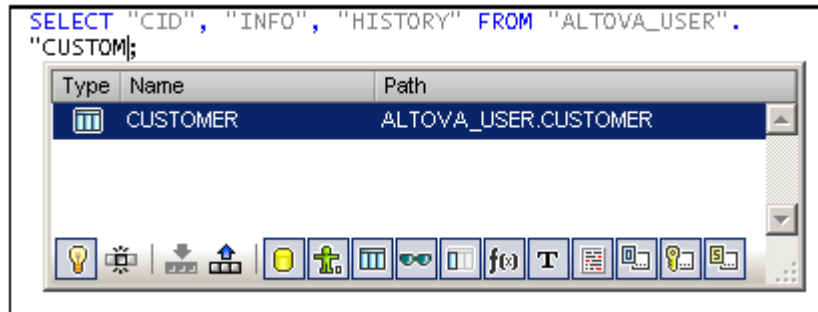
Wenn Sie in der Datenbankabfrage-Symbolleiste auf die Schaltfläche "Optionen" klicken, wird das Dialogfeld "Optionen" angezeigt (*Abbildung unten*). Im Bereich auf der linken Seite können Sie eine Einstellungsseite auswählen, deren Optionen Sie auswählen können. Klicken Sie auf die Schaltfläche **Seite auf Standardwerte zurücksetzen** um die Einstellungen auf dieser Seite auf die ursprünglichen Einstellungen zurückzusetzen.



Die wichtigsten Einstellungen sind die folgenden:

- **Allgemein | Kodierung:** Optionen zum Einstellen der Kodierung von neuen SQL-Dateien, von bestehenden SQL-Dateien, deren Kodierung nicht erkannt wurde und zur Einstellung der Bytefolgemarkierung (Byte Order Mark (BOM)). (Wenn die Kodierung bestehender SQL-Dateien erkannt wird, werden die Dateien geöffnet und gespeichert, ohne dass die Kodierung geändert wird.)
- **SQL Editor:** Optionen zum Aktivieren und Deaktivieren der Syntaxfärbung, eine Ein/Aus-Option zum Herstellen einer Verbindung zur Datenquelle bei Ausführung der SQL-Anweisung. Auch ein Dialogfeld zum Ändern des Timeout-Werts, kann angezeigt werden, wenn die definierte Zeit überschritten wird. Der Puffer für die Informationen der Eingabehilfen kann entweder bei Herstellung der Verbindung zur Datenquelle oder beim ersten Mal, wenn er benötigt wird, befüllt werden.
- **SQL-Editor | SQL-Generierung:** Wenn Sie Objekte aus dem Browser-Fenster in das Abfragefenster ziehen, generiert die Applikation SQL-Anweisungen. Auf dem Register "SQL-Generierung" können Optionen zur Generierung von SQL-Anweisungen ausgewählt werden. Im Bereich *Datenbank* können Sie eine Datenbankart auswählen und individuell für die verschiedenen Datenbankarten, mit denen Sie arbeiten, Optionen zur Generierung von SQL-Anweisungen auswählen. Wenn Sie das Kontrollkästchen *Auf alle Datenbanken anwenden* aktivieren, werden die gerade aktiven Optionen für alle Datenbanken angewählt. Zu den Optionen gehört die Möglichkeit, Semikola an Anweisungen anzuhängen und Identifier innerhalb von Escape-Zeichen zu setzen.
- **SQL-Editor | Autokomplettierung:** Die Autokomplettierungsfunktion schlägt Ihnen während der Eingabe SQL-Anweisungen aus verschiedenen SQL-Syntaxkategorien vor. Sie steht für die folgenden Datenbanken zur Verfügung: MS SQL Server 2000, 2005 und 2008, MS Access 2003 und 2007 und IBM DB2 v.9. Wenn die Option "Autokomplettierung" aktiviert ist, erscheint das Fenster "Autokomplettierung" mit Vorschlägen für die Autokomplettierung (*Abbildung unten*). Wählen Sie den gewünschten Eintrag aus, um ihn einzufügen. Sie können festlegen, ob das Autokomplettierungs-Popup-Fenster nach einer bestimmten Zeitspanne, die Sie im Fenster *Triggering Auto-completion* einstellen können, automatisch aufgerufen werden soll, oder ob Sie das Popup-Fenster manuell aufrufen müssen. Außerdem können Sie auch auswählen, mit welchen Tasten der ausgewählte Vorschlag eingefügt werden soll. Der SQL-Editor kann anhand von Sprachstatistiken intelligente Autokomplettierungsvorschläge machen. Wenn diese Funktion aktiviert ist, werden häufig verwendete Einträge ganz oben in der Liste der Vorschläge angezeigt. Beachten

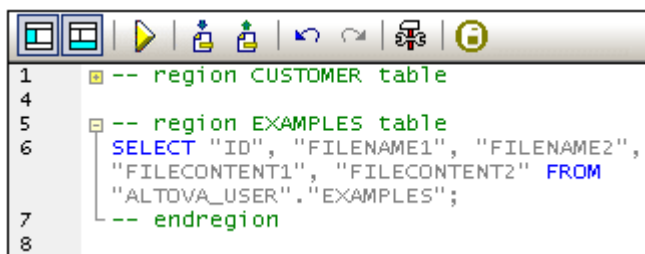
Sie die Schaltflächen am unteren Rand des Autokomplettierungsfensters. Mit der Schaltfläche *Kontextsensitiver Vorschlag* legen Sie fest, ob nur Einträge angezeigt werden sollen, die in diesem Kontext relevant sind, oder ob alle möglichen Einträge mit dieser Schreibweise angezeigt werden sollen. Mit der Schaltfläche *Einzelmodus* können Sie auf eine Categorieschaltfläche klicken, um nur diese Kategorie auszuwählen. Mit der Schaltfläche *Alle Kategorien definieren* werden alle Kategorien ausgewählt. Um eine Kategorie von der Auswahl abzunehmen, klicken Sie anschließend auf die Schaltfläche für diese Kategorie. Mit der Schaltfläche *Alle Kategorien löschen* wird die Auswahl für alle Kategorien aufgehoben. Die anderen Schaltflächen sind die Schaltflächen für die verschiedenen Kategorien.



- **SQL Editor | Ergebnisansicht:** Optionen zum Konfigurieren des Ergebnisregisters.
- **SQL Editor | Schriftarten:** Optionen zum Einstellen der Schriftart für den Text im Text-Editor und in der Ergebnisansicht.

Definition von Regionen in einem SQL Script

Regionen sind Abschnitte in SQL Scripts, die als Einheit markiert und deklariert sind. Regionen können erweitert und reduziert werden, um Teile von SQL Scripts ein- oder auszublenden. Regionen können auch in andere Regionen verschachtelt werden. Regionen werden vor und nach dem ausgewählten Text durch den Kommentar `--region` bzw. `--endregion` gekennzeichnet. Regionen können auch optional einen Namen erhalten, der nach der `--region` Markierung eingegeben wird (siehe Abbildung unten).



Um eine Region einzufügen, wählen Sie die gewünschte(n) Anweisungen aus, die zu einer Region gemacht werden sollen, rechtsklicken Sie und wählen Sie den Befehl **Region einfügen**. Es wird eine erweiterbare/reduzierbare Region erstellt. Fügen Sie ggf. einen Namen hinzu. Beachten Sie in der Abbildung oben auch die Zeilennummerierung. Um eine Region zu löschen, löschen Sie die beiden `--region` und `--endregion` Markierungen.

Einfügen von Kommentaren in ein SQL-Skript

Text in einem SQL-Skript kann auskommentiert werden. Diese Script-Abschnitte werden bei der Ausführung des Scripts übersprungen.

- Um einen Block auszukommentieren, markieren Sie den Block, rechtsklicken Sie und wählen Sie den Befehl **Blockkommentar einfügen/entfernen**. Um den

Blockkommentar zu entfernen, rechtsklicken Sie und wählen Sie den Befehl **Blockkommentar einfügen/entfernen**.

- Um eine Zeile oder den Teil einer Zeile auszukommentieren, platzieren Sie den Cursor an die Stelle, an der der Zeilenkommentar beginnen soll, rechtsklicken Sie und wählen Sie den Befehl **Zeilenkommentar einfügen/entfernen**. Um den Zeilenkommentar zu entfernen, markieren Sie den Kommentar, rechtsklicken Sie und wählen Sie den Befehl **Zeilenkommentar einfügen/entfernen**

Verwendung von Lesezeichen

Lesezeichen können an bestimmten Zeilen eingefügt werden. Sie können anschließend durch die Lesezeichen im Dokument navigieren. Um ein Lesezeichen einzufügen, platzieren Sie den Cursor in die gewünschte Zeile, rechtsklicken Sie und wählen Sie den Befehl **Lesezeichen einfügen/löschen**. Um zum nächsten oder vorhergehenden Lesezeichen zu navigieren, wählen Sie den Befehl **Nächstes Lesezeichen** bzw. **Vorheriges Lesezeichen**. Um ein Lesezeichen zu entfernen, platzieren Sie den Cursor in die Zeile, in der das Lesezeichen entfernt werden soll, rechtsklicken Sie und wählen Sie den Befehl **Lesezeichen einfügen/löschen**. Um alle Lesezeichen zu entfernen, rechtsklicken Sie und wählen Sie den Befehl **Alle Lesezeichen löschen**.

Abfragefenster: Arbeiten mit Abfragen

Nachdem Sie eine Verbindung zu einer Datenbank hergestellt haben, kann ein SQL-Script im SQL Editor eingegeben und ausgeführt werden. In diesem Abschnitt wird beschrieben:

- wie ein SQL-Script in SQL Editor eingegeben wird.
- wie das Script im Fenster "Datenbankabfrage" ausgeführt wird.

In diesem Abschnitt werden die folgenden Schaltflächen verwendet:



Abfrage Ausführen

Führt die aktuell ausgewählte SQL-Anweisung aus. Wenn das Script mehrere Anweisungen enthält und keine ausgewählt wird, werden alle ausgeführt.



SQL-Datei importieren

Öffnet eine SQL-Datei im SQL Editor.

Erstellen von SQL-Anweisungen und Scripts im SQL Editor

Zum Erstellen von SQL-Anweisungen und Scripts können die folgenden GUI-Methoden verwendet werden:

- *Drag-and-Drop*: Ziehen Sie ein Objekt aus dem Browser-Fenster in den SQL Editor. Eine SQL-Anweisung wird generiert, um die Datenbank nach diesem Objekt abzufragen.
- *Kontextmenü*: Rechtsklicken Sie auf ein Objekt im Browser-Fenster und wählen Sie den Befehl **In SQL Editor anzeigen | Auswählen**.
- *Manuelle Eingabe*: Geben Sie die SQL-Anweisung direkt im SQL Editor ein. Beim Bearbeiten hilft Ihnen die Autokomplettierungsfunktion.
- *SQL-Script importieren*: Klicken Sie in der Symbolleiste des Fensters "Datenbankabfrage" auf die Schaltfläche **SQL-Datei importieren**.

Ausführen von SQL-Anweisungen

Wenn das SQL-Script im SQL Editor mehr als eine SQL-Anweisung enthält, wählen Sie die gewünschte Anweisung aus und klicken Sie in der Symbolleiste des Fensters "Datenbankabfrage" auf die Schaltfläche **Ausführen**. Wenn im SQL-Script keine Anweisung ausgewählt ist, werden alle Anweisungen im Script ausgeführt. Die Datenbankdaten werden abgerufen und in einem Raster auf dem [Ergebnisregister angezeigt](#). Meldungen über die Ausführung der Anweisung werden auf dem [Meldungsregister](#) angezeigt.

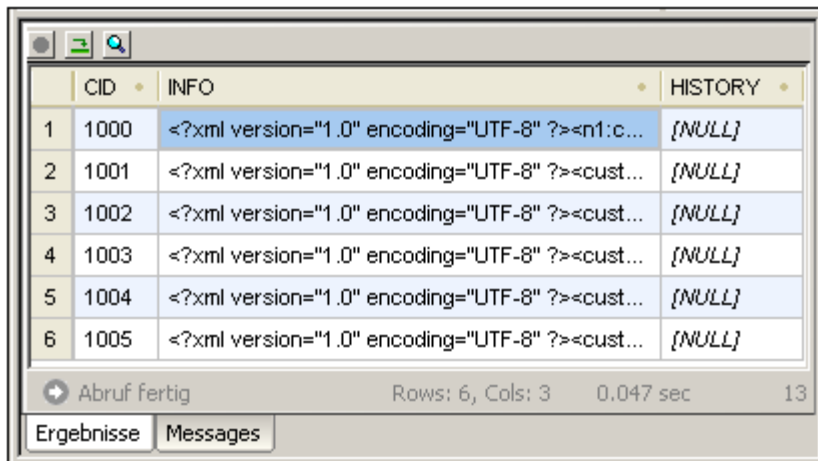
Ergebnisse und Meldungen

Das Ergebnis-/Meldungsfenster hat zwei Register:

- Auf dem [Register "Ergebnisse"](#) werden die von der Abfrage abgerufenen Daten angezeigt.
- Auf dem [Register "Meldungen"](#) sehen Sie Meldungen über die Ausführung der Abfrage.

Register "Ergebnisse"

Die von der Abfrage abgerufenen Daten werden in Tabellenform auf dem Ergebnisregister angezeigt (*Abbildung unten*).



	CID	INFO	HISTORY
1	1000	<?xml version="1.0" encoding="UTF-8" ?><n1:c...	{NULL}
2	1001	<?xml version="1.0" encoding="UTF-8" ?><cust...	{NULL}
3	1002	<?xml version="1.0" encoding="UTF-8" ?><cust...	{NULL}
4	1003	<?xml version="1.0" encoding="UTF-8" ?><cust...	{NULL}
5	1004	<?xml version="1.0" encoding="UTF-8" ?><cust...	{NULL}
6	1005	<?xml version="1.0" encoding="UTF-8" ?><cust...	{NULL}

Abruf fertig Rows: 6, Cols: 3 0.047 sec 13

Ergebnisse Messages

Die folgenden Operationen können auf dem Ergebnisregister über das Kontextmenü, das angezeigt wird, wenn Sie auf dem Ergebnisregister an der entsprechenden Stelle auf die rechte Maustaste klicken, ausgeführt werden:

- **Nach einer Spalte sortieren:** Rechtsklicken Sie an eine beliebige Stelle in der Spalte, nach der die Ergebnisse sortiert werden sollen und wählen Sie den Befehl **Sortierung | Aufsteigend/Absteigend/StandardEinstellung wiederherstellen**.
- **Kopieren in die Zwischenablage:** Dieser Vorgang erfolgt in zwei Schritten: (i) Auswählen des Datenbereichs; und (ii) Kopieren der Auswahl. Die Daten können auf verschiedene Arten ausgewählt werden: (i) durch Klicken auf eine Spaltenüberschrift zur Auswahl der Spalte bzw. Zeile; (ii) durch Auswählen einzelner Zellen (Zur Auswahl mehrerer Zellen verwenden Sie **Umschalt** und/oder **Strg**); (iii) durch Rechtsklick auf eine Zelle und Auswahl von **Auswahl | Zeile/Spalte/Alle**. Klicken Sie anschließend auf die rechte Maustaste und wählen Sie den Befehl **Ausgewählte Zellen kopieren**. Daraufhin wird die Auswahl in die Zwischenablage kopiert, von wo sie in andere Applikationen eingefügt werden kann.

Das Ergebnisregister hat die folgenden Symbolleistenaltflächen:



**Gehe zu
Anweisung**

Markiert die SQL Anweisung im SQL Editor, die das entsprechende Ergebnis erzeugt hat.

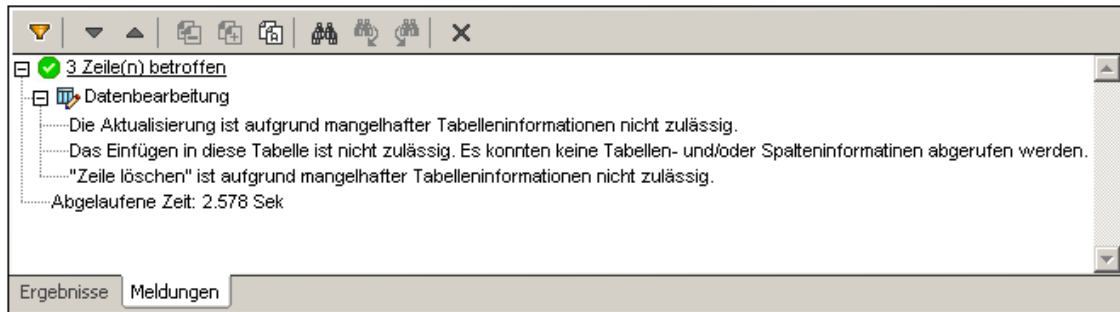


Suchen

Sucht Text im Ergebnisfenster. Aus XML-Dokumentinhalt wird durchsucht.

Register "Meldungen"

Das Register "Meldungen" enthält Informationen über die zuvor ausgeführte SQL-Anweisung und gibt Fehlermeldungen und Warnmeldungen aus.



Die Symbolleiste des Meldungsregisters enthält Schaltflächen, mit denen Sie die Ansicht anpassen, darin navigieren sowie Meldungen in die Zwischenablage kopieren können. Mit Hilfe der Schaltfläche **Filter** können Sie bestimmte Arten von Meldungen ein- und ausblenden. Mit Hilfe der Schaltflächen **Nächste** und **Vorherige** wird die jeweils vorige bzw. nächste Meldung in der Liste markiert. Meldungen können auch mit oder ohne Ihre Sub-Einträge in die Zwischenablage kopiert werden, sodass sie in Dokumente eingefügt werden können. Mit Hilfe der Funktion **Suchen** können Sie einen Suchbegriff angeben und dann die Liste nach diesem Begriff durchsuchen. Mit der Schaltfläche **Löschen** können Sie den Inhalt des Berichtsfensters löschen.

Anmerkung: Diese Symbolleisten-Befehle stehen auch im Kontextmenü zur Verfügung.

Kapitel 14

Authentic-Ansicht

14 Authentic-Ansicht

Die Authentic-Ansicht (*Abbildung unten*) ist eine grafische Darstellung Ihres XML-Dokuments. Sie können in dieser Ansicht XML-Dokumente ohne Markup-Code und mit der richtigen Formatierung und den Dateneingabefunktionen wie z.B. Eingabefeldern, Auswahllisten und Optionsfeldern anzeigen. Die Daten, die der Benutzer in der Authentic-Ansicht eingibt, werden in die XML-Datei eingegeben.

Nanonull, Inc.

Location:

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

Vereno Office Summary: 4 departments, 15 employees.

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

Authentic-Vorschau

Sie können in StyleVision während der Bearbeitung eines SPS eine Vorschau auf die Authentic-Ansicht der zugewiesenen XML-Arbeitsdatei anzeigen. Wenn Sie auf das Register "Authentic-Ansicht" klicken und dem SPS noch keine XML-Arbeitsdatei zugewiesen wurde, werden Sie aufgefordert dies zu tun. In der Authentic-Vorschau können Sie das XML-Dokument ähnlich wie in der Standard-Authentic-Ansicht bearbeiten. Die Änderungen können in der XML-Arbeitsdatei gespeichert werden. In diesem Abschnitt finden Sie eine Beschreibung der [Authentic-Ansicht](#) sowie [eine Anleitung zur Bearbeitung von Dokumenten in der Authentic-Ansicht](#).

14.1 Authentic-Ansicht

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

Dieser Abschnitt enthält:

- eine Übersicht über die Benutzeroberfläche
- eine Beschreibung der Symbolleistensymbole der Authentic-Ansicht
- eine Beschreibung der Ansichtsmodi im Authentic Vorschau-Hauptfenster
- eine Beschreibung der verschiedenen in der Authentic-Ansicht verfügbaren Kontextmenüs

Weitere Informationen zur Authentic-Ansicht finden Sie:

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

Übersicht über die Benutzeroberfläche

In der Authentic-Vorschau stehen Ihnen Menübefehle, Symbolleistenschaltflächen und Kontextmenüs zur Verfügung, mittels derer Sie das im Hauptfenster angezeigte XML-Dokument bearbeiten können.

Menüleiste

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

Symbolleiste

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

Hauptfenster

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

Statusleiste

In der Statusleiste wird der XPath zum aktuell ausgewählten Node angezeigt. In der Authentic-Vorschau von StyleVision wird der XPath zum aktuell ausgewählten Node im Schema-Baumdiagramm angezeigt, wo der jeweils ausgewählte Node grau markiert erscheint. Der XPath wird in der Authentic-Ansicht nicht in der Statusleiste angezeigt.

Kontextmenüs

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

Authentic-Ansicht Symbolleistenschaltflächen

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

Ein- und Ausblenden von XML Markup

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



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



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



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



Gemischte Markup-Symbole einblenden. Im StyleVision Power Stylesheet können Sie für jedes XML-Element oder Attribut einzeln festlegen, wie es im "Gemischten Markup"-Modus angezeigt werden soll (mit großen oder kleinen Markup-Symbolen oder gar nicht). Der Modus heißt "Gemischte Markup-Symbole", weil darin für einige Elemente mit Markup-Symbolen angezeigt werden, andere dagegen ohne. Im gemischten Markup-Modus der Authentic-Ansicht ist die Anzeige der Markup-Symbole also benutzerdefiniert. Beachten Sie allerdings, dass diese Einstellungen von der Person vorgenommen werden, die das StyleVision Power Stylesheet erstellt hat. Sie kann nicht vom Benutzer der Authentic-Ansicht vorgenommen werden.

Bearbeiten von dynamischen Tabellenstrukturen

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

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

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





Zeile anhängen



Zeile einfügen



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



Zeile nach oben



Zeile nach unten



Zeile löschen

Bitte beachten Sie: Diese Befehle sollten nur bei **dynamischen SPS-Tabellen** und nicht innerhalb von statischen SPS-Tabellen verwendet werden.

Symbole für die Navigation in DB-Zeilen



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



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

XML-Datenbankbearbeitung

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

XML-Dateibefehle

Die folgenden Schaltflächen entsprechen von links nach rechts gesehen den unten aufgelisteten Befehlen:



- *XML-Authentic-Daten speichern:* Speichert die XML-Datendatei.
- *XML-Authentic-Daten speichern unter...:* Speichert die XML-Datendatei unter einem anderen Namen.
- *Authentic-Ansicht neu laden:* Lädt die gespeicherte XML-Datei neu. Alle nicht gespeicherten Änderungen gehen verloren.
- *Validieren:* Validiert die XML-Datendatei.

Authentic-Ansicht Hauptfenster

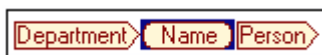
In der Authentic-Ansicht gibt es vier Ansichtsmodi: Große Markup-Symbole; Kleine Markup-Symbole; Gemischte Markup-Symbole; und Alle Markup-Symbole ausblenden. Damit können Sie die Markup-Informationen für Dokumente mit unterschiedlicher Genauigkeit anzeigen. Um zwischen diesen Ansichtsmodi zu wechseln,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

Große Markup-Symbole

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



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



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



Kleine Markup-Symbole

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

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

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

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

Optionale Nodes

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

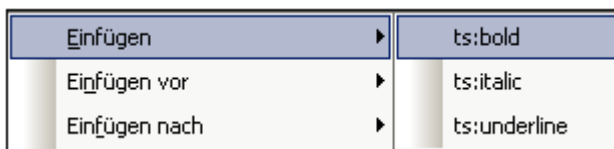
Wenn Sie auf die Eingabeaufforderung klicken, wird das Element hinzugefügt und der Cursor so gesetzt, dass Daten eingegeben werden können. Wenn es mehrere optionale Nodes gibt, wird die Aufforderung "add..." angezeigt. Wenn Sie darauf klicken, wird ein Menü mit den optionalen Nodes angezeigt.

Authentic-Ansicht Kontextmenüs

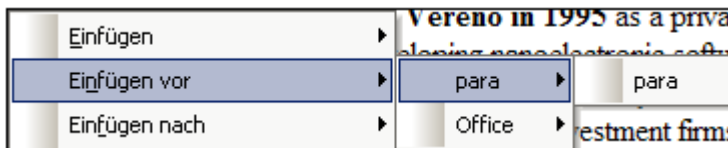
Wenn Sie mit der rechten Maustaste auf einen ausgewählten Dokumenteninhalte oder Node klicken, wird ein Kontextmenü mit Befehlen, die für die Auswahl oder Cursorposition relevant sind, angezeigt.

Einfügen von Elementen

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



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



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

Einfügen einer Entity

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

Einfügen eines CDATA-Abschnitts

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

Entfernen eines Node

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

Entfernung eines übergeordneten Elements auch alle untergeordneten Elementen einschließlich des ausgewählten Elements entfernt werden.

Löschen

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

Übernehmen

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

Kopieren, Ausschneiden, Einfügen

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

Einfügen als

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

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

Anmerkung: Text wird in Nodes kopiert, in denen Text zulässig ist. Sie müssen daher sicher stellen, dass das Dokument durch den kopierten Text nicht ungültig wird. Der kopierte Text sollte daher:

- (i) an der neuen Stelle lexikalisch gültig sein (so wären z.B. nicht numerische Zeichen in einem numerischen Node ungültig) und
- (ii) den Node nicht auf andere Weise ungültig machen (so würde der Node z.B. bei Einfügen einer vierstelligen Zahl in einen Node, in dem nur dreistellige Zahlen zulässig sind, ungültig werden).

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

Löschen

Mit dem Befehl **Löschen** wird der ausgewählte Node und sein gesamter Inhalt gelöscht. Ein

Node gilt in diesem Zusammenhang als ausgewählt, wenn der Cursor in den Node platziert wurde oder indem Sie entweder auf den Start- oder End-Tag des Node klicken.

14.2 Bearbeitung in der Authentic-Ansicht

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

In diesem Abschnitt werden folgende Themen behandelt:

- Die Datumsauswahl ist ein grafischer Kalender, der Daten im korrekten XML-Format eingibt, wenn Sie auf ein Datum klicken. Siehe [Datumsauswahl](#).
- Eine Entity ist ein Kürzel für ein Sonderzeichen oder einen Textstring. Sie können Ihre eigenen Entities definieren, um Sonderzeichen oder Textstrings durch Eingabe der Entity einfügen zu können. Nähere Informationen dazu finden Sie unter [Definieren von Entities](#).
- In der Enterprise und der Professional Edition von Altova-Produkten kann der Benutzer der Authentic-Ansicht XML-Dokumente mit digitalen XML-Signaturen signieren und diese Signaturen überprüfen.
- Die [Bildformate](#), die in der Authentic-Ansicht angezeigt werden können.

Um sich mit der Verwendung aller Funktionen der Authentic-Ansicht vertraut zu machen, arbeiten Sie bitte das Tutorial zur Authentic-Ansicht durch (Verwenden Sie dazu entweder XMLSpy oder die Authentic Desktop). Die Tutorials zur Authentic-Ansicht sind zusammen mit diesen Produkten erhältlich.

Grundlegendes zur Bearbeitung

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

Vereno Office Summary: 4 departments, 16 employees.

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

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

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

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

para para

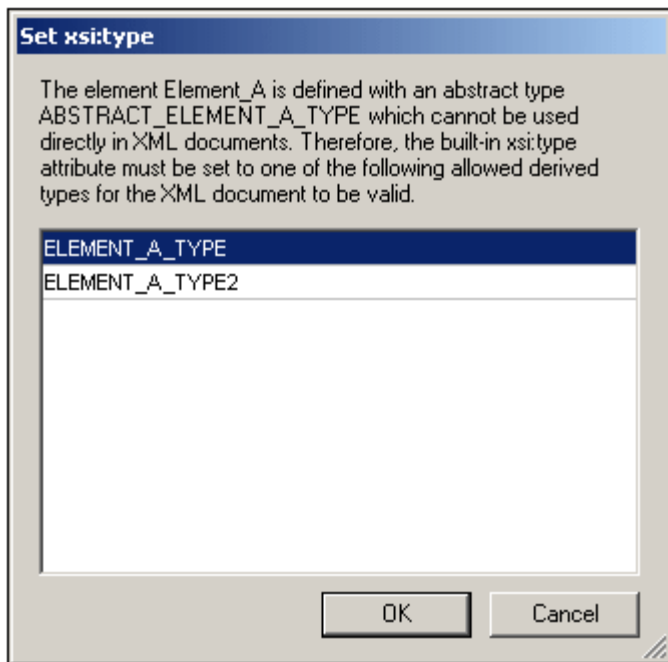
Ein editierbares Dokument in der Authentic-Ansicht mit XML-Markup-Tags

Einfügen von Nodes

Oft kommt es vor, dass ein neuer Node in das XML-Authentic-Dokument eingefügt werden muss. So muss z.B. möglicherweise ein neues `Person` Element zu einem Dokument vom Typ Adressbuch hinzugefügt werden. In solchen Fällen würde das XML-Schema das Hinzufügen des neuen Elements gestatten. Klicken Sie einfach mit der rechten Maustaste auf den Node im Authentic-Ansichtsdokument, vor bzw. hinter dem Sie den neuen Node hinzufügen möchten. Wählen Sie im Kontextmenü, das daraufhin angezeigt wird, je nach Bedarf den Befehl **Einfügen vor** oder **Einfügen nach** aus. Die Nodes, die an dieser Stelle in das Dokument eingefügt werden dürfen, werden in einem Untermenü aufgelistet. Klicken Sie auf den

gewünschten Node, um ihn einzufügen. Der Node und alle obligatorischen untergeordneten Nodes werden eingefügt. Falls ein untergeordneter Node optional ist, wird ein Link [Add NodeName](#) angezeigt, auf den Sie klicken können, wenn der optionale Node eingefügt werden soll.

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



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

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

Der ausgewählte derived type kann in der Folge durch Ändern des Werts des `xsi:type` Attributs des Elements in der Attribut-Eingabehilfe geändert werden. Wenn der Typ des Elements auf diese Art geändert wird, werden alle Nodes des früheren Content Model entfernt und die Nodes des Typs des neuen Content Model werden eingefügt.

Bearbeitung von Text

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

Kopieren als XML oder als Text

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

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

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

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

Formatierung von Text

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

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

Verwendung von RichEdit in der Authentic-Ansicht

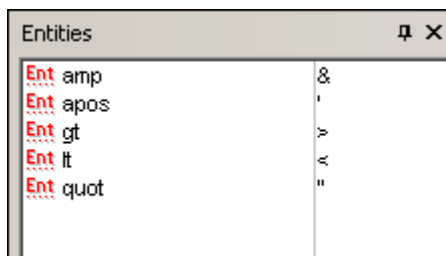
Wenn Sie den Cursor in der Authentic-Ansicht innerhalb von einem Element platzieren, das als RichEdit-Komponente erstellt wurde, werden die Schaltflächen und Steuerelemente in der RichEdit-Symbolleiste (*Abbildung unten*) aktiv. Anderfalls sind sie ausgegraut.



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

Einfügen von Entities

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



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

Einfügen von CDATA-Abschnitten

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

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

Bearbeiten von Links und Navigieren zum Ziel eines Links

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

Tabellen in der Authentic-Ansicht

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

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

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

SPS-Tabellen

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

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

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

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

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



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

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

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

drücken, wird eine neue Zeile erstellt.

CALS/HTML-Tabellen

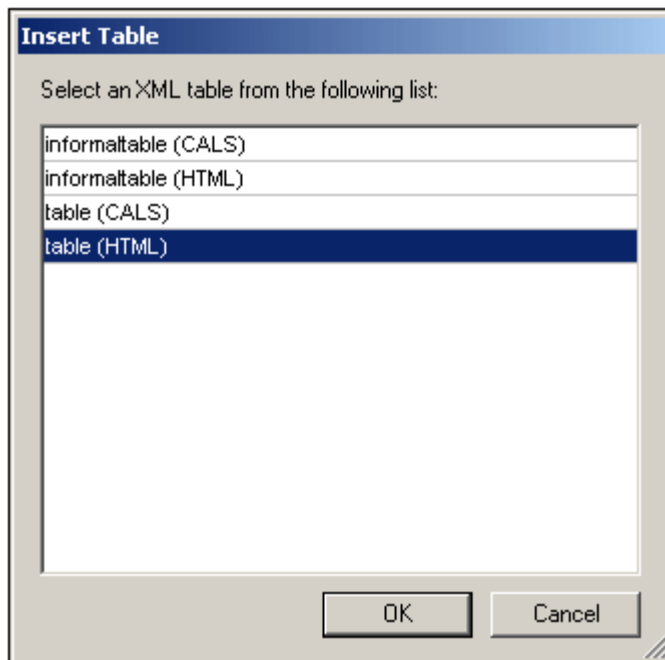
CALS/HTML-Tabellen können von Ihnen, dem Benutzer der Authentic-Ansicht für bestimmte XML-Datenstrukturen verwendet werden, die als Tabellenformat definiert wurden. Die Arbeit mit CALS/HTML-Tabellen erfordert drei Schritte: Einfügen der Tabelle, Formatierung der Tabelle und Eingabe von Daten. Die Befehle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

Einfügen von Tabellen

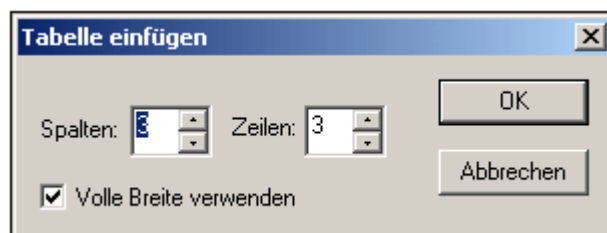
So fügen Sie eine CALS/HTML-Tabelle ein:

1. Setzen Sie Ihren Cursor an die Stelle, an der die Tabelle eingefügt werden soll und

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



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



Daraufhin wird anhand der Angaben oben die folgende Tabelle erstellt.

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

Formatieren von Tabellen und Eingabe von Daten

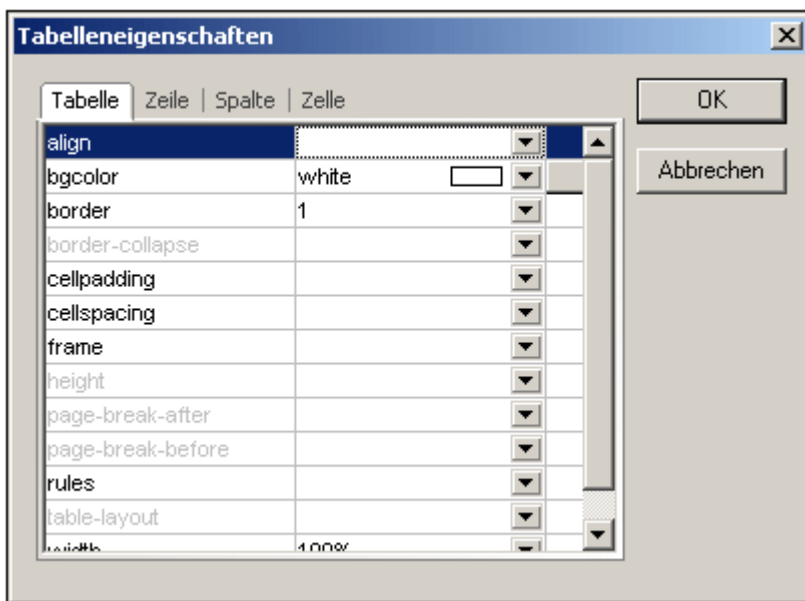
Die Tabellenformatierung wurde bereits im Dokumentdesign zugewiesen. Unter bestimmten Umständen können Sie die Tabellenformatierung jedoch ändern. Diese Umstände sind:

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

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

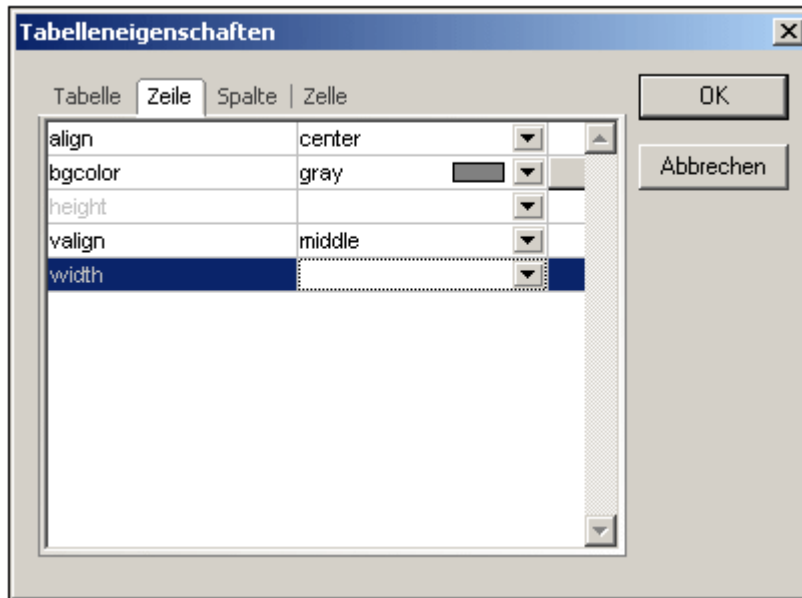
1. Platzieren Sie Ihren Cursor zum Formatieren der Tabelle an eine beliebige Stelle in der

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



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

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



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

Name	Telephone	Email

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

Name	Telephone		Email

5. Setzen Sie nun den Cursor in die Zelle unterhalb der Zelle mit dem Eintrag

"Telephone" und klicken Sie auf das Symbol  (Horizontal teilen). Geben Sie dann die Spaltenüberschriften "Office" und "Home" ein. Ihre Tabelle sieht nun folgendermaßen aus:

Name	Telephone		Email
	Office	Home	

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

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

Navigieren innerhalb von Tabellenzellen

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

Eingabe von Daten in eine Zelle

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

Formatieren von Text

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

Name	Telephone		Email
	Office	Home	

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

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

Schaltflächen zum Bearbeiten von CALS/HTML-Tabellen

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

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

Tabelle einfügen



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

Tabelle löschen



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

Zeile anhängen



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

Spalte anhängen



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

Zeile einfügen



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

Spalte einfügen



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

Links zusammenfügen



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

Rechts zusammenfügen



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

Unten zusammenfügen



Der Befehl "Unten zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der

Zelle unterhalb davon zusammen. Die Inhalte von beiden Zellen werden in der neuen Zelle zusammengeführt.

Oben zusammenfügen



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

Horizontal teilen



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

Vertikal teilen



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

Bündig oben



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

Vertikal zentrieren



Dieser Befehl zentriert den Inhalt der Zelle.

Bündig unten

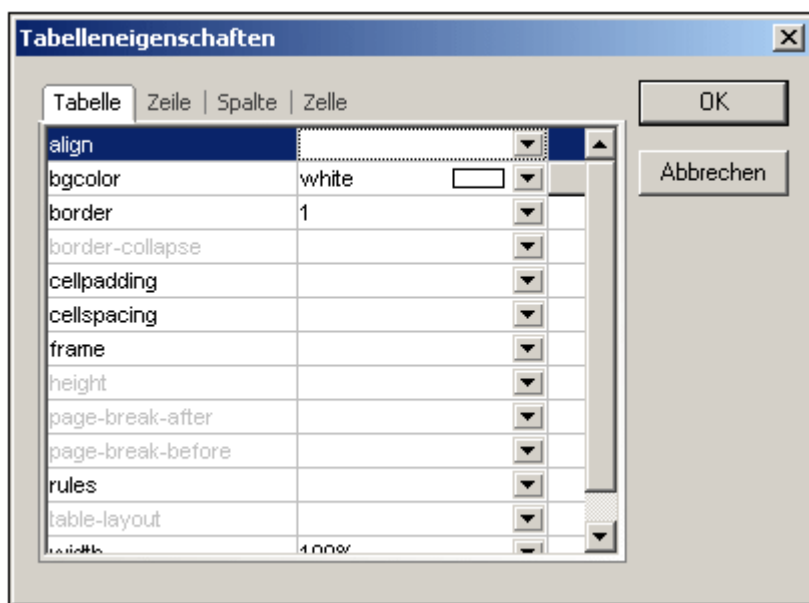


Dieser Befehl richtet den Inhalt der Zellen bündig am unteren Zellenrand aus.

Tabelleneigenschaften



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



Bearbeiten einer DB

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

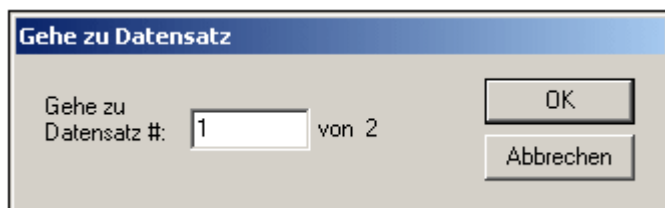
- Die Anzahl der Datensätze in einer DB-Tabelle, die in der Authentic-Ansicht angezeigt werden können, wurde unter Umständen vom Autor des StyleVision Power Stylesheet absichtlich eingeschränkt, um das Design kompakter zu gestalten. In diesem Fall wird anfangs nur diese beschränkte Anzahl an Datensätzen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

Navigieren in einer DB-Tabelle

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



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



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

XML-Datenbanken

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

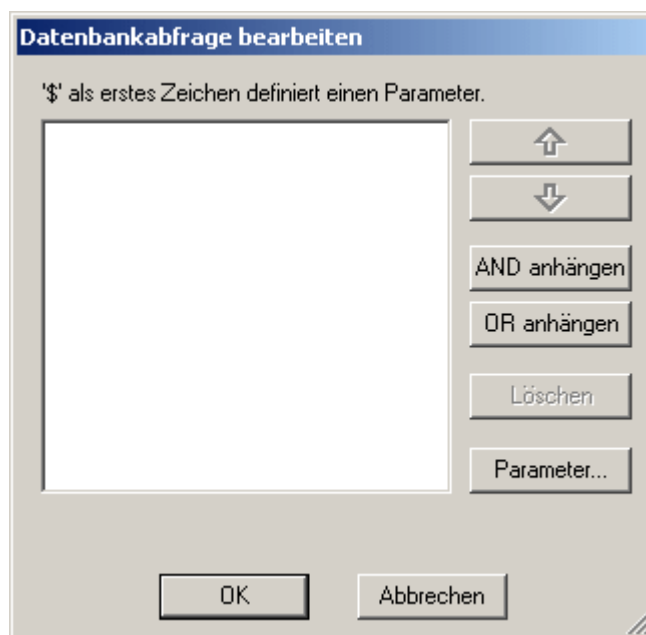
DB-Abfragen

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

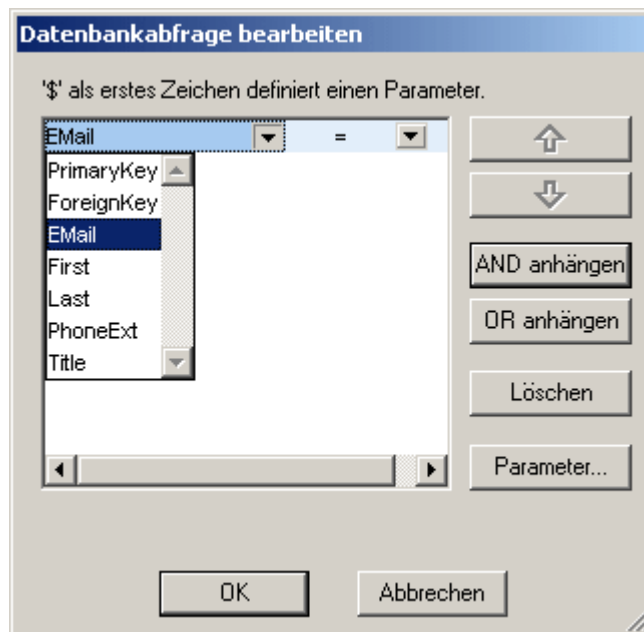
Bitte beachten Sie: Falls eine Meldung angezeigt wird, dass zu viele Tabellen geöffnet sind, können Sie die Anzahl der offenen Tabellen mit Hilfe einer Abfrage durch Filtern reduzieren.

So erstellen Sie eine Abfrage und senden diese ab:

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



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



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

Ausdrücke in Kriterien

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

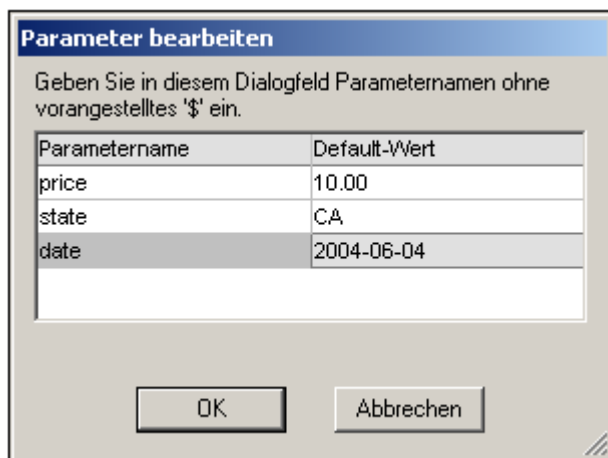
=	Ist gleich
<>	Ist nicht gleich
<	Kleiner als
<=	Kleiner oder gleich
>	Größer als
>=	Größer oder gleich
LIKE	Phonetisch ähnlich
NOT LIKE	Phonetisch nicht ähnlich
IS NULL	Ist leer
NOT NULL	Ist nicht leer

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

Verwendung von Parametern mit DB-Abfragen

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

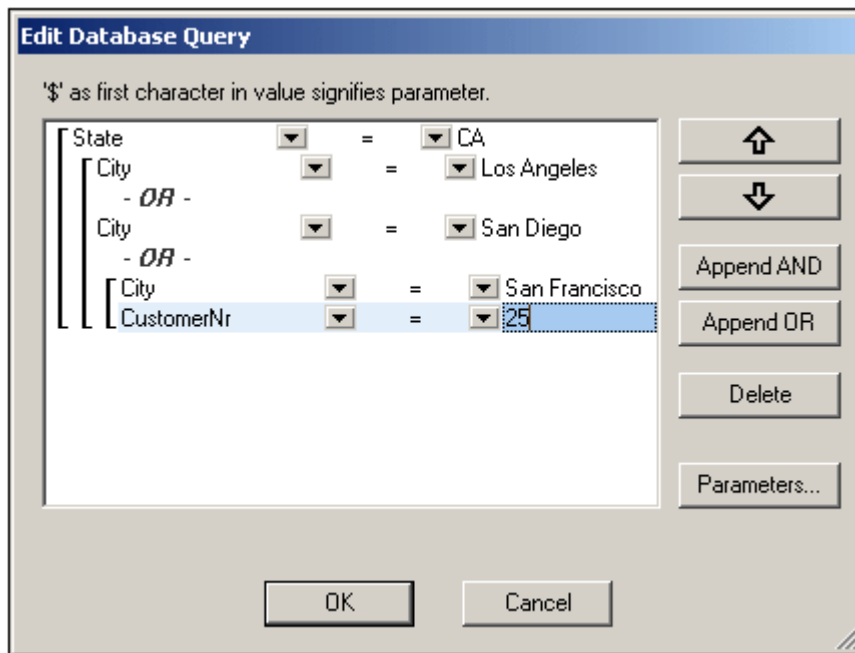
Um die Parameter, die für das SPS definiert wurden, anzuzeigen, klicken Sie im Dialogfeld **Datenbankabfrage bearbeiten** auf die Schaltfläche **Parameter**, um das unten gezeigte Dialogfeld **Parameter anzeigen** aufzurufen.



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

Neuordnen von Kriterien in DB-Abfragen

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



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

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

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

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

Beachten Sie die folgenden Punkte:

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

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

Ändern einer DB-Abfrage

So ändern Sie eine DB-Abfrage:

1. Klicken Sie auf die Schaltfläche "Abfrage" , um das Dialogfeld "Datenbankabfrage bearbeiten" aufzurufen. Sie können nun die Ausdrücke in jedem der aufgelisteten Kriterien ändern, neue Kriterien hinzufügen oder Kriterien

- aus der DB-Abfrage löschen.
2. Klicken Sie auf OK. Die Daten aus der DB werden automatisch in Authentic neu geladen, um die Änderungen an der DB-Abfrage anzuzeigen.

Ändern einer DB-Tabelle

Hinzufügen eines Datensatzes

So fügen Sie einen Datensatz hinzu:

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

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

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

Ändern eines Datensatzes

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

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

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

Bitte beachten Sie:

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

Löschen eines Datensatzes

So löschen Sie einen Datensatz:

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

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

Arbeiten mit Datumsangaben

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

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

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

Anmerkung zu Datumsformaten

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

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

Datumswahl

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

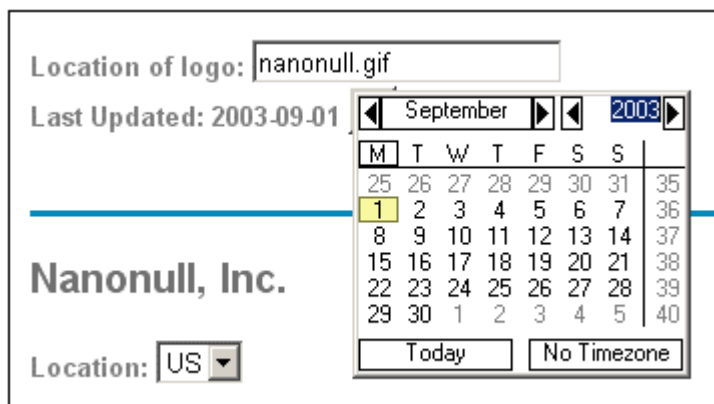


Organization Chart

Location of logo:

Last Updated: 2003-09-01 

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



Location of logo:

Last Updated: 2003-09-01

Nanonull, Inc.

Location:

September 2003

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

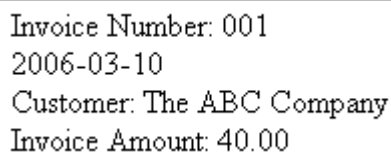
Today No Timezone

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

Texteintrag

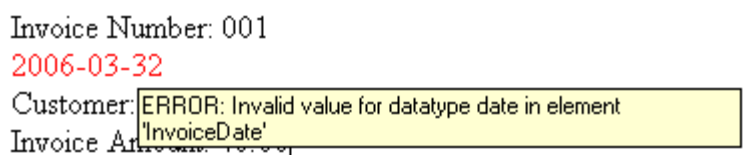
In Datumsfeldern, die keine Datumswahl haben (*siehe Screenshot*) können Sie das Datum direkt durch Eingabe des neuen Werts bearbeiten.

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



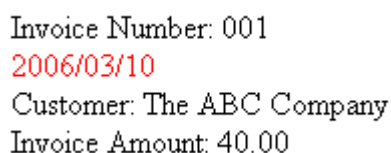
Invoice Number: 001
2006-03-10
Customer: The ABC Company
Invoice Amount: 40.00

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



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

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



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

Definieren von Entities

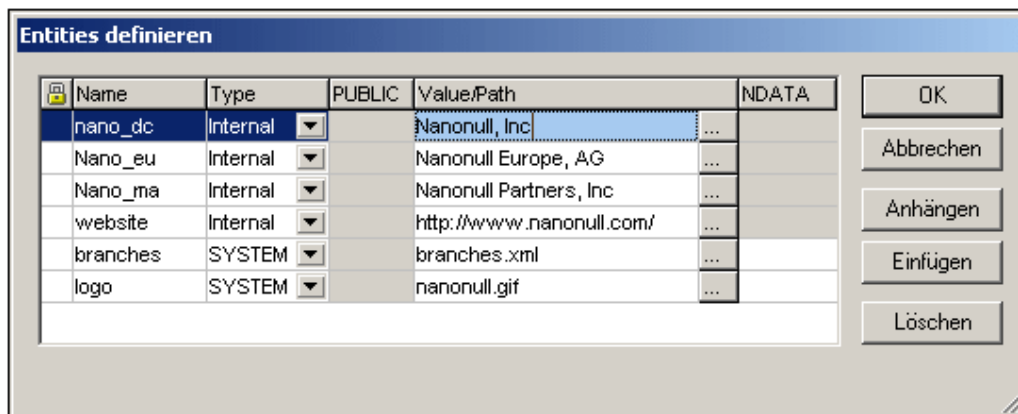
Unabhängig davon, ob Ihr Dokument auf einer DTD oder einem XML-Schema basiert, können Sie für die Verwendung in der Authentic-Ansicht Entities definieren. Diese Entities werden anschließend in der Entities-Eingabehilfe und im Kontextmenü-Untermenü von **Entity einfügen** angezeigt. Wenn Sie in der Entities-Eingabehilfe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

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

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

So definieren Sie eine Entity:

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



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

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

Dialogfeldfunktionen

Im Dialogfeld "Entities definieren" können Sie Folgendes tun:

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

Einschränkungen für Entities

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

Grafiken in der Authentic-Ansicht

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

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

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

Relative Pfade

Relative Pfade werden relativ zur SPS-Datei aufgelöst.

Verwendung von Tasten in der Authentic-Ansicht

Eingabetaste

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

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

Verwendung der Tastatur

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

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

Kapitel 15

Authentic-Skripterstellung

15 Authentic-Skripterstellung

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

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

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

* Keine Authentic-Ansicht

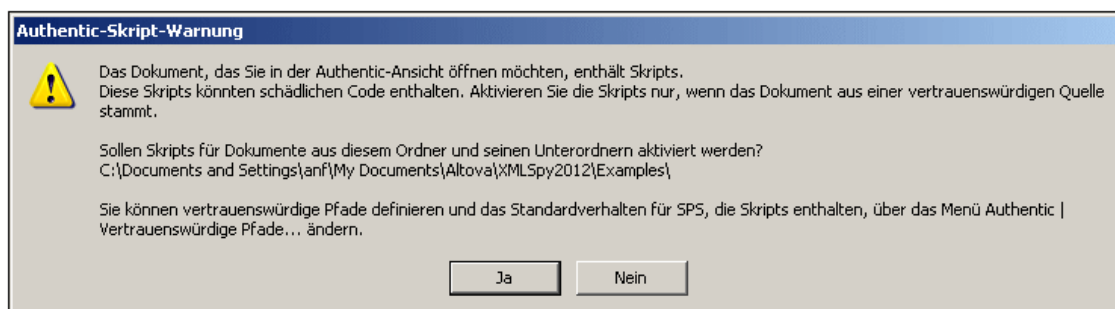
** Sowohl Trusted als auch Untrusted Version

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

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

Das Dialogfeld "Authentic-Skript-Warnung"

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



Sie können eine der folgenden Optionen wählen:

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

Eine Beschreibung des Dialogfelds "Vertrauenswürdige Pfade" finden Sie in der Benutzerreferenz unter der Beschreibung des Menübefehls **Authentic | Vertrauenswürdige Pfade**.

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

Funktionsweise der Authentic Skripterstellung

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

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

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

In diesem Abschnitt

In diesem Abschnitt wird erläutert, wie Authentic Skripts funktionieren. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- [Skript-Editor](#): Hier wird die Benutzeroberfläche beschrieben, über die ein Skript-Projekt, das die Skripts für das SPS enthält, gespeichert werden.
- [Makros](#): Hier wird beschrieben, wie Makros mit Design-Elementen, Authentic-Symbolleisten-Schaltflächen und Authentic-Kontextmenüs verknüpft werden können.
- [Event Handler](#): Hier wird beschrieben, wie Event Handler Skripts erstellt und verwendet werden
- [Skripterstellungsoptionen](#): Enthält eine Dokumentation der Optionen

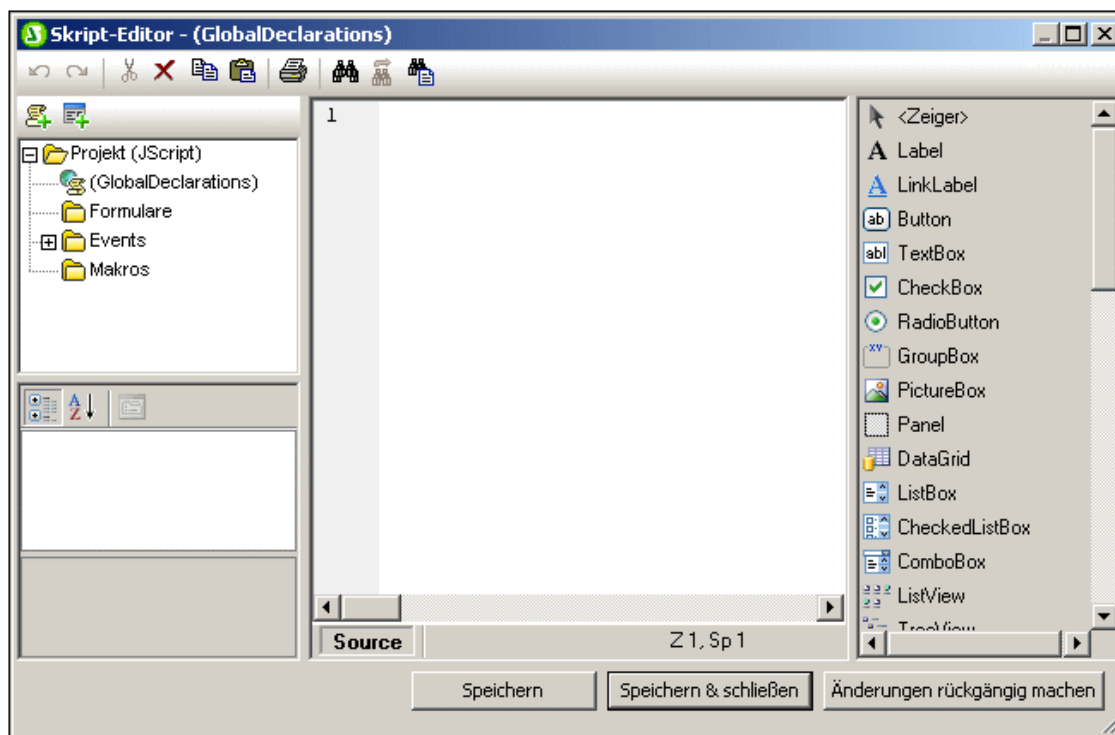
Beispieldateien

Die SPS-Beispieldateien mit Authentic-Skripterstellung befinden sich im Projektfenster von StyleVision im `Examples` Projekt im Ordner `Authentic\Scripting`.

15.1 Skript-Editor

Mit dem Skript-Editor von StyleVision (*Abbildung unten*) können Sie Skripts für Formulare, Makros und Event Handler zum SPS hinzufügen.

Der Skript-Editor kann über den Menübefehl **Authentic | Authentic Scripts bearbeiten** oder durch Auswahl des Eintrags **Authentic Skript** im [Menü zum Wechseln der Ansicht auf dem Register "Design-Ansicht"](#) aufgerufen werden.



Die Skriptsprache kann durch Rechtsklick auf den Eintrag "Project" in der Projektstruktur (*siehe Abbildung oben*) und Auswahl des Befehls **Skriptsprache** ausgewählt werden. Zur Erstellung von Skripts im Skript-Editor können Sie entweder JScript oder VBScript verwenden. Der Benutzer kann auch Klassen und Erweiterungen des Microsoft .NET Framework verwenden: z.B. zum Erstellen von Formularen.

Eine Beschreibung zum [Skript-Editor](#) und seinen Funktionen finden Sie in der [Referenz für Programmierer](#).

Jedes SPS hat ein einziges Skripting-Projekt, das im SPS gespeichert ist. Durch das **Speichern des Skripting-Projekts** wird das Skript nicht im SPS gespeichert; die Änderungen am Skripting-Projekt werden nur im Arbeitsspeicher gespeichert. Die Änderungen am Skripting-Projekt werden erst beim Speichern des SPS im SPS gespeichert.

15.2 Makros

Makros sind eigenständige Programme, die bei einer bestimmten Benutzeraktion ausgeführt werden. Makros enthalten keine Parameter, geben keinen Wert zurück und rufen keine anderen Makros direkt auf. Code, der von mehreren Makros oder Event Handlern wiederverwendet werden muss, kann jedoch in eine Funktion im Bereich "Globale Deklarationen" platziert werden.

Ein Makro, das keiner spezifischen Benutzeraktion zugewiesen wurde, wird nicht ausgeführt. Zu den gültigen Benutzeraktionen zählen eine Interaktion mit einem [bestimmten Design-Element](#), [ein Klick auf eine Symbolleisten-Schaltfläche](#) oder die Auswahl eines [benutzerdefinierten Kontextmenübefehls](#).

Die Eigenschaft `EventContext`

Ein SPS-Design-Element kann an mehreren Stellen vorkommen. Um festzulegen, in welchem Kontext ein Makro ausgeführt werden soll, hat die `AuthenticView` Schnittstelle eine Eigenschaft namens `EventContext` für den Kontext. Über die Eigenschaft `EventContext` kann ein Makro den XPath-Pfad abfragen, von dem aus das Makro gestartet wurde, den XPath-Ausdruck im aktuellen Node auswerten und Variablen, die im Geltungsbereich des Pfads definiert sind, abrufen. Nähere Informationen dazu finden Sie im Abschnitt `AuthenticView API`.

Überprüfen von Makroreferenzen

Bei der Makroerstellung könnten Makros irrtümlich umbenannt oder aus dem Skripting-Projekt gelöscht werden, obwohl sie im Design noch referenziert werden. Um dies zu verhindern bzw. zu korrigieren, überprüft der Befehl **Authentic | Makro-Referenzen überprüfen** das Design auf ungültige Referenzen. Es erscheint ein Dialogfeld, in dem der Benutzer die Referenz ersetzen oder löschen kann.

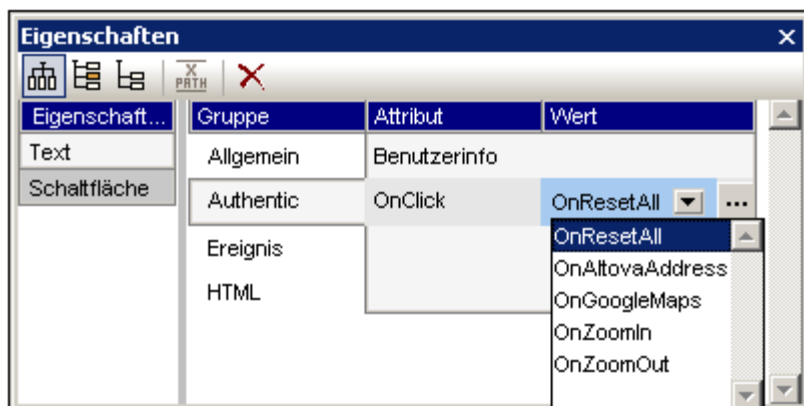
In diesem Abschnitt

In diesem Abschnitt wird beschrieben, wie Makros mit gültigen Benutzeraktionen verknüpft werden können.

- [Makros an Design-Elementen](#)
- [Makros an Symbolleisten-Schaltflächen](#)
- [Makros an Kontextmenübefehlen](#)

Macros an Design-Elementen

Makros können Design-Elementen über das Authentic-Fenster "Eigenschaften" zugewiesen werden. In der Abbildung unten wurde das Makro `OnResetAll` aus einer Liste verfügbarer Makros ausgewählt und dem `OnClick` Event einer Schaltfläche zugewiesen:



Unterschiedliche Design-Elemente unterstützen unterschiedliche Benutzeraktionen, denen Makros zugewiesen werden können. Die Tabelle unten enthält eine komplette Liste.

Benutzeraktion	Unterstütztes Design-Element	Kann Event abbrechen
OnClick	Schaltfläche, Bild	Nein
OnBeforeLinkClick	Link	Ja
OnBeforeChange	Inhalt, Restlicher Inhalt, Bearbeitungsfeld, Mehrzeiliges Bearbeitungsfeld, Kontrollkästchen, Optionsfeld, Auswahlliste	Ja
OnAfterChange	Inhalt, Restlicher Inhalt, Bearbeitungsfeld, Mehrzeiliges Bearbeitungsfeld, Kontrollkästchen, Optionsfeld, Auswahlliste	Nein
OnSetFocus	Inhalt, Restlicher Inhalt, Bearbeitungsfeld, Mehrzeiliges Bearbeitungsfeld, Kontrollkästchen, Optionsfeld, Auswahlliste	Nein
OnKillFocus	Inhalt, Restlicher Inhalt, Bearbeitungsfeld, Mehrzeiliges Bearbeitungsfeld, Kontrollkästchen, Optionsfeld, Auswahlliste	Nein

Makros, die diesen Benutzeraktionen zugewiesen wurden, werden nur bei Aufruf durch den Benutzer aufgerufen. Sie werden nicht bei einer Rückgängig-Aktion aufgerufen oder wenn die Aktion von außerhalb der Authentic-Ansicht kommt (z.B. von einem COM API-Aufruf).

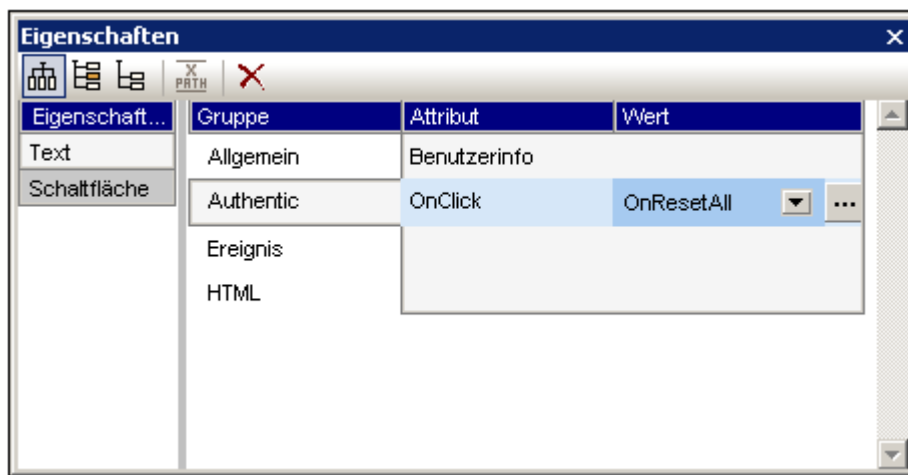
Wenn ein Makro dazu verwendet werden soll, um die weitere Event-Verarbeitung abzubrechen, muss der Benutzer an der erforderlichen Stelle im Makro einen COM API-Aufruf `AuthenticView.DoNotPerformStandardAction()` platzieren. Dieser Aufruf kann z.B. in das Makro `BeforeChangeToEU` (*Liste oben*), das einem `OnBeforeChange` Event für ein Optionsfeld zugewiesen wurde, gesetzt werden.

```
if ( !confirm( "Are you sure to change address data to EU?\nAll existing
address data will be deleted!") )
    AuthenticView.DoNotPerformStandardAction();
```

Damit wird verhindert, dass der Wert des Optionsfelds geändert wird, wenn der Benutzer im Bestätigungsdialogfeld auf Abbrechen geklickt hat. Sie finden dieses Beispiel in der Datei `OnChange.sps` im Ordner `Authentic\Scripting` des Projekts `Examples` im Projektfenster. Wenn Sie diese Datei öffnen, sehen Sie, wie das Makro funktioniert.

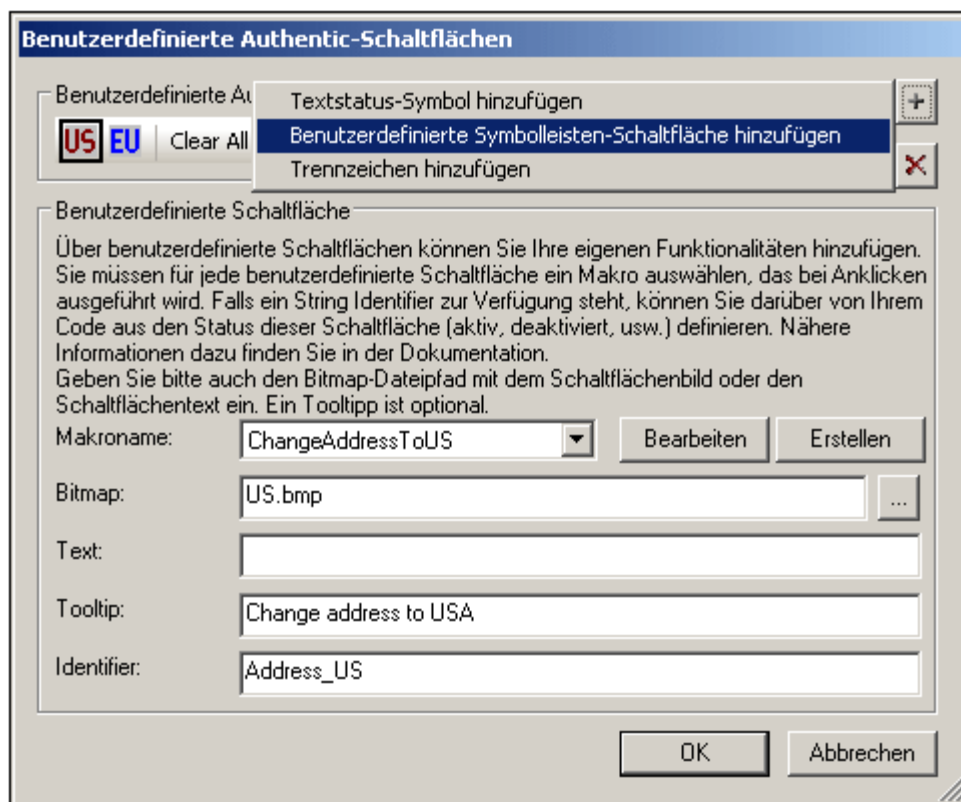
Nützliche Funktionen

Nachdem ein Makro im Fenster "Eigenschaften" (*Abbildung unten*) zugewiesen wurde, kann es schnell für die Bearbeitung im Skript-Editor aufgerufen werden. Klicken Sie dazu einfach rechts vom Namen des Makros auf die Schaltfläche **Makro bearbeiten** (siehe *Abbildung unten*).



Makros an Symbolleisten-Schaltflächen

Sie können im SPS-Design benutzerdefinierte Schaltflächen für den Aufruf bestimmter Makros definieren. Diese Schaltflächen können zur Authentic-Symbolleiste hinzugefügt werden. Um eine benutzerdefinierte Schaltfläche zur Authentic-Symbolleiste hinzuzufügen, wählen Sie zuerst den Befehl **Authentic | Benutzerdefinierte Symbolleisten-Schaltflächen** aus, klicken Sie anschließend rechts oben im Dialogfeld "Benutzerdefinierte Authentic-Schaltflächen" (siehe *Abbildung unten*) auf die Schaltfläche **Hinzufügen** und wählen Sie den Befehl **Benutzerdefinierte Symbolleisten-Schaltfläche hinzufügen**.



Benutzerdefinierte Schaltflächen erhalten die folgenden Parameter zugewiesen:

- Den Speicherpfad eines Bilds für die Schaltfläche (im Feld "Bitmap") oder Schaltflächentext (im Feld "Text").
- Wählen Sie in der Auswahlliste "Makroname" ein Makro aus der Dropdown-Liste aus. Hier sind die Makros aufgelistet, die mit dem SPS gespeichert wurden. Wenn Sie auf die Schaltfläche **Neu** klicken, wird der [Skript-Editor](#) von StyleVision in einem eigenen Fenster geöffnet. Hier können Sie ein Makro schnell und einfach erstellen und es mit dem SPS speichern. Wenn Sie auf die Schaltfläche **Bearbeiten** klicken, wird das ausgewählte Makro für die Bearbeitung im [Skript-Editor](#) geöffnet.
- Als Hilfe für den Benutzer der Authentic-Ansicht können Sie optional einen Tooltip eingeben, der angezeigt wird, wenn der Benutzer die Maus über die benutzerdefinierte Schaltfläche platziert.
- Geben Sie im Feld "Identifier" einen Textstring ein, der als Identifier für die benutzerdefinierte Schaltfläche verwendet wird. Über diesen Identifier kann ein Designer den Status der Schaltfläche im Skript-Code über die [AuthenticView API](#) steuern.

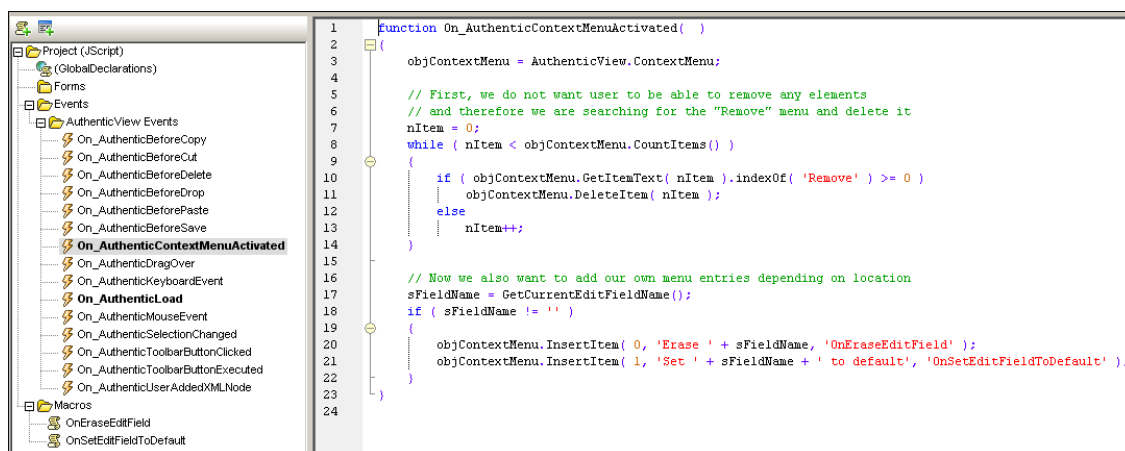
In der Abbildung oben sehen Sie, dass der benutzerdefinierten Schaltfläche **US** das Makro

`ChangeAddressToUS` zugewiesen ist. Für diese benutzerdefinierte Schaltfläche wird ein Bild mit dem Namen `US.bmp` verwendet. Zusätzlich dazu kann Text für die Schaltfläche eingegeben werden. Es wurde auch ein Tooltip eingegeben und die benutzerdefinierte Schaltfläche hat den Identifier `Address_US`. Dieses Beispiel stammt aus der Datei `ToolBarButtons.sps`, das sich im Projektfenster im Ordner `Authentic\Scripting` des Projekts `Examples` befindet.

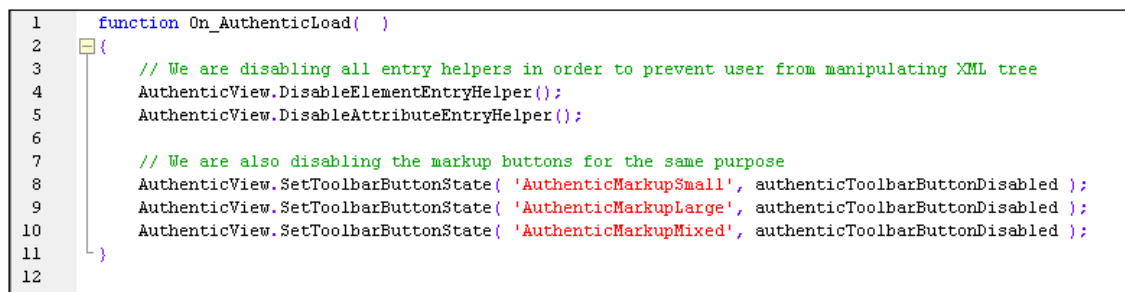
Makros an Kontextmenübefehlen

Makros können auch zum Anpassen von Authentic-Kontextmenüs verwendet werden. Dadurch können im SPS-Design benutzerdefinierte Menübefehle zu einem Kontextmenü hinzugefügt werden, die ein Makro ausführen.

Dies kann nicht über die Benutzeroberfläche erfolgen. Der SPS-Designer muss einen Event Handler für das `On_AuthenticContextMenuActivated` Event hinzufügen und das Menü anschließend über die [AuthenticView API](#) bearbeiten. Ein Beispiel dafür sehen Sie im Projektfenster, im Projekt `Examples` im Ordner `Authentic\Scripting` in der Beispieldatei `OnContextMenu.sps`. In der Abbildung unten sehen Sie das Event Handler-Skript `On_AuthenticContextMenuActivated` im Skript-Editor.



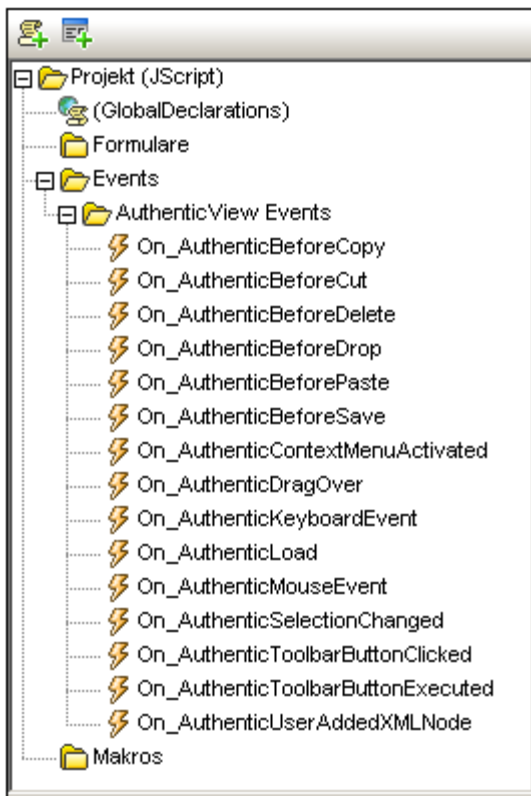
Mit diesem Event Handler für das `On_AuthenticContextMenuActivated` Event werden zwei Menübefehle erstellt und diesen Makros zugewiesen. Außerdem enthält diese Beispieldatei auch einen Event Handler für das `On-AuthenticLoad` Event, mit dem die Eingabehilfen und Markup-Schaltflächen beim Laden der Authentic-Ansicht deaktiviert werden.



15.3 Event Handler

Sie haben bei der Authentic Skripterstellung die Möglichkeit Event Handler zu definieren, um auf allgemeine - nicht aber Designelement-spezifische - Benutzeraktionen zu reagieren. Dasselbe geschieht auch, wenn Sie in externen Applikationen COM Listener für Events der Authentic-Ansicht hinzufügen. Im Unterschied zu diesen können diese Event Handler einfacher bearbeitet und im Design gespeichert werden.

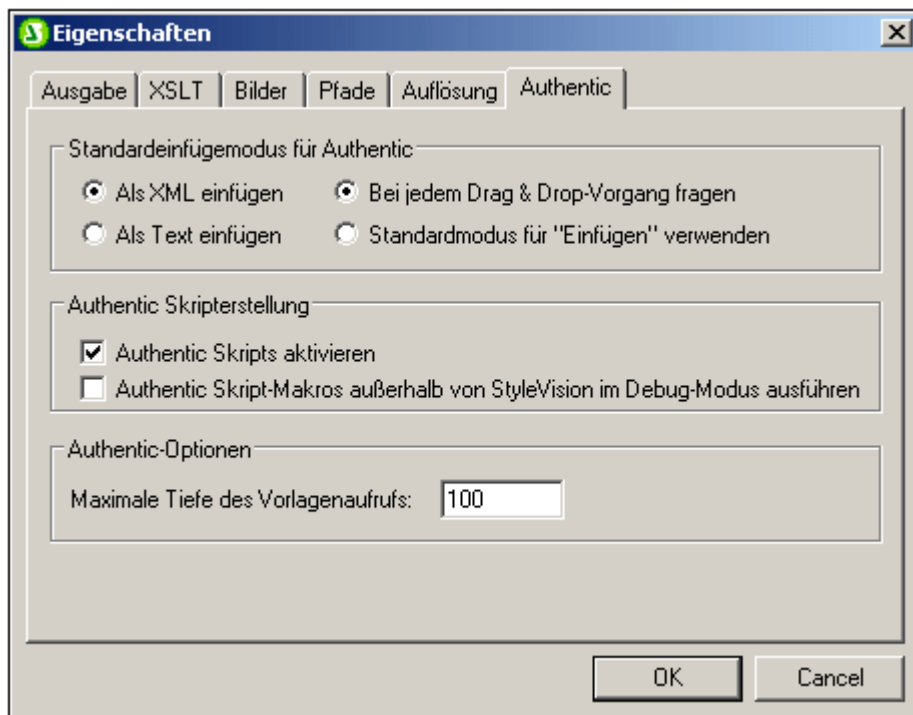
Um ein Event Handler Skript hinzuzufügen, starten Sie den Skript-Editor (**Authentic | Authentic Skripts bearbeiten**). Die verfügbaren Event Handler werden im Projektfenster unter dem Node `AuthenticView Events` angezeigt.



Doppelklicken Sie auf einen Event, um seine Skriptvorlage im Skriptfenster aufzurufen. Bearbeiten Sie das Event Handler Skript im Skriptfenster nach Ihren Wünschen. Ein Beispiel zur Verwendung von Event Handlern finden Sie im Abschnitt [Makros an Kontextmenübefehlen](#).

15.4 Optionen für die Skripterstellung

Damit interaktive Dokumente mit Authentic Scripting angezeigt werden können, muss die Authentic Skripterstellung in den Eigenschaften der SPS-Datei aktiviert sein. Wählen Sie dazu den Befehl **Datei | Eigenschaften**. Klicken Sie im Dialogfeld "Eigenschaften", das nun angezeigt wird, (Abbildung unten) auf das Register "Authentic" und aktivieren Sie die Option *Authentic Skripts aktivieren*.



Zusätzlich dazu können Sie festlegen, ob Makros außerhalb von STYLEVISION%> (also in der Authentic-Ansicht von anderen Altova-Produkten) im Debug-Modus ausgeführt werden sollen oder nicht. Diese Funktion ist hilfreich, wenn Sie ein Makro außerhalb der StyleVision-Umgebung testen und debuggen möchten.

Kapitel 16

Automatisierte Verarbeitung

16 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](#) generiert werden:

- XSLT-Dateien für die HTML- und RTF-Ausgabe.
- HTML- und RTF-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).

Anmerkung: Bei Verwendung von Datenbanken als Quelle können auf Basis der Struktur und des Inhalts der Datenbank XML-Schema- und XML-Datendateien generiert werden.

Die Generierung der endgültigen HTML- und RTF-Ausgabedateien erfolgt in einem Schritt, in der das XML-Dokument anhand eines XSLT Stylesheet in das Ausgabeformat transformiert wird.

StyleVisionBatch und AltovaXML: 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](#)), stehen zwei weitere Methoden zur Verfügung, um Ausgabedateien zu generieren:

1. mit dem [StyleVisionBatch Tool](#), das die Dateigenerierungsfunktionen von StyleVision aufruft, ohne die Benutzeroberfläche zu öffnen, können Sie eine Reihe von Input-Parametern verwenden, um verschiedene Arten von Ausgabedateien zu erzeugen. Einer der Parameter, die Sie definieren können, ist die SPS Datei selbst, von der aus alle XSLT-Stylesheets und somit alle Endausgabedateien generiert werden können. (Die Enterprise und die Professional Edition bieten mehrere Ausgabeformate.) StyleVisionBatch wird über die Befehlszeile verwendet und ermöglicht daher die Automation der Dateigenerierungsfunktionen von StyleVision. Eine Beschreibung zur Verwendung von StyleVisionBatch finden Sie im Unterabschnitt [Befehlszeilenschnittstelle: StyleVisionBatch](#).
2. mit [AltovaXML](#), einer kostenlosen Standalone-Applikation von Altova, die den Altova XML Validator, die Altova XSLT-Prozessoren (1.0 und 2.0), and den Altova XQuery 1.0 Prozessor enthält. Die XSLT-Prozessoren in AltovaXML können für die Transformation von XML in ein Ausgabeformat (durch Verarbeitung der XML-Dokumente mittels eines XSLT-Dokuments) verwendet werden. Aus diesem Grund muss zuerst die XSLT-Datei erstellt werden, damit sie AltovaXML als Input-Parameter zur Verfügung gestellt werden kann. (AltovaXML akzeptiert kein %SPS%> als Input-Parameter.) Die Vorteile der Verwendung von AltovaXML sind: (i) schnellere und weniger speicherintensive Verarbeitung als bei Verwendung von StyleVisionBatch; und (ii) AltovaXML 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 AltovaXML für Transformationen finden Sie im Unterabschnitt [AltovaXML](#).

Automation mit geplanten Aufgaben

Da sowohl [StyleVisionBatch](#) als auch [AltovaXML](#) über die Befehlszeile aufgerufen werden können, können ihre Funktionen automatisiert und nach einem Zeitplan aufgerufen werden. Nähere Informationen dazu finden Sie im Abschnitt [Anleitung zum Automatisieren der Verarbeitung](#).

16.1 Befehlszeilenschnittstelle: StyleVisionBatch

Die Dateigenerierungsfunktion von StyleVision kann über das StyleVisionBatch Dienstprogramm aufgerufen werden, das in Ihrer StyleVision Installation enthalten ist. Das Dienstprogramm `StyleVisionBatch.exe` befindet sich im StyleVision Applikationsordner. Die Syntax zum Aufrufen der StyleVision Befehle über StyleVisionBatch wird im Unterabschnitt [StyleVisionBatch Syntax](#) näher beschrieben. Bei Ausführung eines Befehls läuft StyleVision im Hintergrund (d.h. ohne dass die Benutzeroberfläche angezeigt wird), generiert die erforderlichen Ausgabedateien und wird geschlossen.

Ausgabedateien

Mit Hilfe von StyleVisionBatch können Sie eine oder mehr der folgenden Dateien generieren:

- XSLT-für-HTML (. `xslt`)-Datei anhand des angegebenen SPS
- HTML (. `html`)-Datei unter Verwendung der XML- und XSLT-Datei im angegebenen SPS oder unter Verwendung einer anderen XML- und/oder XSLT-Datei
- XSLT-für-RTF (. `xslt`)-Datei anhand des angegebenen SPS
- RTF-Datei unter Verwendung der im SPS definierten XML- und XSLT-für-RTF-Datei oder unter Verwendung einer anderen XML- und/oder XSLT-für RTF-Datei
- XML-Schema-Datei eines Datenbank-basierten SPS
- XML-Datendatei eines Datenbank-basierten SPS

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. in einem DOS-Fenster zum Verzeichnis gehen, in dem sich das StyleVisionBatch Dienstprogramm befindet, einen Befehl wie den folgenden eingeben: `StyleVisionBatch -v Test.sps -OutXSLT=Test.xslt.`, 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
StyleVisionBatch -v Test.sps -inpXSLT=EN.xslt -OutHTML=TestEN.html
StyleVisionBatch -v Test.sps -inpXSLT=DE.xslt -OutHTML=TestDE.html
StyleVisionBatch -v Test.sps -inpXSLT=FR.xslt -OutHTML=TestFR.html
```

Bei der Verarbeitung der Batch-Datei werden die Befehle ausgeführt und die Dateien werden generiert.

StyleVisionBatch Syntax

Die Syntax für das Befehlszeilen-Dienstprogramm StyleVisionBatch lautet:

```
StyleVisionBatch [ <Stylevision exe> ] [ <options> ]
```

wobei

StyleVisionBatch	das CLI-Dienstprogramm ist, das sich im StyleVision Applikationsordner befindet
<Stylevision exe>	die ausführbare StyleVision Datei ist; sie muss nur dann angegeben werden, wenn die ausführbare StyleVision Datei nicht stylevision.exe heißt oder sich nicht im selben Ordner wie StyleVisionBatch.exe befindet. Bei Angabe der Datei, muss der Name mit .exe enden.
<options>	Eine oder mehrere der unten aufgelisteten Optionen.

StyleVisionBatch Optionen

StyleVisionBatch 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-, 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* gekennzeichnet.

- **Dienstprogramm**

-help oder -?	Zeigt die Syntax der Befehlszeile an
-verbose oder -v	Zeigt die Verarbeitungsinformationen in Laufzeit an
-FOPBatFile=<file>	Definiert die FOP-Prozessor Batch-Datei (<i>Enterprise Edition</i>)

- **SPS und Parameter**

<stylesheet>	Definiert das SPS (.sps) Stylesheet
\$<paramname>=<value>	Weist einem Parameter einen Wert zu. Wenn der Wert ein Leerzeichen enthält, setzen Sie den Wert innerhalb doppelter Anführungszeichen. Z.B.: \$paramname="A value". Mehrere Parameter werden durch Leerzeichen voneinander getrennt.

- **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
-OutXSLFO=<file>	Schreibt die XSLT-für-FO-Ausgabe in die angegebene Datei (<i>nur Enterprise Edition</i>)
- OutXSLWord2007=<file>	Schreibt die XSLT-für-Word 2007+-Ausgabe in die angegebene Datei (<i>nur Enterprise Edition</i>)

- **Eingabedateien**

-InpXML=<file>	Definiert die XML-Eingabedatei
-InpXSLT=<file>	Definiert die XSLT-für-HTML-Eingabedatei
-InpXSLRTF=<file>	Definiert die XSLT-für-RTF-Eingabedatei
-InpXSLFO=<file>	Definiert die XSLT-für-FO-Eingabedatei (<i>Enterprise Edition</i>)
-InpXSLWord2007=<file>	Definiert die XSLT-für-Word 2007+-Eingabedatei (<i>Enterprise Edition</i>)

- **Ausgabedateien**

-OutHTML=<file>	Schreibt die HTML-Ausgabe in die angegebene Datei
-OutRTF=<file>	Schreibt die RTF-Ausgabe in die angegebene Datei
-OutFO=<file>	Schreibt die FO-Ausgabe in die angegebene Datei (<i>Enterprise Edition</i>)
-OutPDF=<file>	Schreibt die PDF-Ausgabe in die angegebene Datei (<i>Enterprise Edition</i>)
-OutWord2007=<file>	Schreibt die Word 2007+-Ausgabe in die angegebene Datei (<i>Enterprise Edition</i>)

- **DB-Datenausgabe**

-OutDBXML=<file>	Schreibt die anhand der Datenbank generierten XML-Daten in die angegebene Datei. Bei Datenbank-XML-Daten kann die Schemaquelle optional mit dem -DBWhere Flag definiert werden.
-OutDBSchema=<file>	Schreibt das anhand der Datenbank generierte XML-Schema in die angegebene Datei. Bei Datenbank-XML-Daten kann die Schemaquelle optional mit dem -DBWhere Flag definiert werden.

- **Zusätzliches Flag für XML-Datenbanken**

-DBWhere: <param>=<cond>	Definiert die Zelle in der XML-Datenbank, die ausgegeben werden soll. Der optionale Parameter <param> bezeichnet die Schemaquelle der DB-Zelle und <cond> ist eine einfache SQL WHERE -Klausel, die die Zelle(n) definiert, die ausgegeben werden soll(en). Wenn kein Parameter definiert wird, wird die Schemaquelle des SPS verwendet. <i>Siehe auch Anmerkung unten und Abschnitt Beispiele.</i>

Einige ergänzende Erklärungen

Im Folgenden finden Sie einige ergänzende Erklärungen zur StyleVisionBatch Syntax und der Verwendung der Befehlszeile.

- Wenn StyleVisionBatch aufgerufen wird, sieht die Datei im aktuellen Verzeichnis für StyleVision.exe nach. Wenn Ihre ausführbare StyleVision-Datei einen anderen

Namen hat oder sich in einem anderen Ordner befindet, definieren Sie die ausführbare Datei mit Hilfe des Arguments `<Stylevision.exe>`.

- Pfade können absolut oder relativ sein und sollten umgekehrte Schrägstriche verwenden.
- Optionen wird entweder ein Minus-Zeichen (z.B.: `-OutHTML`) oder ein Schrägstrich (z.B.: `/OutHTML`) vorangestellt.
- Wenn der Dateiname oder der Pfad zur Datei ein Leerzeichen enthält, sollte der gesamte Pfad innerhalb von Anführungszeichen gesetzt werden. Z.B.: `"c:\MyFiles\MyXML.xml"` oder `"c:\MyFiles\My XML.xml"`.
- Die Groß- und Kleinschreibung wird bei Befehlen, Pfaden und Ordner- oder Dateinamen nicht beachtet.
- Wenn die SPS Datei definiert wurde, wird die damit verknüpfte XML-Arbeitsdatei und das davon generierte XSLT Stylesheet zum Generieren der Ausgabe verwendet; daher ist keine XML-Eingabedatei oder XSLT-Datei erforderlich. Wenn jedoch keine SPS Datei angegeben ist, müssen eine XML-Eingabedatei und eine XSLT-Eingabedatei in Form von Optionen definiert werden. Auch wenn der SPS-Datei keine XML-Arbeitsdatei zugewiesen wurde, muss eine XML-Eingabedatei definiert werden. Zur Ausgabe anhand einer Datenbank muss ein SPS definiert werden.
- Parameterdeklarationen beziehen sich auf Parameter im XSLT Stylesheet. Bei Parameternamen und -werten muss die Groß- und Kleinschreibung beachtet werden. Wenn das SPS auf einer Datenbank basiert und einen DB-Filter hat, der Parameter verwendet, wird die anhand der DB generierte XML-Datei anhand der in der Befehlszeile definierten Parameterwerte gefiltert. Jeder Parameterdeklaration in der Befehlszeile muss ein `$` vorangestellt werden. Wenn mehrere Parameter verwendet werden, müssen diese durch Leerzeichen voneinander getrennt werden. Wenn der Wert des Parameters ein Leerzeichen enthält, muss der Wert in doppelte Anführungszeichen gesetzt werden.
- Da keine Standardausgabe definiert ist, müssen Sie die gewünschte Ausgabe festlegen. Z.B.: `OutHTML=Test.html`.
- Wenn Sie nur die Ausgabedatei definieren (keine XML-Datei oder XSLT-Datei), wird als XML-Quelldatei die im SPS definierte XML-Arbeitsdatei oder Datenbank verwendet und die erforderliche XSLT-Datei wird anhand des SPS generiert.
- Alle temporären Dateien, die erstellt werden, werden am Ende der Verarbeitung gelöscht.
- Die Option `-verbose` liefert einen ausführlichen Bericht über die einzelnen während der Verarbeitung des Befehls durchgeführten Schritte.
- Stellen Sie beim Definieren der HTML-Ausgabe sicher, dass die generierte Datei in einen Ordner platziert wird, in dem die relativen Pfade zu Bildern usw. korrekt auf die Zieldateien verweisen. Dasselbe gilt auch für Hyperlinks.
- Bei Verwendung des Flags `-DBWhere` (`-DBWhere: [<param>]=<cond>`) bezeichnet der Parameter `<param>` die Schemaquelle der DB-Zelle. Wenn der Parameter z.B. `$DBXMLFIELD` ist, so ist die Schemaquelle dem SPS bekannt, da der Parameter im SPS über einen Schlüssel mit einem bestimmten Schema verbunden wäre. Der Wert von `<cond>` ist eine SQL `WHERE`-Klausel, die die zu verwendenden Zelle(n) definiert. Normalerweise hat sie die Form `"COLNAME = ROWNUM"`, wobei `COLNAME` der Name der Spalte ist, die die XML-Datenzelle enthält und `ROWNUM` die Zeile(n) in der Spalte bezeichnet sowie durch Erweiterung die Zelle(n) in der Spalte. Siehe [Beispiele im nächsten Abschnitt](#). Wenn kein Parameter angegeben wurde wird bei SPSs mit einer einzigen Schemaquelle die Schemaquelle des SPS verwendet.
- Wenn das Flag `-DBWhere` verwendet wird und mehr als eine Zeile (oder Zelle) definiert sind, und wenn das DB-Zellenschema das Hauptschema im SPS ist und für den angegebenen Befehl ein Input aus der DB-Zelle erforderlich ist (diese Befehle sind: `-OutHTML`, `-OutRTF`, `-OutFO`, `-OutPDF`), dann wird der Befehl einmal für jede Zelle ausgeführt und jede Zelle wird in einer separaten Datei ausgegeben; wenn das DB-Zellenschema jedoch ein Sekundärschema im SPS ist, wird der Befehl nur auf die erste Zeile (Zelle) angewendet. Siehe [Beispiele im nächsten Abschnitt](#).

StyleVisionBatch Beispiele

Die Beispiele unten sind nach Ausgabedatei geordnet.

XSLT Stylesheets

XSLT Stylesheets können anhand der SPS Dateien generiert werden. Die einzig erforderliche Eingabe ist die SPS Datei.

- Die XSLT-für-HTML-Datei wird anhand des SPS generiert.
`StyleVisionBatch -v Test.sps -OutXSLT=Test.xslt`
- Die XSLT-für-RTF-Datei wird anhand des SPS generiert.
`StyleVisionBatch -v Test.sps -OutXSLRTF=Test.xslt`

HTML-Ausgabe

Die HTML-Ausgabe wird durch Transformieren einer XML-Datei mittels eines XSLT Stylesheet erzeugt. Bei der XML-Datei kann es sich um die XML-Arbeitsdatei handeln, die im SPS zugewiesen wurde oder um eine in der Befehlszeile definierte Datei. Bei der XSLT-Datei kann es sich um die anhand des SPS generierte handeln oder Sie können sie in der Befehlszeile definieren.

- XML-Arbeitsdatei im SPS, die mit Hilfe des anhand des SPS generierten XSLT Stylesheet transformiert wird.
`StyleVisionBatch -v Test.sps -OutHTML=Test.html`
- XML-Datei wird definiert und mittels des vom SPS generierten Stylesheet transformiert.
`StyleVisionBatch -v Test.sps -InpXML=External.xml -OutHTML=Test.html`
- XML-Arbeitsdatei im SPS wird mit Hilfe eines definierten XSLT Stylesheet transformiert.
`StyleVisionBatch -v Test.sps -InpXSLT=External.xslt -OutHTML=Test.html`
- Eine definierte XML-Datei wird mittels eines definierten XSLT Stylesheet transformiert.
`StyleVisionBatch -v -InpXML=External.xml -InpXSLT=External.xslt -OutHTML=Test.html`
`StyleVisionBatch -v Test.sps -InpXML=External.xml -InpXSLT=External.xslt -OutHTML=Test.html`

RTF-Ausgabe

Die RTF-Ausgabe wird durch Transformation einer XML-Datei mittels eines XSLT Stylesheet generiert. Bei der XML-Datei kann es sich um die XML-Arbeitsdatei handeln, die im SPS zugewiesen wurde, oder um eine in der Befehlszeile definierte Datei. Bei der XSLT-Datei kann es sich um die anhand des SPS generierte oder um eine in der Befehlszeile definierte handeln.

- XML-Arbeitsdatei im SPS wird mit Hilfe eines vom SPS generierten XSLT Stylesheet transformiert.
`StyleVisionBatch -v Test.sps -OutRTF=Test.rtf`
- Definierte XML-Datei wird mit Hilfe eines vom SPS generierten XSLT Stylesheet transformiert.
`StyleVisionBatch -v Test.sps -InpXML=External.xml -OutRTF=Test.rtf`
- XML-Arbeitsdatei im SPS wird mittels eines definierten XSLT Stylesheet transformiert.
`StyleVisionBatch -v Test.sps -InpXSLRTF=External.xslt -OutRTF=Test.rtf`

- Definierte XML-Datei wird mittels eines definierten XSLT Stylesheet transformiert.
`StyleVisionBatch -v -InpXML=External.xml -InpXSLT=External.xslt`
`-OutRTF=Test.rtf`
`StyleVisionBatch -v Test.sps -InpXML=External.xml`
`-InpXSLT=External.xslt -OutRTF=Test.rtf`

Word 2007+-Ausgabe

Die Word 2007+-Ausgabe wird durch Transformation einer XML-Datei mit einem XSLT-Stylesheet erzeugt. Bei der XML-Datei kann es sich um die im SPS zugewiesene XML-Arbeitsdatei handeln oder um eine in der Befehlszeile definierte Datei. Bei der XSLT-Datei kann es sich um die vom SPS generierte oder um eine in der Befehlszeile definierte handeln.

- XML-Arbeitsdatei in SPS, die mit einem vom SPS generierten XSLT-Stylesheet transformiert wird.
`StyleVisionBatch -v Test.sps -OutWord2007=Test.docx`
- Definierte XML-Datei, die mit einem vom SPS generierten XSLT-Stylesheet transformiert wird.
`StyleVisionBatch -v Test.sps -InpXML=External.xml`
`-OutWord2007=Test.docx`
- XML-Arbeitsdatei in SPS, die mit einem definierten XSLT-Stylesheet transformiert wird.
`StyleVisionBatch -v Test.sps -InpXSLWord2007=External.xslt`
`-OutWord2007=Test.docx`
- Definierte XML-Datei, die mit einem definierten XSLT Stylesheet transformiert wird.
`StyleVisionBatch -v -InpXML=External.xml -InpXSLT=External.xslt`
`-OutWord2007=Test.docx`
`StyleVisionBatch -v Test.sps -InpXML=External.xml`
`-InpXSLT=External.xslt -OutWord2007=Test.docx`

DB-Daten

XML-, XML-Schema, XSLT- und Ausgabedateien können anhand eines DB-basierten SPS generiert werden. Für die jeweiligen Ausgabedateien müssen die entsprechenden Switches gesetzt werden.

- XML-Schema-Datei anhand eines DB-basierten SPS.
`StyleVisionBatch -v DB.sps -OutDBSchema=DB.xsd`
- XML-Datendatei anhand eines DB-basierten SPS.
`StyleVisionBatch -v DB.sps -OutDBXML=DB.xml`
- XSLT-für-HTML-Datei anhand eines DB-basierten SPS.
`StyleVisionBatch -v DB.sps -OutXSLT=DB.xslt`
- HTML-Datei anhand eines DB-basierten SPS.
`StyleVisionBatch -v DB.sps -OutHTML=DB.html`
- Kombinationen von Datenbankdatenausgaben in einer einzigen Befehlszeile.
`StyleVisionBatch -v DB.sps -OutDBSchema=DB.xsd -OutXSLT=DB.xslt`
`StyleVisionBatch -v DB.sps -OutDBXML=DB.xml -OutXSLT=DB.xslt`
`-OutHTML=DB.html`

Im Fall von XML-Datenbanken unterscheidet man zwischen zwei Situationen: Wenn es sich bei der Schemaquelle der DB-Zelle um: (i) das Hauptschema des SPS handelt oder (ii) ein Sekundärschema des SPS. Betrachten Sie den folgenden Beispielbefehl:

```
StyleVisionBatch DBCellTest.sps -OutHTML=Out.html
-DBWhere:$DBXMLFIELD="NHE_TEST_PK < 4"
```

- Der Parameter `$DBXMLFIELD` bezeichnet das Quellschema und auch die Tabelle und Spalte, die die XML-Datenzeile enthält, da diese Informationen implizit im DB-Zellenschema enthalten sind.
- Die SQL WHERE-Klausel `"NHE_TEST_PK < 4"` definiert die ersten drei Zeilen der Spalte `NHE_TEST_PK`.
- Wenn das durch `$DBXMLFIELD` identifizierte Schema das Hauptschema von `DBCellTest.sps` ist, dann wird der Befehl `-outHTML` drei Mal verarbeitet, je ein Mal für jede der ersten drei Zeilen (Zellen), und es werden drei Ausgabedateien mit dem Namen `Out.html`, `Out(2).html` bzw. `Out(3).html` erstellt. Wenn es sich jedoch bei dem durch `$DBXMLFIELD` identifizierten Schema um ein Sekundärschema von `DBCellTest.sps`, handelt, wird nur die erste Zeile (Zelle) in der Datei `Out.html` ausgegeben.

Parameterverwendung

Bei der XSLT-Transformation können Parameter über die Befehlszeile an das XSLT Stylesheet übergeben werden.

- Parameter, die an ein XSLT Stylesheet übergeben werden, das anhand des SPS generiert wird.

```
StyleVisionBatch -v Test.sps -OutHTML=Test.html $myparam=MyText
StyleVisionBatch -v Test.sps -inpXML=External.xml -OutHTML=Test.html
$myparam="My Text"
StyleVisionBatch -v Test.sps -OutHTML=Test.html -OutFO=Test.fo
$myparam="MyText"
StyleVisionBatch -v Test.sps -OutHTML=Test.html $myparam=2006
StyleVisionBatch -v Test.sps -OutHTML=Test.html $myparam="2006"
```
- Parameter, die an ein definiertes XSLT Stylesheet übergeben werden.

```
StyleVisionBatch -v Test.sps -inpXSLT=External.xslt -OutHTML=Test.html
$myparam=MyText
StyleVisionBatch -v Test.sps -inpXSLT=External.xslt -OutHTML=Test.html
$myparam="My Text"
```

Mehrere Ausgabedateien

Wenn anhand derselben Quelle(n) mehrere Ausgabedateien erzeugt werden sollen, können diese mit Hilfe eines einzigen Befehls generiert werden.

- XSLT Stylesheets anhand von HTML-, RTF- und FO-Dateien, die anhand des SPS generiert werden.

```
StyleVisionBatch -v Test.sps -OutXSLT=Test.xslt -OutXSLRTF=Test.xslt
-OutXSLFO=Test.xslt
```
- HTML-, RTF-, FO- und PDF-Ausgabedateien, die anhand des SPS generiert werden.

```
StyleVisionBatch -v Test.sps -OutHTML=Test.html -OutRTF=Test.rtf
-OutFO=Test.fo -OutPDF=Test.pdf -FOPBatFile="FOP\fop.bat"
StyleVisionBatch -v Test.sps -OutHTML=Test.html -OutRTF=Test.rtf
-OutFO=Test.fo -OutPDF=Test.pdf -FOPBatFile="FOP\fop.bat" $myparam="My
Text"
```
- HTML-, RTF- und FO-Ausgabedateien, die anhand einer definierten XML-Datei und eines anhand des SPS generierten XSLT Stylesheet erzeugt werden.

```
StyleVisionBatch -v Test.sps -inpXML=External.xml -OutHTML=Test.html
```

```
-OutRTF=Test.rtf -OutFO=Test.fo
```

- HTML-, RTF- und FO-Ausgabedateien mit XML-Arbeitsdatei im SPS und definierter XSLT-Datei.

```
StyleVisionBatch -v Test.sps -inpXSLT=ExtHTML.xslt -OutHTML=Test.html  
-inpXSLRTF=ExtRTF.xslt -OutRTF=Test.rtf -inpXSLFO=ExtFO.xslt  
-OutFO=Test.fo
```

16.2 Verwendung von AltovaXML

AltovaXML ist ein kostenloses Produkt, das den Altova XML Validator, den XSLT 1.0- und 2.0-Prozessor und den XQuery 1.0-Prozessor enthält. Das Programm kann von der [Altova Website](#) heruntergeladen werden. AltovaXML kann über die Befehlszeile ausgeführt werden und hat Schnittstellen für COM, Java und .NET. Sie können AltovaXML daher jederzeit von diesen Umgebungen aus starten, um XML-Dokumente zu validieren, XSLT-Transformationen durchzuführen und XQuery-Dokumente auszuführen.

Am wichtigsten für Benutzer von StyleVision sind die XSLT 1.0 und 2.0-Transformationsfunktionen von AltovaXML. Normalerweise werden die Funktionen folgendermaßen verwendet:

1. Anhand eines SPS wird mit Hilfe des Befehls [Datei | Generierte Dateien speichern](#) oder mittels [StyleVisionBatch](#) ein XSLT-Stylesheet generiert. Beachten Sie, dass AltovaXML nicht wie [StyleVisionBatch](#) dazu verwendet werden kann, XSLT Stylesheets anhand eines SPS zu generieren.
2. Mit Hilfe des generierten XSLT Stylesheet werden XML-Dokumente mit AltovaXML generiert.

Mit AltovaXML können Sie HTML- und RTF-Ausgabedokumente generieren.

Vorteile von AltovaXML

Die Vorteile der Verwendung von AltovaXML sind die folgenden:

- AltovaXML ist ein schlankeres Programmpaket als [StyleVisionBatch](#) und bietet daher eine schnellere Validierung und XSLT-Transformation. Dies liegt daran, dass StyleVisionBatch den Altova Validator und XSLT-Prozessor in StyleVision verwendet und daher mehr Speicherplatz und Zeit benötigt.
- Einfache Verwendung mit der Befehlszeilenschnittstelle, der COM-, der Java- und .NET-Schnittstelle.
- Automation und Ansetzen von geplanten Aufgaben mittels Batch-Dateien und über die "geplanten Aufgaben" von Windows.

In diesem Abschnitt

Dieser Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- [XSLT 1.0 CLI-Transformationen](#) beschreibt die Syntax für Aufrufe des Altova XSLT 1.0-Prozessors von AltovaXML und enthält Beispiele für die Verwendung.
- [XSLT 2.0 CLI-Transformationen](#) beschreibt die Syntax für Aufrufe des Altova XSLT 2.0-Prozessors von AltovaXML und enthält Beispiele für die Verwendung.

Eine Beschreibung, wie AltovaXML 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](#).

Weitere ausführlichere Informationen zur Verwendung von AltovaXML, u.a. die COM-, Java- und .NET-Schnittstelle finden Sie in der [Benutzerdokumentation zu AltovaXML](#).

XSLT 1.0 CLI-Transformationen

Syntax

Die Syntax zum Aufrufen der **XSLT 1.0 Transformation** lautet:

```
AltovaXML -xslt1 xslt-Datei -in xml-Datei [-out Ausgabedatei]
[options]
```

wobei

AltovaXML	die Applikation aufruft.
-xslt1	angibt, dass für eine XSLT-Transformation der Altova XSLT 1.0 Prozessor zu verwenden ist; der Prozessor verwendet die XSLT 1.0-Datei <i>xslt-Datei</i> für die Transformation.
-in	angibt, dass die <i>xml-Datei</i> transformiert werden soll und deren Pfad angibt.
-out	die <i>Ausgabedatei</i> und deren Pfad angibt. Wenn diese Option weggelassen wird, wird die Ausgabe in die Standardausgabe geschrieben.

Es stehen die folgenden Optionen zur Verfügung:

-param	Ruft die Anweisung <code>paramname=XPath expression</code> auf. Der <code>-param</code> Switch wird vor den einzelnen globalen Parametern verwendet. Wenn in einem XPath-Ausdruck - egal ob in einem XPath-Ausdruck selbst oder in einem String Literal im Ausdruck - Leerzeichen enthalten sind, müssen doppelte Anführungszeichen verwendet werden. Siehe Beispiele.
-xslstack	Die Stack-Größe ist die maximale Tiefe, bis zu der Anweisungen ausgeführt werden. Sie kann über den Wert <code>-xslstack</code> geändert werden. Der zulässige Mindestwert beträgt 100. Die Standard-Stack-Größe ist 1000. Bei Überschreitung der Stack-Größe während einer Transformation wird ein Fehler ausgegeben.
-namedTemplate (oder -n)	Definiert die named Template am Anfang. Das Argument wird durch ein Leerzeichen von seinem Wert getrennt. Beispiel: <code>-namedTemplate MyTemplate</code>
-mode (oder -m)	Definiert den Vorlagenmodus am Anfang. Das Argument wird durch ein Leerzeichen von seinem Wert getrennt. Beispiel: <code>-mode MyMode</code>

Hinweis:

- Die XSLT-Datei muss in der Befehlszeilenanweisung definiert werden; eine XSLT-Datei, die in einer `<?xml-stylesheet?>` Verarbeitungsanweisung im XML-Dokument referenziert wird, wird nicht automatisch verwendet.
- Wenn der `-out` Parameter nicht angegeben wird, wird die Ausgabe in die Standardausgabe geschrieben.

Beispiele

- `AltovaXML -xslt1 test.xslt -in test.xml -out testout.xml`
- `AltovaXML -xslt1 test.xslt -in test.xml -out testout.xml -param date=//node/@att1`
- `AltovaXML -xslt1 test.xslt -in test.xml -out testout.xml`

- ```
-param date="//node/@att1 | //node/@att2"
```
- ```
AltovaXML -xslt1 test.xslt -in test.xml -out testout.xml  
-param date=node/@att1 -param title='stringwithoutspace'
```
- ```
AltovaXML -xslt1 test.xslt -in test.xml -out testout.xml
-param date=node/@att1 -param title="'string with spaces' "
```



## XSLT 2.0 CLI-Transformationen

### Syntax

Die Syntax zum Aufrufen der **XSLT 2.0 Transformation** lautet:

```
AltovaXML -xslt2 xslt-Datei -in xml-Datei [-out Ausgabedatei]
[options]
```

wobei

|           |                                                                                                                                                                                |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AltovaXML | die Applikation aufruft.                                                                                                                                                       |
| -xslt2    | angibt, dass für eine XSLT-Transformation der Altova XSLT 2.0 Prozessor zu verwenden ist; der Prozessor verwendet die XSLT 2.0-Datei <i>xslt-Datei</i> für die Transformation. |
| -in       | angibt, dass die <i>xml-Datei</i> transformiert werden soll und deren Pfad angibt.                                                                                             |
| -out      | die <i>Ausgabedatei</i> und deren Pfad angibt. Wenn diese Option weggelassen wird, wird die Ausgabe in die Standardausgabe geschrieben.                                        |

Es stehen die folgenden Optionen zur Verfügung:

|                             |                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -param                      | Ruft die Anweisung <code>paramname=XPath expression</code> auf. Der <code>-param</code> Switch wird vor den einzelnen globalen Parametern verwendet. Wenn in einem XPath-Ausdruck - egal ob in einem XPath-Ausdruck selbst oder in einem String Literal im Ausdruck - Leerzeichen enthalten sind, müssen doppelte Anführungszeichen verwendet werden. Siehe Beispiele |
| -xslstack                   | Die Stack-Größe ist die maximale Tiefe, bis zu der Anweisungen ausgeführt werden. Sie kann über den Wert <code>-xslstack</code> geändert werden. Der zulässige Mindestwert beträgt 100. Die Standard-Stack-Größe ist 1000. Bei Überschreitung der Stack-Größe während einer Transformation wird ein Fehler ausgegeben.                                                |
| -namedTemplate<br>(oder -n) | Definiert die named Template am Anfang. Das Argument wird durch ein Leerzeichen von seinem Wert getrennt. Beispiel: <code>-namedTemplate MyTemplate</code>                                                                                                                                                                                                            |
| -mode (oder -m)             | Definiert den Vorlagenmodus am Anfang. Das Argument wird durch ein Leerzeichen von seinem Wert getrennt. Beispiel: <code>-mode MyMode</code>                                                                                                                                                                                                                          |

### Hinweis:

- Die XSLT-Datei muss in der Befehlszeilenanweisung definiert werden; eine XSLT-Datei, die in einer `<?xml-stylesheet?>` Verarbeitungsanweisung im XML-Dokument referenziert wird, wird nicht automatisch verwendet.
- Wenn der `-out` Parameter nicht angegeben wird, wird die Ausgabe in die Standardausgabe geschrieben.
- Der XSLT 2.0-Prozessor kann zur Verarbeitung eines XSLT 1.0 Stylesheet im Rückwärtskompatibilitätsmodus verwendet werden. Die Ausgabe kann sich allerdings von der eines XSLT 1.0-Prozessors, der dasselbe XSLT 1.0 Stylesheet verarbeitet, unterscheiden.

### Beispiele

- `AltovaXML -xslt2 test.xslt -in test.xml -out testout.xml`
- `AltovaXML -xslt2 test.xslt -in test.xml -out testout.xml  
-param date="//node/@att1"`
- `AltovaXML -xslt2 test.xslt -in test.xml -out testout.xml  
-param date="//node/@att1 | //node/@att2"`
- `AltovaXML -xslt2 test.xslt -in test.xml -out testout.xml  
-param date=node/@att1 -param title='stringwithoutspace'`
- `AltovaXML -xslt2 test.xslt -in test.xml -out testout.xml  
-param date=node/@att1 -param title="'string with spaces' "`

## 16.3 Automatisieren mit geplanten Tasks (Windows XP)

Die Verarbeitung kann auf zwei Arten automatisiert werden:

- Sie können Befehle definieren, um eine Verarbeitung nach der anderen auszuführen. Damit wird die Ausführung einer Reihenfolge von Befehlen automatisiert. Dies kann ganz einfach mittels Batch-Dateien durchgeführt werden und ist im Unterabschnitt [Erstellen von Batch-Dateien](#) beschrieben.
- Sie können einen Befehl oder eine Reihe von Befehlen definieren, die zu einem bestimmten Zeitpunkt ausgeführt werden. Dies erfolgt über das Windows-Tool "Geplante Aufgaben" und ist im Abschnitt [Automatisieren mit geplanten Aufgaben](#) beschrieben.

## Erstellen von Batch-Dateien

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. Siehe [Automatisieren mit geplanten Aufgaben](#).

### Batch-Datei mit einer Folge von Befehlen

Eine Folge auszuführender Befehle wird wie folgt eingegeben:

```
@ECHO OFF
CLS
StyleVisionBatch -v Test.sps -inpXSLT=EN.xslt -OutHTML=TestEN.html
StyleVisionBatch -v Test.sps -inpXSLT=DE.xslt -OutHTML=TestDE.html
StyleVisionBatch -v Test.sps -inpXSLT=FR.xslt -OutHTML=TestFR.html
```

Bei Verarbeitung der Batch-Datei werden die Befehle ausgeführt und die Dateien generiert. Die obige Batch-Datei generiert mittels StyleVisionBatch drei HTML-Ausgabedateien, von denen jede mit einem anderen XSLT-Stylesheet generiert wird. Die Input-Datei ist die XML-Arbeitsdatei der SPS-Datei.

### Batch-Datei, die eine Ausgabedatei eines vorhergehenden Befehls verwendet

Die Batch-Datei ruft AltovaXML auf, um eine XML-Datei zu generieren und verwendet diese XML-Datei anschließend als Input für eine XSLT-Transformation mit StyleVisionBatch. (Zur Vereinfachung nehmen wir an, dass die Aufrufe von AltovaXML und StyleVisionBatch den richtigen Pfad zu den exe-Dateien haben.)

```
@ECHO OFF
CLS
AltovaXML -xslt2 Test.xslt -in Test.xml -out TestOut.xml
StyleVisionBatch -v Test.sps -inpXML=TestOut.xml -OutHTML=TestOut.html
```

Bei Verarbeitung der Batch-Datei wird vom ersten Befehl die Datei `TestOut.xml` generiert. Der zweite Befehl verwendet `TestOut.xml` als XML-Input-Datei und verarbeitet sie anhand des XSLT-für-HTML-Stylesheet, das von der SPS Datei `Test.sps` on the fly generiert wird. Die Ausgabedatei ist die HTML-Datei `TestOut.html`.

## Automatisieren mit geplanten Tasks (Windows XP)

Sie können einen StyleVisionBatch Befehl oder eine Reihe von Befehlen (Beschreibung siehe vorheriger Abschnitt) einrichten, um eine vordefinierte Aufgabe durchzuführen. Dies erfolgt über den Windows-Assistenten für geplante Tasks. Der Assistent für geplante Tasks öffnet das StyleVisionBatch Tool und führt den in der Aufgabe definierten Task durch.

Um einen geplanten Task zu erstellen, gehen Sie folgendermaßen vor:

1. Falls Sie vorhaben, mehrere StyleVisionBatch Befehle - und nicht nur einen einzigen Befehl - als geplanten Task auszuführen, sollten diese Befehle in Form einer Batch-Datei erstellt werden (siehe [Erstellen von Batch-Dateien](#)) und die Batch-Datei sollte als der auszuführende Befehl definiert werden. Wenn Sie nur einen einzigen StyleVisionBatch Befehl ausführen möchten, überspringen Sie Schritt 1 und fahren Sie fort mit Schritt 2.
2. Öffnen Sie den Windows-Assistenten für geplante Tasks (**Start | Systemsteuerung | Geplante Tasks | Geplanten Task hinzufügen**).
3. Klicken Sie auf **Weiter** um den Task zu definieren.
4. Wählen Sie im Fenster, in dem das auszuführende Programm ausgewählt wird, entweder `StyleVisionBatch.exe` (für einen einzigen StyleVisionBatch Befehl) oder eine Batch-Datei aus (die mehrere StyleVisionBatch Befehle enthält). Suchen Sie die gewünschte Datei und wählen Sie sie aus. Daraufhin erscheint das nächste Dialogfeld (*Abbildung unten*).

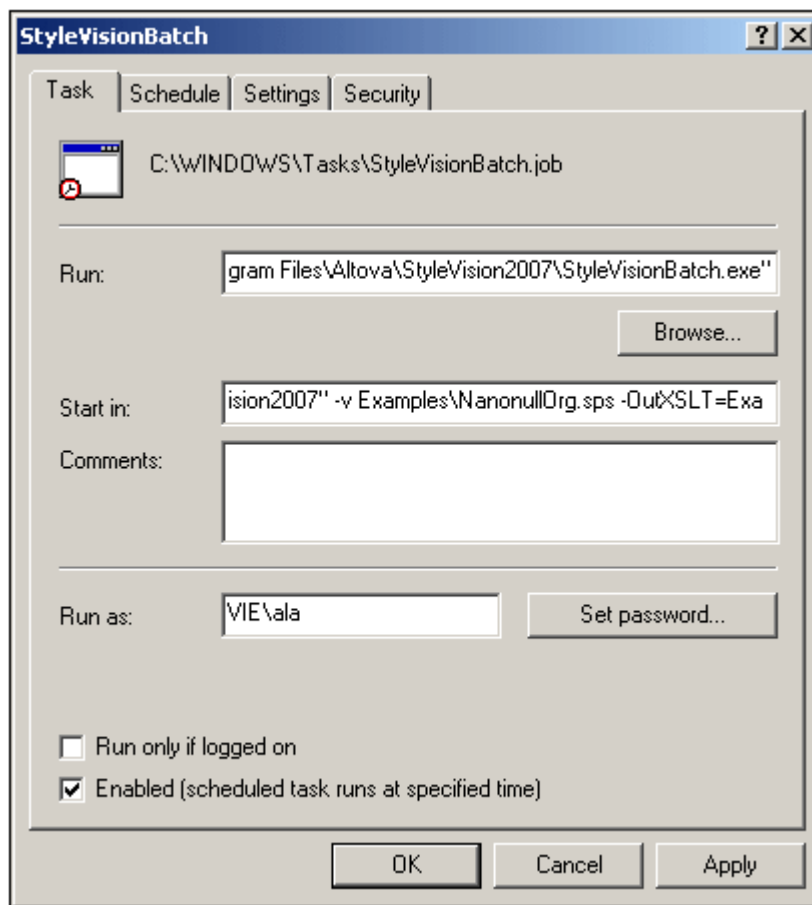


5. Geben Sie dem Task einen Namen und legen Sie fest, wie oft er ausgeführt werden soll. Klicken Sie anschließend auf **Weiter**.
6. Wählen Sie den Starttag und eine Uhrzeit für die Aufgabe aus und klicken Sie auf **Weiter**.
7. Geben Sie den entsprechenden Benutzernamen und das Passwort ein und klicken Sie auf **Weiter**.
8. Aktivieren Sie im letzten Dialogfeld (*Abbildung unten*) die Option zum Öffnen der erweiterten Eigenschaften, wenn Sie einen einzigen StyleVisionBatch-Befehl definiert und daher als auszuführendes Programm `SVBATCH%.exe` ausgewählt haben. (Der StyleVisionBatch Befehl wird im Dialogfeld "Erweiterte Eigenschaften" definiert). Klicken Sie anschließend auf **Fertig stellen**.



Wenn Sie eine Batch-Datei als auszuführendes Programm ausgewählt haben, müssen keine erweiterten Eigenschaften festgelegt werden und Sie können die Option zum Öffnen der erweiterten Eigenschaften deaktiviert lassen. In diesem Fall ist der Task nun fertig definiert.

9. Dieser Schritt ist nur dann nötig, wenn Sie einen einzigen StyleVisionBatch Befehl als Ihren Task definieren. Wenn Sie auf **Fertig stellen** klicken, und das Kontrollkästchen zum Öffnen der erweiterten Eigenschaften deaktiviert ist, wird ein Dialogfeld mit den Eigenschaften des Tasks eingeblendet (*Abbildung unten*).



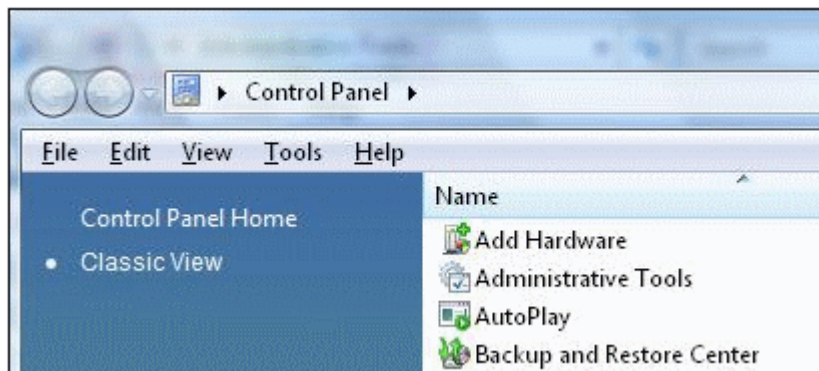
Geben Sie im Feld "Start in" (*Abbildung oben*) den gewünschten StyleVisionBatch Befehl, z.B. "C:\Program Files\Altova\StyleVision20XX" -v Examples \NanonullOrg.sps -OutXSLT=Examples\Nanol.xslt ein. Verwenden Sie Anführungszeichen, wenn Ihre Datei oder Ordnernamen Leerzeichen enthalten und verwenden Sie umgekehrte Schrägstriche zur Pfadangabe. Geben Sie auf Wunsch einen Kommentar zur Beschreibung des Tasks ein. Klicken Sie zum Fertigstellen auf **OK**.

## Automatisieren mit geplanten Tasks (Windows Vista)

Sie können einen Befehl oder eine Reihe von Befehlen (die z.B. StyleVisionBatch oder AltovaXML aufrufen) einrichten, um eine vordefinierte Aufgabe durchzuführen. Dies erfolgt über den Windows-Assistenten für geplante Tasks. Der Assistent für geplante Tasks öffnet das Tool bzw. die aufgerufene Applikation und führt den in der Aufgabe definierten Befehl aus.

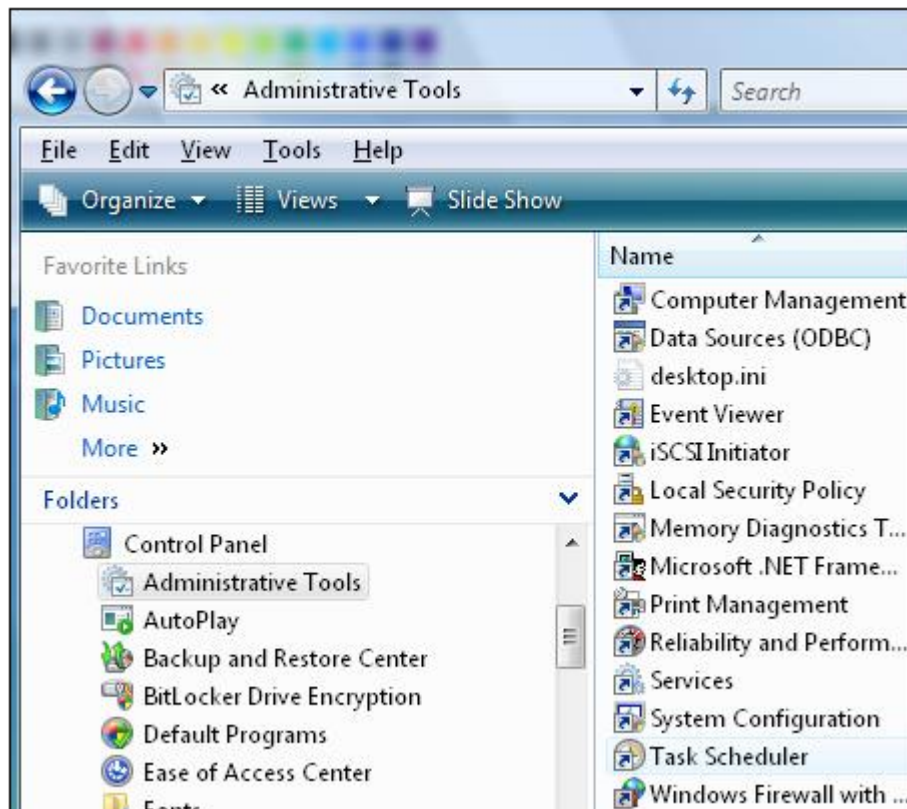
Um auf einem Windows Vista-System eine geplante Aufgabe zu erstellen, gehen Sie folgendermaßen vor. Im nachfolgenden Beispiel wird StyleVisionBatch als das Programm verwendet, das aufgerufen werden soll; um AltovaXML zu verwenden, ersetzen Sie StyleVisionBatch einfach durch AltovaXML.

1. Falls Sie vorhaben, mehrere StyleVisionBatch Befehle - und nicht nur einen einzigen Befehl - als geplante Aufgabe auszuführen, sollten diese Befehle in Form einer Batch-Datei erstellt werden (*siehe [Erstellen von Batch-Dateien](#)*) und die Batch-Datei sollte als der auszuführende Befehl definiert werden. Wenn Sie nur einen einzigen StyleVisionBatch Befehl ausführen möchten, überspringen Sie Schritt 1 (diesen Schritt) und fahren Sie fort mit Schritt 2.
2. Wählen Sie **Start | Einstellungen | Systemsteuerung**.
3. Doppelklicken Sie auf **Verwaltung** (*siehe Abbildung unten*).

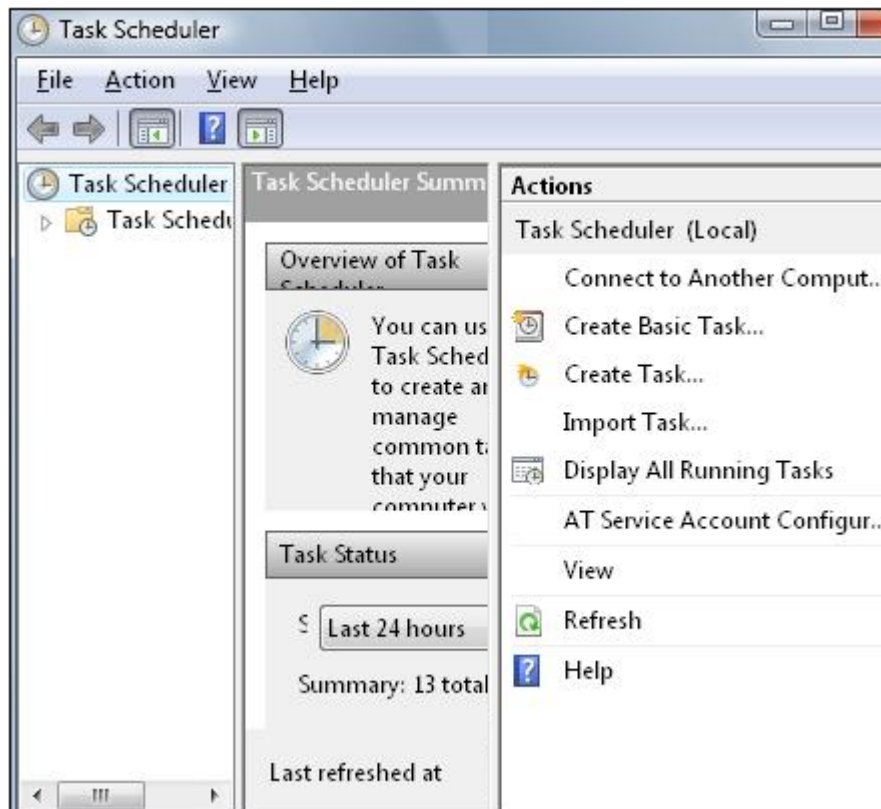


4. Doppelklicken Sie im Fenster "Verwaltung" auf **Taskplaner** (*Abbildung unten*).

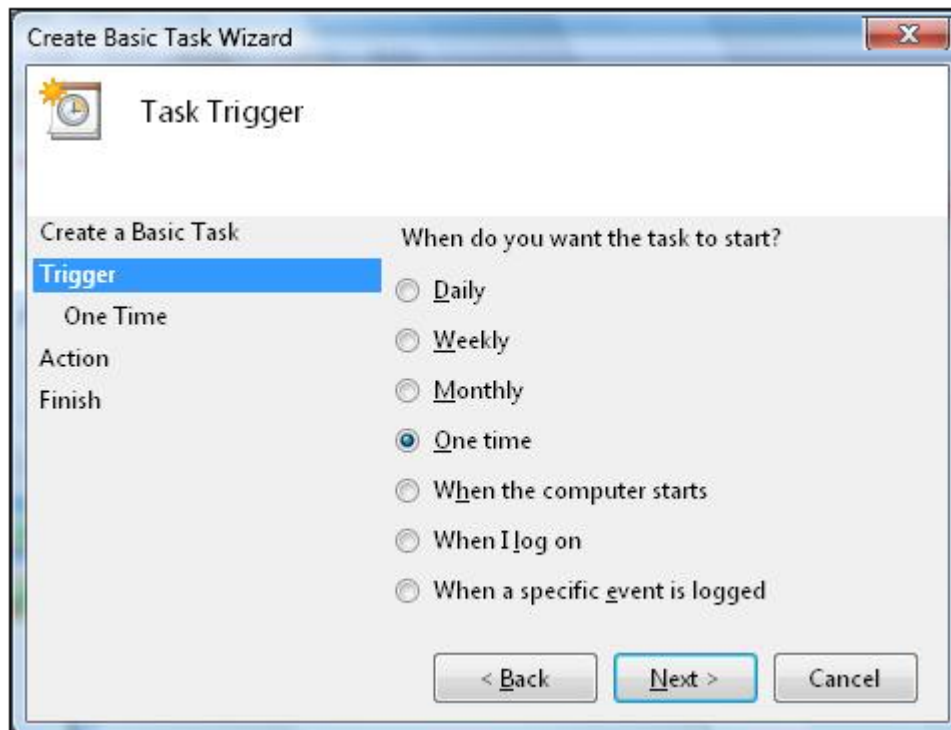




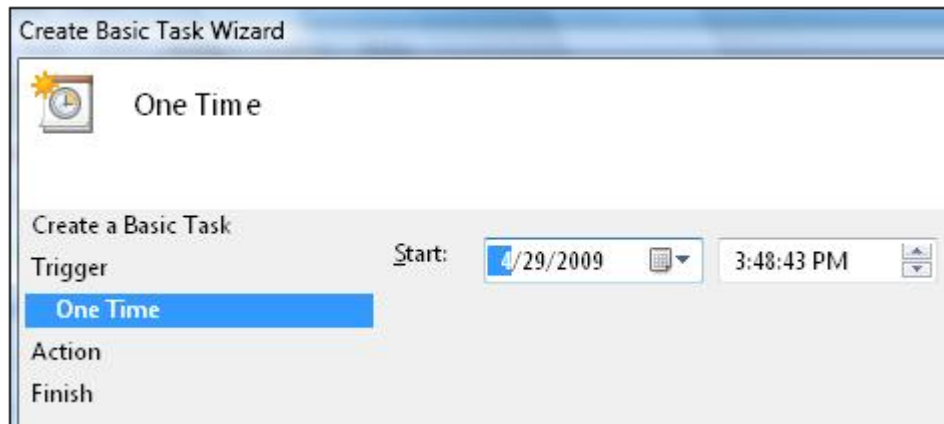
5. Daraufhin wird das Fenster "Taskplaner" (siehe Abbildung unten) geöffnet. Doppelklicken Sie auf **Create Basic Task**.



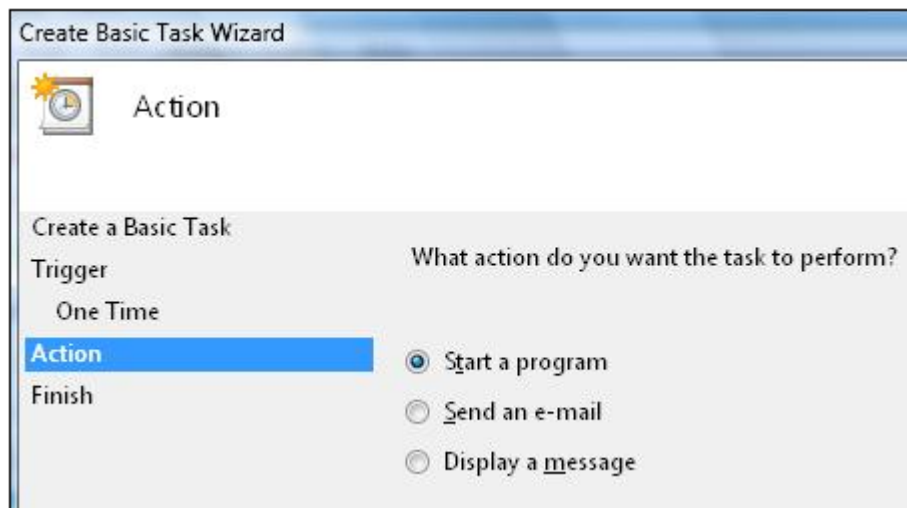
6. Wählen Sie im Fenster "Create Basic Task" (Abbildung unten) den Befehl **Trigger** und setzen Sie den Trigger (Auslöser) wie gewünscht. In der Abbildung unten wurde der Trigger auf *One time* gesetzt. Klicken Sie anschließend auf **Weiter**.



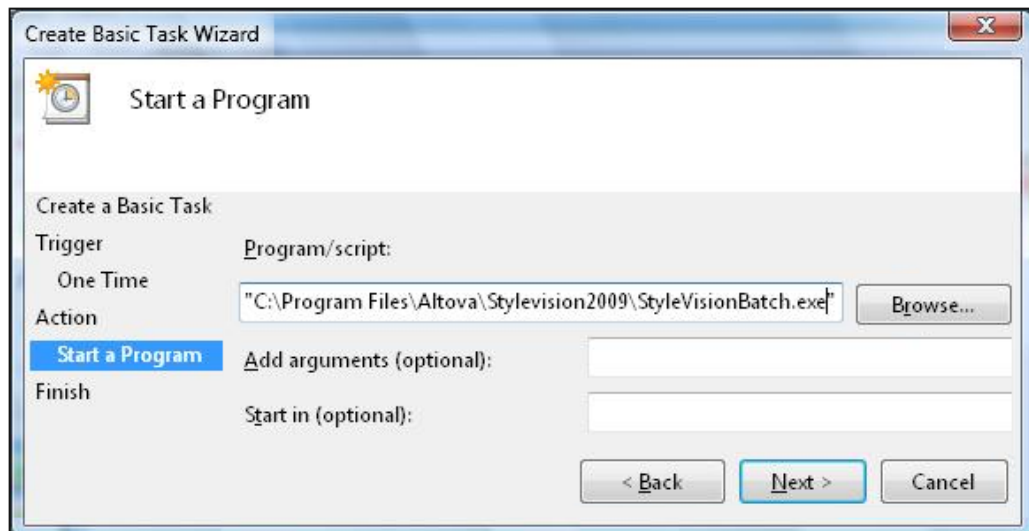
7. Wählen Sie in der Menüleiste die Option "One Time" (*Abbildung unten*) und definieren Sie die Uhrzeit. Klicken Sie anschließend auf **Weiter**.



8. Wählen Sie in der Menüleiste die Option "Action" (*Abbildung unten*), klicken Sie auf "Start a program" und anschließend auf **Weiter**.

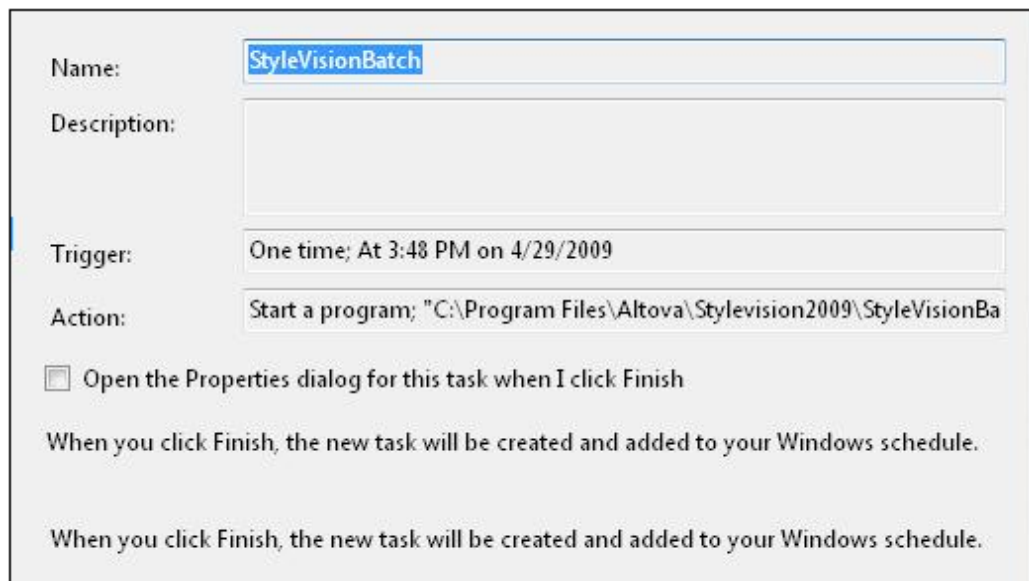


9. Wählen Sie in der Menüleiste die Option "Start a program" (*Abbildung unten*) und wählen Sie entweder `StyleVisionBatch.exe` (für einen einzigen StyleVisionBatch Befehl) oder eine Batch-Datei (die mehrere StyleVisionBatch Befehle enthält). Navigieren Sie zur gewünschten Datei und wählen Sie sie aus.



Wenn Sie StyleVisionBatch (und keine Batch-Datei) verwenden, geben Sie im Feld *Start in* (Abbildung oben) den gewünschten StyleVisionBatch Befehl, z.B. "C:\Program Files\Altova\StyleVision20XX" -v Examples\NanonullOrg.sps - OutXSLT=Examples\Nano1.xslt ein. Verwenden Sie Anführungszeichen, wenn Ihre Datei oder Ordernamen Leerzeichen enthält und verwenden Sie umgekehrte Schrägstriche zur Pfadangabe. Bei Verwendung einer Batch-Datei sollte das Textfeld *Start in* leer bleiben. Klicken Sie auf **Weiter**.

10. Weisen Sie der Aufgabe im Fenster "Finish" (Abbildung unten) einen Namen zu und erstellen Sie eine Beschreibung dazu. Wenn Sie einen einzigen StyleVisionBatch Befehl als Aufgabe ansetzen und daher als auszuführendes Programm `SVBATCH%.exe` ausgewählt haben, aktivieren Sie das Kontrollkästchen *Open the Properties dialog...*. Klicken Sie anschließend auf **Fertig stellen**.



Wenn Sie für die Aufgabe eine Batch-Datei definiert haben, können Sie das Kontrollkästchen *Open the Properties* deaktiviert lassen. In diesem Fall haben Sie die Taskplanung nun abgeschlossen.

11. Dieser Schritt ist nur erforderlich, wenn Sie einen einzigen StyleVisionBatch Befehl als Aufgabe ansetzen. Wenn Sie auf **Fertig stellen** klicken und das Kontrollkästchen "Open Properties" aktiviert ist, erscheint ein Dialogfeld, in dem die Eigenschaften der

Aufgabe angezeigt werden. Überprüfen Sie, ob Sie den richtigen Befehl für die Ausführung von StyleVisionBatch definiert haben.



## **Kapitel 17**

---

### **StyleVision in Visual Studio**

## 17 StyleVision in Visual Studio

Sie können StyleVision in die Microsoft Visual Studio IDE-Versionen 2005, 2008 und 2010 integrieren. Auf diese Weise stehen Ihnen sowohl die hoch entwickelten Funktionen zum Erstellen von SPS-Dateien als auch die professionelle Entwicklungsumgebung von Visual Studio zur Verfügung:

In diesem Abschnitt wird Folgendes beschrieben:

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



## 17.1 Installieren des StyleVision Plugin

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

- Installieren Sie Microsoft Visual Studio
- Installieren Sie StyleVision (Enterprise oder Professional Edition)
- Downloaden Sie das StyleVision Integrationspaket für Microsoft Visual Studio. Dieses Paket steht auf der StyleVision (Enterprise und Professional Edition) Download-Seite unter [www.altova.com](http://www.altova.com) zur Verfügung. (**Bitte beachten Sie:** Sie müssen das richtige Integrationspaket für Ihre jeweilige StyleVision Version (die aktuelle Version ist 2012) und die jeweilige Edition (Enterprise oder Professional) verwenden.

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

### So aktivieren Sie das Plug-In:

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

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

## 17.2 Unterschiede zur StyleVision Standalone-Version

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

### Eingabehilfen (Tool-Fenster in Visual Studio)

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

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

### StyleVision Befehle als Visual Studio-Befehle

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

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

## **Kapitel 18**

---

### **StyleVision in Eclipse**

## 18 StyleVision in Eclipse

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

Das StyleVision Plugin für die Eclipse-Plattform ist ein Eclipse Plug-In, das Ihnen gestattet, auf die Funktionalitäten von StyleVision über die Eclipse 3.5 / 3.6 / 3.7-Plattform zuzugreifen. Es steht auf Windows-Plattformen zur Verfügung. In diesem Abschnitt wird beschrieben, wie man das StyleVision Plugin für Eclipse [installiert](#) und die [StyleVision-Perspektive einrichtet](#). Anschließend stehen die Komponenten der Benutzeroberfläche von StyleVision und die Menübefehle von StyleVision über die Benutzeroberfläche von Eclipse zur Verfügung.

## 18.1 Installation des StyleVision Plugin für Eclipse

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

- StyleVision Enterprise oder Professional Edition.
- Java Runtime Environment (JRE) Version 1.5 oder höher, welches für Eclipse benötigt wird. Empfohlen wird JRE5. Nähere Informationen dazu finden Sie auf der [Eclipse Website](#).
- Eclipse-Plattform 3.5 / 3.6 / 3.7.

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

### Anmerkung zu JRE

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

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

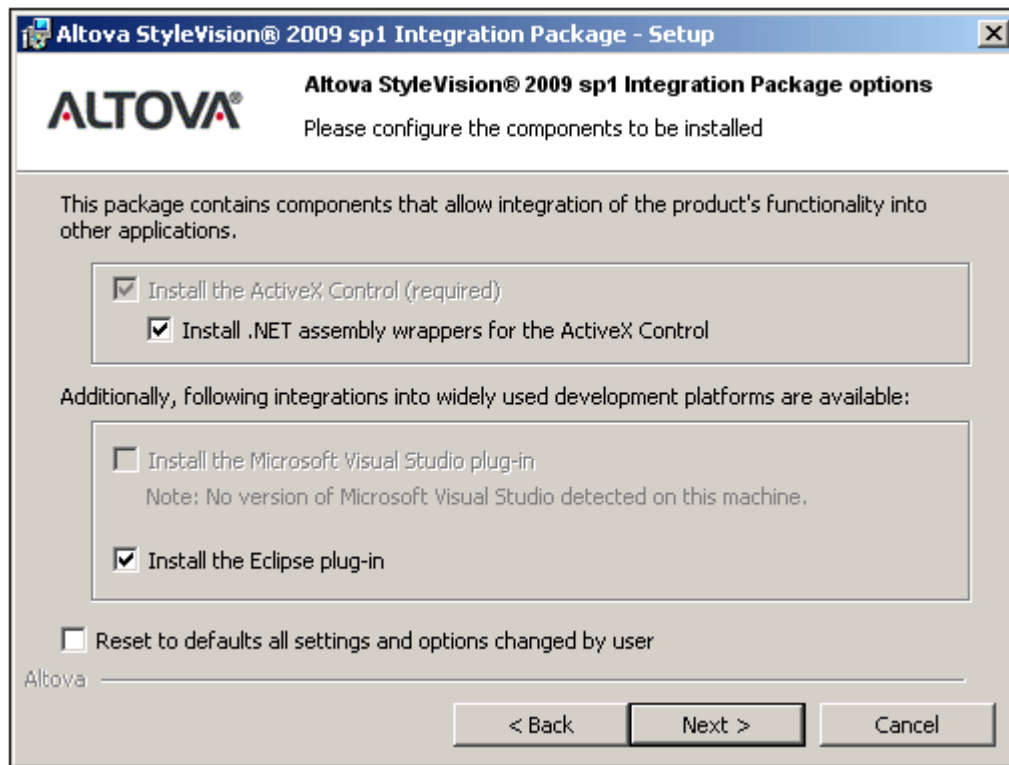
dann weist das darauf hin, dass Eclipse ein älteres JavaRTE verwendet. Eclipse verwendet zum Suchen der Datei `javaw.exe` die Umgebungsvariable `PATH`.

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

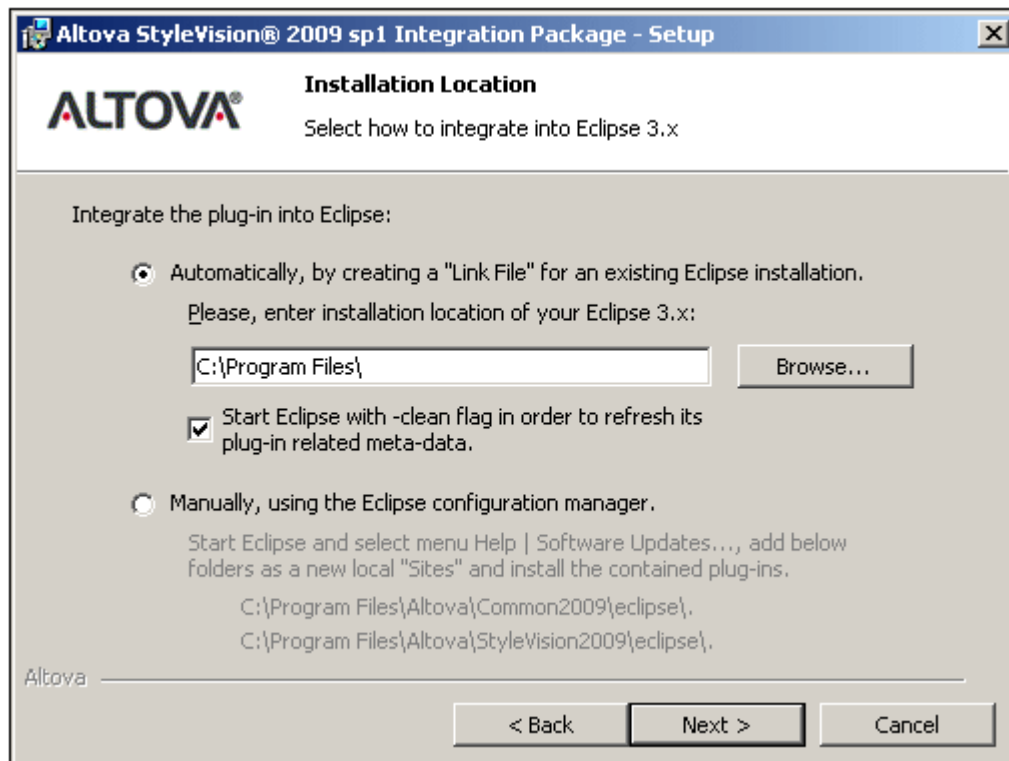
### StyleVision Integrationspaket

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

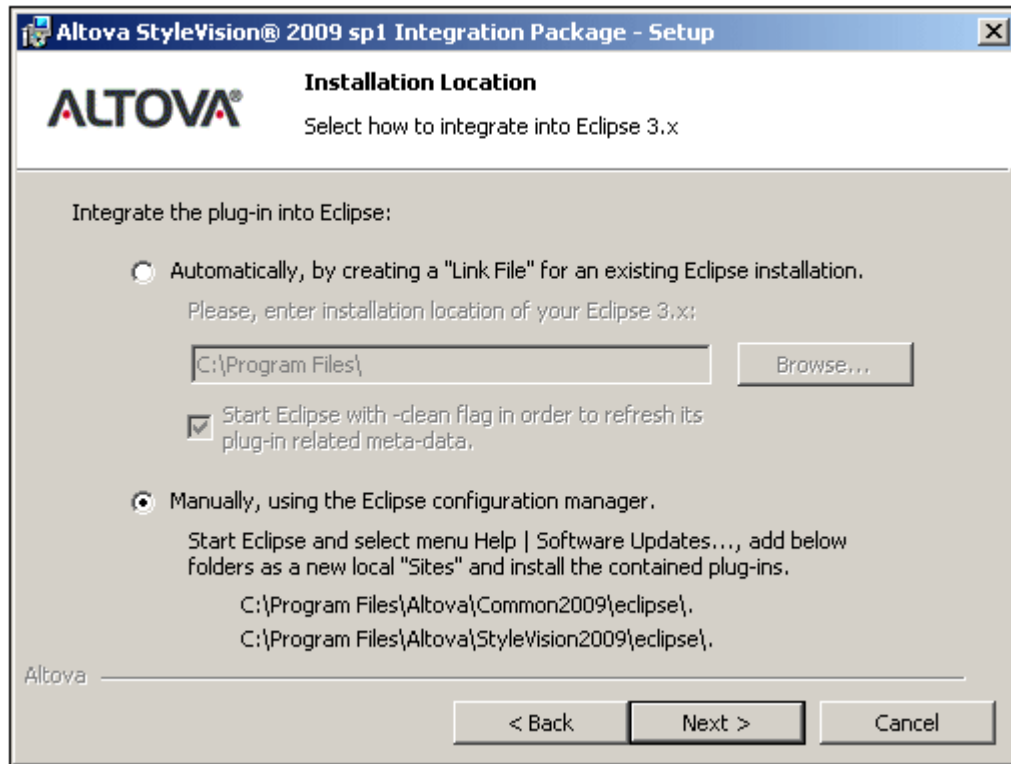
1. Stellen Sie sicher, dass StyleVision, JRE und Eclipse bereits installiert sind (*siehe oben*).
2. Laden Sie das StyleVision Integrationspaket von der [Komponenten-Download-Seite](#) der [Altova Website](#) herunter und installieren Sie es. Bei der Installation gibt es zwei wichtige Schritte, die weiter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



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



Wir empfehlen die Integration über den Installationsassistenten vorzunehmen. Aktivieren Sie dazu die Option "Automatisch" und suchen Sie nach dem Ordner, in dem sich die ausführbare Datei von Eclipse befindet (`eclipse.exe`). Klicken Sie anschließend auf **Weiter**. Wenn Sie die manuelle Integration des StyleVision Plugin für Eclipse wählen, wählen Sie die Option "Manuell" aus (*Abbildung unten*). Informationen zur manuellen Integration des Plugin in Eclipse finden Sie im folgenden Abschnitt.

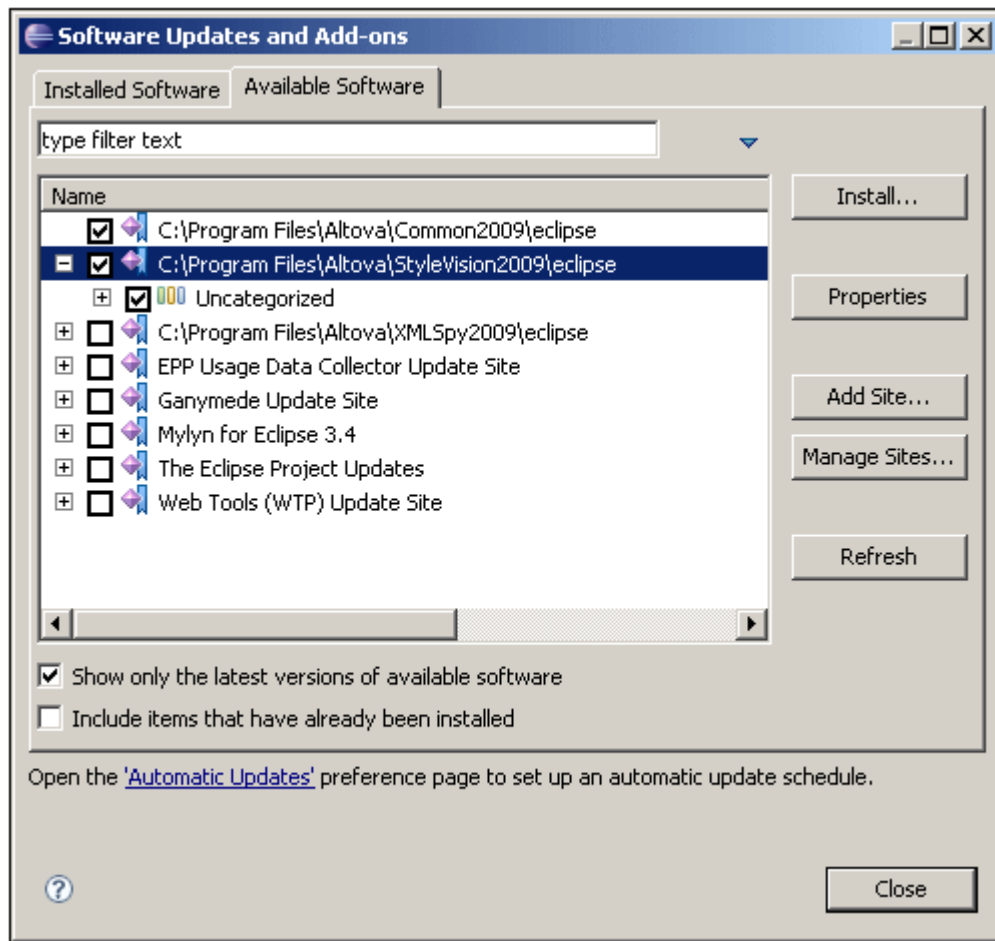


5. Stellen Sie die Installation fertig. Wenn Sie die automatische Integration wählen, wird das StyleVision Plugin für Eclipse in Eclipse integriert und steht beim nächsten Start von Eclipse bereits zur Verfügung.

### Manuelle Integration des StyleVision Plugin in Eclipse

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

1. Klicken Sie in Eclipse auf den Menübefehl **Help | Software Updates**.
2. Klicken Sie im Dialogfeld "Software Updates and Add-Ons", das daraufhin angezeigt wird, auf das Register "Available Updates" (*Abbildung unten*).
3. Klicken Sie auf die Schaltfläche **Add Site**.
4. Klicken Sie im Dialogfeld "Add Site" auf die Schaltfläche **Local**.
5. Navigieren Sie zum Ordner `c:\Program Files\Altova\Common2012\eclipse`, wählen Sie ihn aus und klicken Sie auf **OK**.
6. Wiederholen Sie die Schritte 3 bis 5 und wählen Sie diesmal den Ordner `c:\Program Files\Altova\StyleVision2012\eclipse` aus.
7. Die beiden hinzugefügten Ordner werden auf dem Register "Available Software" (*Abbildung unten*) angezeigt. Aktivieren Sie bei jedem Ordner das Kontrollkästchen auf der obersten Ebene, um die Plug-ins auszuwählen und klicken Sie auf die Schaltfläche **Install**.



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

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

#### Derzeit installierte Version

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



## 18.2 Einstiegspunkte für Stylevision in Eclipse

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

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

### StyleVision-Perspektive

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

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

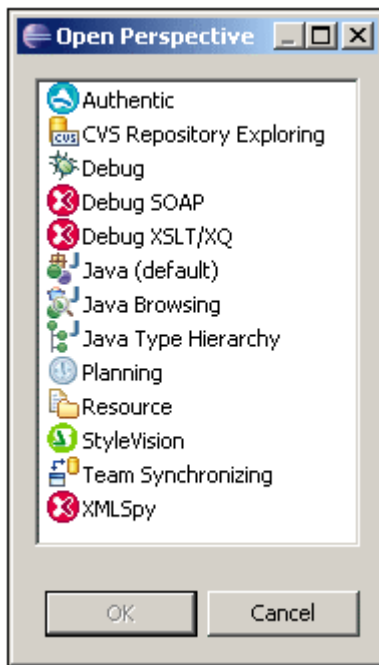
1. Die Arbeitsumgebung für die aktive Datei kann schnell gewechselt werden und
2. Sie können zwischen Dateien wechseln, ohne eine neue Entwicklungsumgebung öffnen zu müssen (die mit der Datei verknüpfte Umgebung steht in einer Perspektive zur Verfügung).

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

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

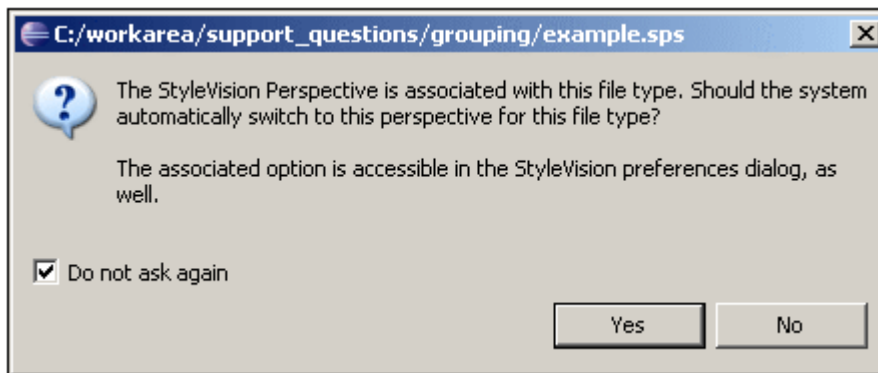
### Wechseln in die StyleVision Perspektive

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



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

Perspektiven können auch gewechselt werden, wenn eine Datei geöffnet oder zur aktiven gemacht wird. Die Perspektive der Applikation, die mit dem Dateityp einer Datei verknüpft ist, wird automatisch geöffnet, wenn diese Datei das erste Mal geöffnet wird. Vor dem Wechseln der Perspektive wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob die Standardperspektive automatisch mit diesem Dateityp verknüpft werden soll (*Abbildung unten*).

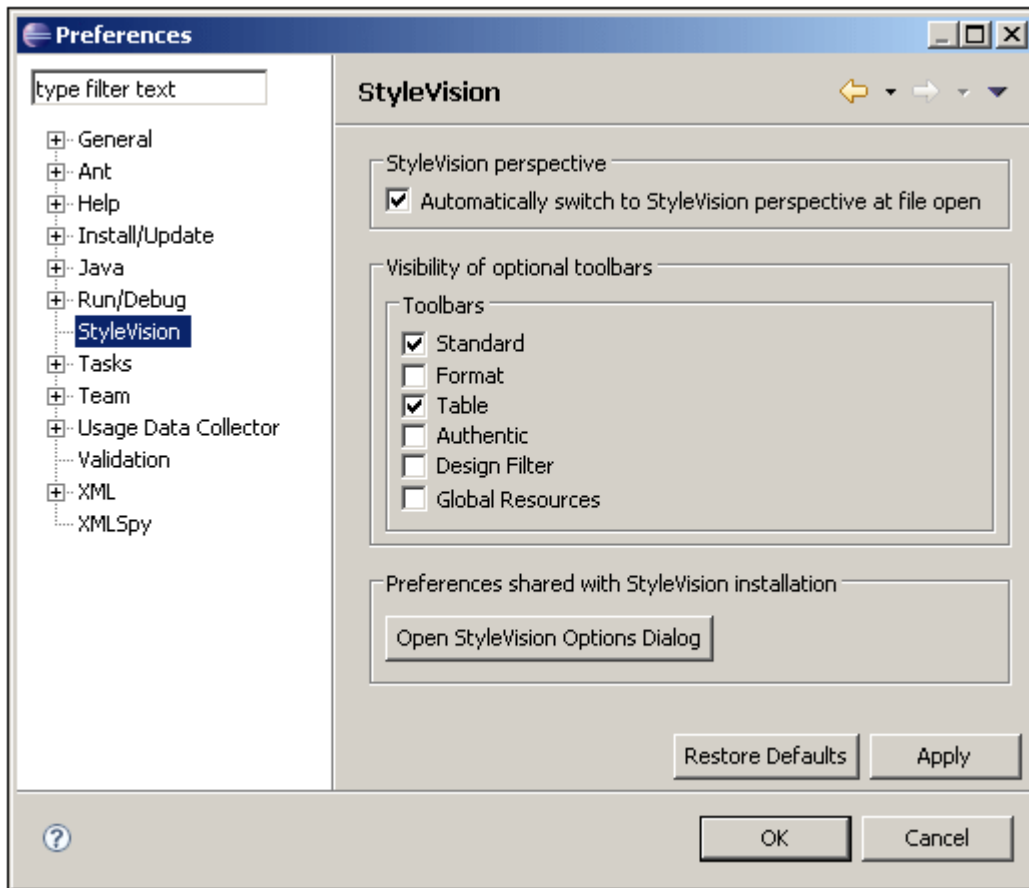


Aktivieren Sie die Option *Do Not Ask Again*, wenn die Perspektive mit dem Dateityp verknüpft werden soll, ohne dass Sie jedes Mal gefragt werden, wenn eine Datei dieses Typs geöffnet wird, und klicken Sie anschließend auf **OK**.

### Festlegen der Einstellungen für die StyleVision Perspektive

Zu den Einstellungen einer Perspektive gehören: (i) eine Einstellung, um die Perspektive automatisch zu ändern, wenn eine Datei eines verknüpften Dateityps geöffnet wird (*siehe oben*)

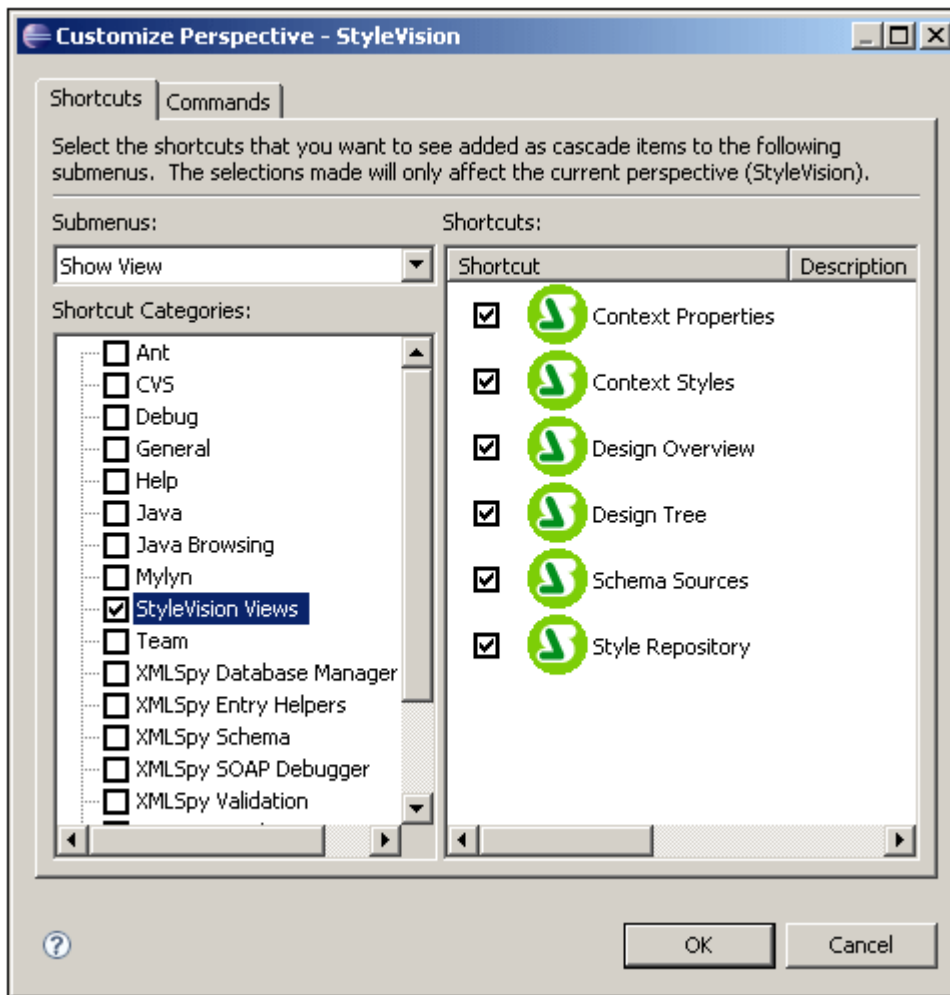
und (ii) Optionen zum Inkludieren oder Ausblenden einzelner StyleVision Symbolleisten. Wählen Sie den Menübefehl **Window | Preferences** (*Abbildung unten*), um das Dialogfeld "Preferences" aufzurufen.



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

### Anpassen der StyleVision Perspektive

Mit Hilfe der Anpassungsoptionen können Sie festlegen, welche Shortcuts und Befehle in der Perspektive enthalten sein sollen. Um dieses Dialogfeld aufzurufen (*in der Abbildung unten sehen Sie das Dialogfeld für die StyleVision-Perspektive*) machen Sie die Perspektive (in diesem Fall die StyleVision-Perspektive) aktiv und wählen Sie den Befehl **Window | Customize Perspective** aus.



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

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

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

### StyleVision Symbolleisten-Schaltflächen

Es werden automatisch zwei StyleVision-Schaltflächen in der Symbolleiste angelegt (Abbildung unten).



Diese dienen zum: (i) Öffnen der StyleVision Hilfe und (ii) Erstellen neuer StyleVision Dokumente (entsprechend Befehlen im Menü **Datei** von StyleVision).



## **Kapitel 19**

---

### **Referenz**

## 19 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 [Symbolleisten und Symbole](#) sowie eine Beschreibung, wie Sie die Symbolleistenansichten anpassen können.
- Beschreibungen der [in der Design-Ansicht verwendeten Symbole](#) und des [Dialogfelds "XPath-Ausdruck bearbeiten"](#).
- 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](#)
- [Struktur der SPS-Datei](#)
- [SPS-Datei - Komplexe Funktionen](#)
- [SPS-Datei - Darstellung](#)
- [Zusätzliche Funktionen für die SPS-Datei](#)
- [SPS-Datei und Datenbanken](#)

Informationen zur Verwendung der Befehlszeile finden Sie unter [Befehlszeilenschnittstelle: StyleVisionBatch](#).



## 19.1 Symbolleisten

Eine Reihe von StyleVision Befehlen können über die Symbolleisten aufgerufen werden. Es stehen die folgenden Symbolleisten zur Verfügung:

- Formatierung
- [Tabelle](#)
- [Authentic](#)
- [RichEdit](#)
- [Einfügen von Design-Elementen](#)
- [Designfilter](#)
- [Globale Ressourcen](#)
- [Standard](#)

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:

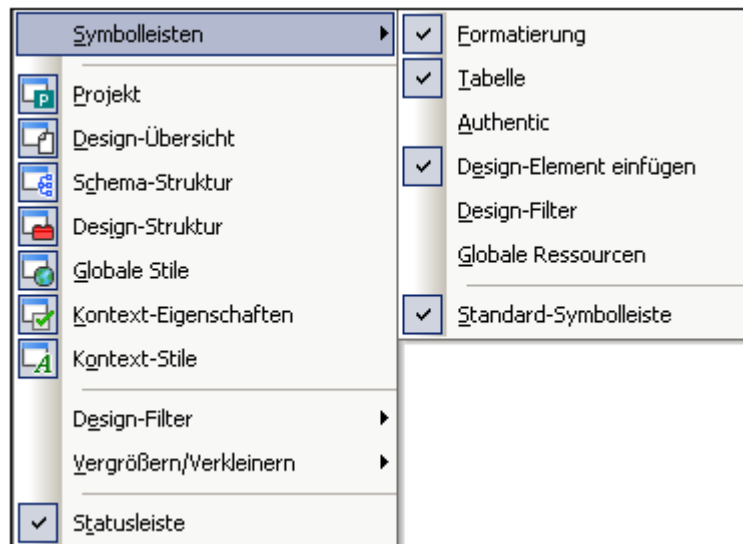
1. Ziehen Sie die Symbolleiste an ihrem Ziehpunkt (wenn sich die Symbolleiste bereits im Symbolleistenbereich befindet) oder an der Titelleiste (bei einer freischwebenden Symbolleiste).
2. Ziehen Sie die Symbolleiste in den gewünschten Symbolleistenbereich und lassen Sie die Maustaste an der gewünschten 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).

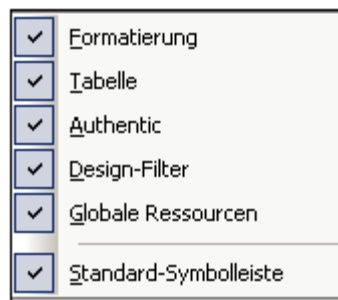
### Ein- und Ausblenden der Symbolleisten

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.

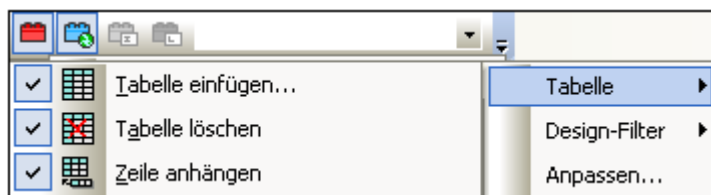


- Schalten Sie die Anzeige der einzelnen Symbolleisten im [Dialogfeld "Anpassen"](#) ([Extras | Anpassen](#)) auf dem Register "Symbolleisten" ein oder aus, indem Sie das Kontrollkästchen für die jeweilige Symbolleiste aktivieren bzw. deaktivieren. Klicken Sie anschließend auf **Schließen**, um das Dialogfeld 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*). 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"](#) 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.

## 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 (Symbolleiste "Tabelle") zu sehen.



### 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**

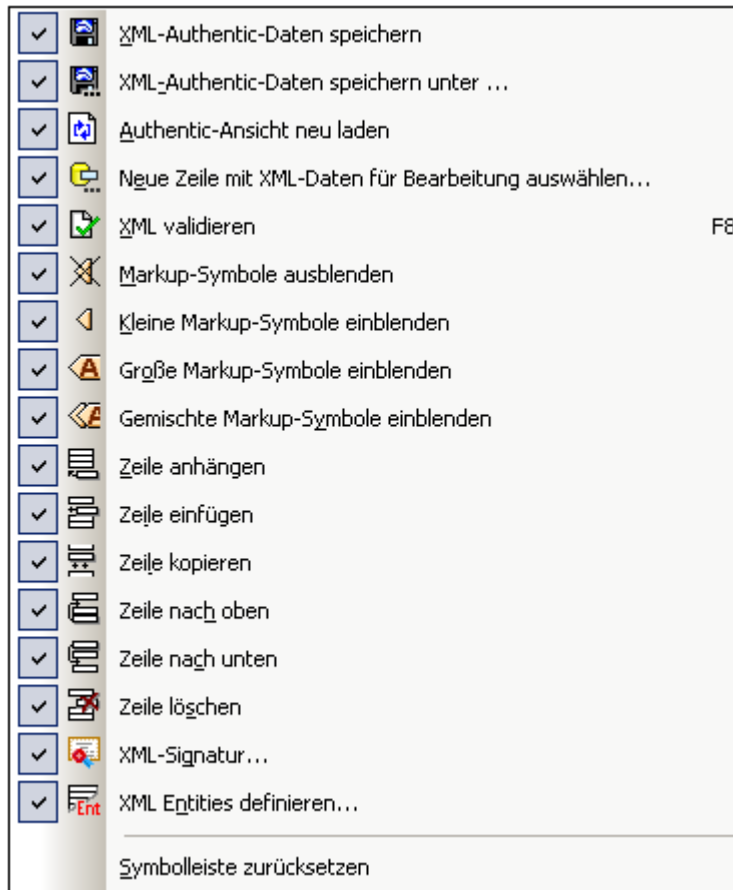
Die Zelle einer SPS Tabelle, in der der Cursor sich befindet, kann mit jeder der vier benachbarten Zellen verbunden werden. Diese Operation ist der HTML-Operation zum Zusammenfügen von Tabellenzellen ähnlich. Die Schaltflächen für diese Operationen sind **Zelle rechts verbinden/Zelle links verbinden/Zelle 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.

## Authentic

Die **Authentic-Symbolleiste** enthält Befehle zum Anpassen der Authentic-Ansicht und zum Bearbeiten von XML-Dokumenten in der Authentic-Ansicht. Diese Befehle werden in der Abbildung unten gezeigt (Menü zum [Hinzufügen und Entfernen von Schaltflächen in der Authentic-Symbolleiste](#)).

Alle diese Funktionen stehen dem Benutzer der Authentic-Ansicht zur Verfügung. Sie, als Ersteller des SPS können damit das SPS unter Verwendung von Funktionen, die dem Benutzer der Authentic-Ansicht zur Verfügung stehen, testen.



### Validieren, Speichern und Neuladen von XML-Dokumenten

Während Sie ein XML-Dokument in der Authentic-Ansicht bearbeiten, können Sie die Gültigkeit der XML-Arbeitsdatei über die Schaltfläche **XML validieren** überprüfen. Änderungen können über die Schaltfläche **XML-Authentic-Daten speichern** in der XML-Arbeitsdatei gespeichert werden. Das Dokument kann außerdem jederzeit anhand der zuletzt gespeicherten Version neu geladen werden.

### Neue Zeile mit XML-Daten für Bearbeitung auswählen

Dieser Befehl ist nur bei SPSs aktiviert, die auf einer XML-Datenbank basieren. Mit Hilfe des Befehls kann eine neue Zeile aus der XML-Spalte zur Bearbeitung in die Authentic-Ansicht geladen werden. Nähere Informationen finden Sie unter der [Beschreibung des Befehls](#).

### Markup Tags in der Authentic-Ansicht

Die Anzeige der Markup Tags in der Authentic-Ansicht kann angepasst werden. Markup Tags

können ausgeblendet werden (**Markup-Symbole ausblenden**), mit Node-Namen angezeigt werden (**Große Markup-Symbole einblenden**), ohne Node-Namen angezeigt werden (**Kleine Markup-Symbole einblenden**), oder es kann für jeden einzelnen Node eine dieser Optionen ausgewählt werden (**Gemischte Markup-Symbole einblenden**).

#### **Bearbeiten von dynamischen Tabellen in der Authentic-Ansicht**

Sie können in der Authentic-Ansicht Zeilenoperationen an dynamischen Tabellen ausführen. Zeilen können mittels der entsprechenden Symbolleisten-Schaltflächen eingefügt, angehängt, kopiert und nach oben oder unten verschoben werden (**Zeile einfügen**, **Zeile anhängen**, **Zeile kopieren**, **Zeile löschen**, **Zeile nach oben** und **Zeile nach unten**).

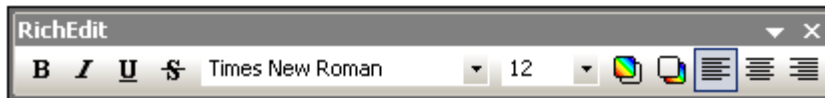
#### **Definieren von DTD Entities**

Sie können während der Bearbeitung der XML-Arbeitsdatei in der Authentic-Ansicht jederzeit Entities definieren. Wenn Sie auf die Schaltfläche **DTD Entities definieren** klicken, wird das gleichnamige Dialogfeld angezeigt. (Nähere Informationen zur Verwendung siehe [Authentic | Entities definieren](#).)

## RichEdit

Die **RichEdit-Symbolleiste** enthält Befehle, mit denen Text in der Authentic-Ansicht mit Textstileigenschaften versehen werden kann.

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



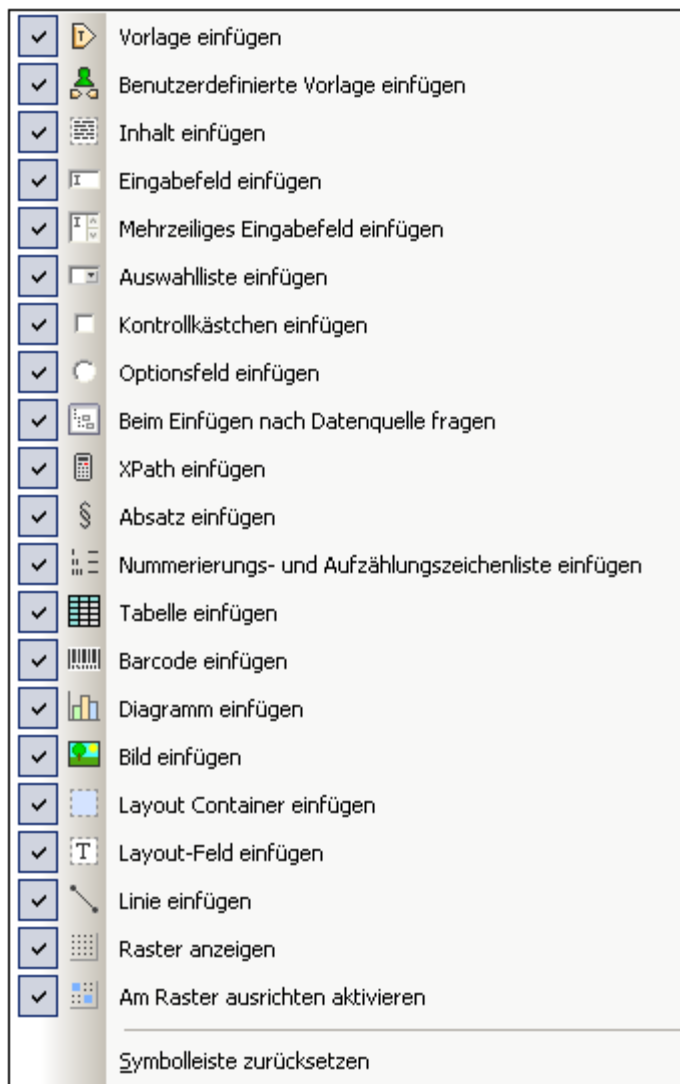
Mit Hilfe von RichEdit kann der Benutzer der Authentic-Ansicht die Schriftart (font), Schriftbreite (font-weight), Schriftverzierung (font-decoration), Schriftgröße (font-size), Hintergrundfarbe und Ausrichtung des Texts definieren. Der mit Stilen versehene Text wird in die [Tags des Stilelements eingeschlossen](#).



## Design-Elemente einfügen

Die **Symboleiste "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 Symboleisten-Schaltflächen eingefügt werden können. Die Symboleiste enthält drei Arten von Elementen:

1. [Design-Elemente](#), die Kontext-Node-sensitiv sind (die Mehrzahl der Elemente in der Symboleiste).
2. [Layout-Elemente](#), die unabhängig vom Node-Kontext sind,
3. [Ein/Aus-Befehle zur Rasteranzeige](#), die als Design-Hilfe dienen.

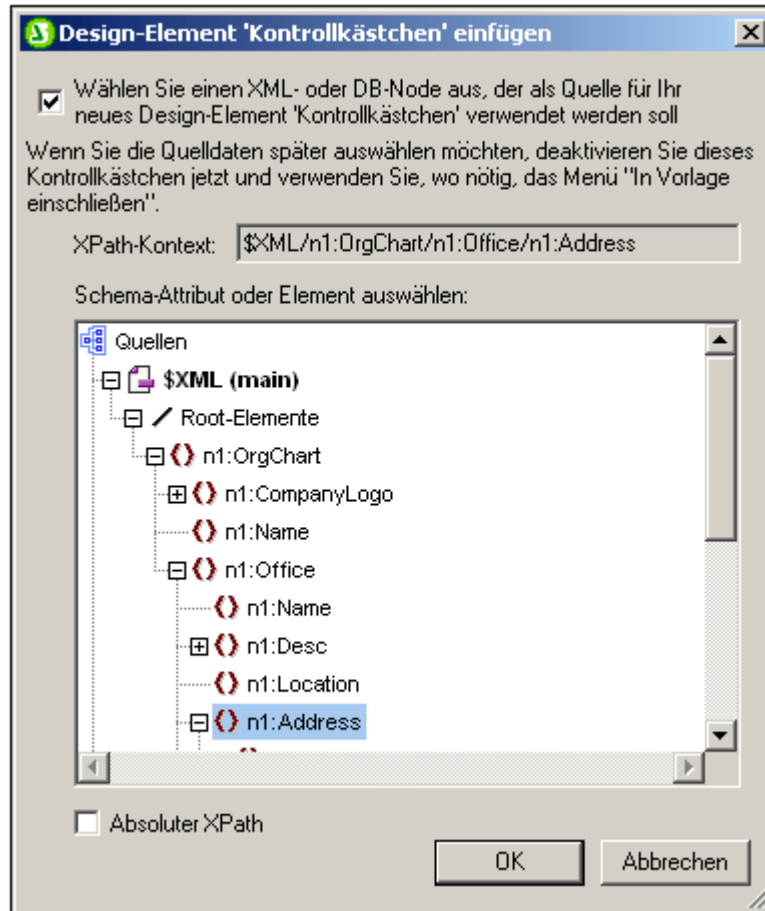


### Design-Elemente

Die Design-Elemente sind die Kontext-Node-sensitiven Elemente, die im Menü **"Einfügen"** zur Verfügung stehen. Um ein Design-Element über die entsprechende Symboleisten-Schaltfläche einzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie die Symboleisten-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 (*Abbildung unten*). 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](#) 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.

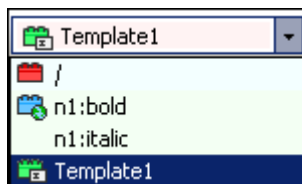
## Design-Filter

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                                                                                                                      |
|--------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|              | <b>Immer nur eine Vorlage anzeigen</b>  | Zeigt nur die ausgewählte Vorlage an. Platzieren Sie den Cursor in eine Vorlage und klicken Sie, um nur diese Vorlage anzuzeigen. |
|              | <b>Alle Vorlagentypen anzeigen</b>      | Zeigt alle Vorlagen im SPS an (Hauptvorlage, globale Vorlagen, Design-Fragmente und Layout-Vorlagen) .                            |
|              | <b>Importierte Vorlagen anzeigen</b>    | Blendet importierte Vorlagen ein und aus.                                                                                         |
|              | <b>Hauptvorlage ein-/ausblenden</b>     | Blendet die Hauptvorlage ein und aus.                                                                                             |
|              | <b>Globale Vorlagen ein-/ausblenden</b> | Blendet die globalen Vorlagen ein und aus.                                                                                        |
|              | <b>Design-Fragmente ein-/ausblenden</b> | Blendet die Design-Fragmente ein und aus.                                                                                         |

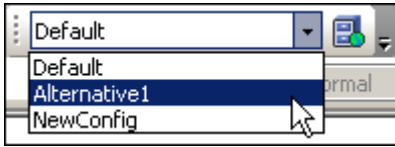
Die Design-Filter-Auswahlliste (Abbildung unten) zeigt eine Liste aller Vorlagen im SPS an.



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.

## 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"](#) 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.

## 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                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------|---------------------------------------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <a href="#">Neues Design von XML-Schema / DTD</a> | 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. |
|    | <a href="#">Öffnen</a>                            | Strg+O           | Öffnet ein vorhandenes SPS Dokument.                                                                                                                                                               |
|                                                                                     | Neu laden                                         |                  | Lädt das SPS anhand der zuletzt gespeicherten Version neu                                                                                                                                          |
|    | <a href="#">Design speichern</a>                  | Strg+S           | Speichert das aktive SPS Dokument.                                                                                                                                                                 |
|   | <a href="#">Alles speichern</a>                   | Strg +Umschalt+S | Speichert alle geöffneten SPS Dokumente.                                                                                                                                                           |
|  | <a href="#">Drucken</a>                           | Strg+P           | Druckt die Authentic-Ansicht der XML-Arbeitsdatei.                                                                                                                                                 |
|  | <a href="#">Druckvorschau</a>                     |                  | Zeigt eine Druckvorschau der Authentic-Ansicht der XML-Arbeitsdatei an.                                                                                                                            |
|  | <a href="#">Ausschneiden</a>                      | Umschalt +Entf   | Schneidet die Auswahl aus und platziert sie in die Zwischenablage.                                                                                                                                 |
|  | <a href="#">Kopieren</a>                          | Strg+C           | Kopiert die Auswahl in die Zwischenablage.                                                                                                                                                         |
|  | <a href="#">Einfügen</a>                          | Strg+P           | Fügt den Inhalt der Zwischenablage in der Cursorposition ein.                                                                                                                                      |
|  | <a href="#">Löschen</a>                           | Entf             | Löscht die Auswahl.                                                                                                                                                                                |
|  | <a href="#">Rückgängig</a>                        | Alt+ Rücktaste   | Macht eine Bearbeitung rückgängig. Aktionen können unbegrenzt rückgängig gemacht werden.                                                                                                           |
|  | <a href="#">Wiederherstellen</a>                  | Strg+Y           | Stellt eine rückgängig gemachte Bearbeitung wieder her.                                                                                                                                            |
|  | <a href="#">Absatz</a>                            |                  | Setzt die Auswahl innerhalb eines Absatzes.                                                                                                                                                        |
|  | <a href="#">Suchen</a>                            | Strg+F           | Sucht Text in der Authentic -Ansicht und den Ausgabeansichten.                                                                                                                                     |
|  | <a href="#">Weitersuchen</a>                      | F3               | Sucht die nächste Instanz des gesuchten Texts.                                                                                                                                                     |

| Schl.                                                                             | Befehl                                                          | Tastenkürzel | Beschreibung                                                 |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------|--------------|--------------------------------------------------------------|
|                                                                                   | <a href="#">Vergrößern/<br/>Verkleinern</a>                     |              | Definiert den Zoom-Faktor der Design-Ansicht.                |
|  | <b>Kleine<br/>Design<br/>Markup-<br/>Symbole<br/>einblenden</b> |              | Wechselt zur Anzeige des kleinen Markup-Formats für die Tags |
|  | <b>Große<br/>Design<br/>Markup-<br/>Symbole<br/>einblenden</b>  |              | Wechselt zur Anzeige des großen Markup-Formats für die Tags  |
|  | <a href="#">XSLT 1.0</a>                                        |              | Legt als Stylesheet Sprache XSLT 1.0 fest.                   |
|  | <a href="#">XSLT 2.0</a>                                        |              | Legt als Stylesheet Sprache XSLT 2.0 fest.                   |
|  | <a href="#">Rechtschrei<br/>bprüfung</a>                        |              | Führt am SPS Dokument eine Rechtschreibprüfung durch.        |

## 19.2 Design-Ansicht

In der Design-Ansicht werden die SPS Struktur und die Darstellungseigenschaften zugewiesen. Sie haben damit eine grafischen Darstellung Ihres Design zur Verfügung. Die Symbole, die die verschiedenen Komponenten des SPS kennzeichnen, sind wichtig, um die Struktur und das Layout des SPS zu verstehen. Diese Symbole sind im Unterabschnitt [Symbole](#) dieses Abschnitts erklärt. Ein wichtiger Mechanismus zum Aufrufen von Nodes in XML-Dokumenten ist XPath. Eine Reihe von StyleVision Funktionen verwenden XPath. Ein Dialogfeld, das bei allen diesen Funktionen zur Anwendung kommt ist das Dialogfeld "XPath-Ausdruck bearbeiten". In diesem Dialogfeld können Sie einen XPath-Ausdruck erstellen. Nähere Informationen dazu finden Sie im Unterabschnitt [Dialogfeld "XPath"](#) dieses Abschnitts.



## Symbole

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](#)
- XML-Dokument-Inhalt
- Dateneingabelemente
- Vordefinierte Formate
- XPath-Objekte
- URI-Objekte

Jeder dieser Komponententypen kann:

- mit Drag und Drop verschoben werden;
- (i) mit Hilfe der Befehle im [Menü "Ansicht"](#) (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.

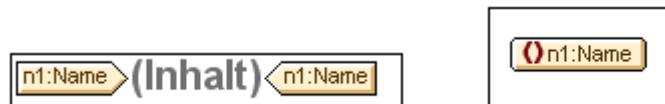
Die folgenden Arten von Nodes können dargestellt werden:

- **Dokument-Node**



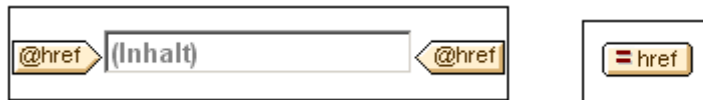
Der **Dokument-Node** (dargestellt durch `$XML`) steht für das XML-Dokument in seiner Gesamtheit. Er wird durch einen grünen `$XML` Tag gekennzeichnet, wenn die Schemaquelle mit einem XML-Dokument verknüpft ist und mit `$DB`, wenn die Schemaquelle mit einer Datenbank verknüpft ist. Der Dokument-Node 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*).

### 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-Elemente 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`, und `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.

### 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:

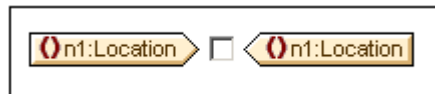
- **Eingabefelder (ein- oder mehrzeilig)**



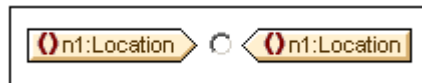
- **Auswahllisten**



- **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](#).

### Vordefinierte Formate

Verdefinierte Formate werden mit violetten Tags angezeigt, die durch Doppelklick erweitert und reduziert werden können.

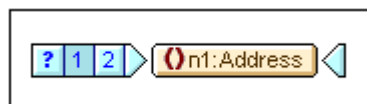


In der Abbildung oben sehen Sie Tags für das vordefinierte Format `p (para)`, auf der linken Seite in erweiterter und auf der rechten Seite in reduzierter Form. Um ein vordefiniertes Format anzuwenden, markieren Sie das Objekt, das in das vordefinierte Format eingeschlossen werden soll (durch Klicken auf die Komponente und/oder Markieren von Text) und [fügen Sie das vordefinierte Format ein](#).

### XPath-Objekte

In StyleVision werden XPath-Ausdrücke auf zwei Arten verwendet:

- **Bedingungsvarianten**



**Bedingungsvarianten** 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 Bedingungsvariantenverzweigung kann durch Anklicken ausgewählt werden. Die Nummer der ausgewählten Bedingungsvariantenverzweigung erscheint im Start-Tag markiert und die Vorlage für diese Variantenverzweigung wird (innerhalb des Start- und End-Tags der Bedingung) angezeigt. Der XPath-Ausdruck für die ausgewählte Bedingungsvariantenverzweigung 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](#).

## 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** erstellt und kann entweder 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.

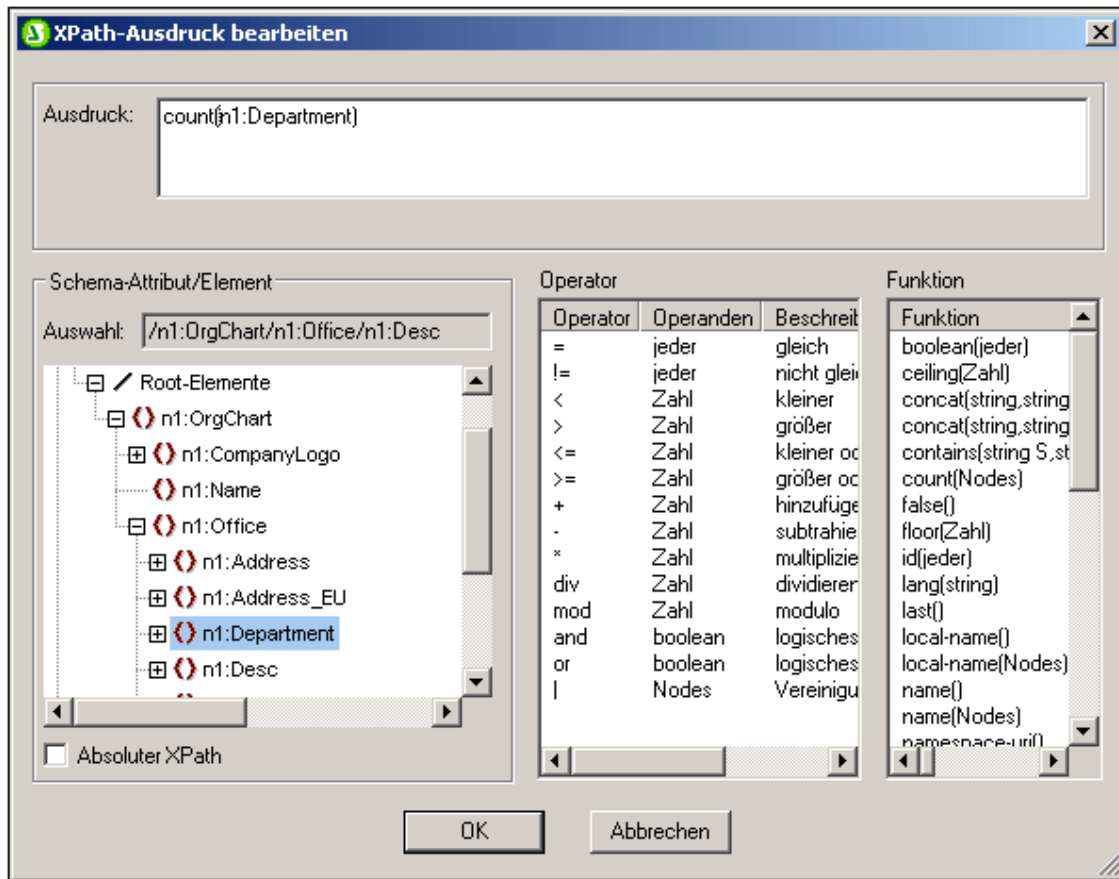
- **Links**



**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.

## XPath-Ausdruck bearbeiten

Das Dialogfeld **XPath-Ausdruck bearbeiten** (Abbildung unten) dient zum Bearbeiten und Zuweisen von XPath-Ausdrücken für:



Im Dialogfeld "XPath-Ausdruck bearbeiten" können Sie (i) einen Ausdruck über die Tastatur in das Textfeld für den Ausdruck eingeben oder (ii) Nodes, Operatoren und Funktionen durch Doppelklick auf das jeweilige Objekt in der entsprechenden Liste einfügen. XPath-Achsen werden unter Operatoren aufgelistet und XML-Schema-Konstruktorfunktionen unter Funktionen. In den Listen für die Operatoren und Funktionen werden automatisch je nach ausgewählter XSLT-Version (XPath 1.0 für XSLT 1.0 und XPath 2.0 für XSLT 2.0) die XPath 1.0 Operatoren und Funktionen bzw. die XPath 2.0 Operatoren und Funktionen angezeigt. Wenn Sie einen Teil eines XPath-Ausdrucks falsch eingeben, wird dieser rot unterstrichen angezeigt. Bei Schreibfehlern erscheint ein Popup-Fenster mit Alternativen dazu.

Über das Dialogfeld "XPath-Ausdruck bearbeiten" können Sie auf folgende Arten XPath-Ausdrücke erstellen.

- Kontext-Node**  
 Der Kontext-Node für den XPath-Ausdruck wird im Bereich "Schema Elem. oder Attr. ausw." im Textfeld "Auswahl" angezeigt. Die Bedingung, die automatische Berechnung, usw. für die der Ausdruck erstellt wird, wird an einer Stelle innerhalb dieses Kontexts eingefügt und der XPath-Ausdruck wird mit diesem Node als Kontext ausgewertet.
- Einfügen eines Node aus dem Schema in den Ausdruck**  
 Im Bereich "Attr. oder Elem. ausw." wird das gesamte Schema angezeigt. Durch

Doppelklick auf den entsprechenden Node können Sie einen Node aus dem Schema in den XPath-Ausdruck einfügen. Wenn das Kontrollkästchen "Absoluter XPath" nicht aktiviert ist, wird der Node mit einer Pfadangabe eingegeben, die relativ zum Kontext-Node ist. So wurde z.B. in der Abbildung oben, das Element `Location`, welches ein Child-Element des Elements `Office` (dem Kontextnode) ist, mit einem Pfad eingefügt, der relativ zum Kontext-Node ist (d.h. als `Location`). Wäre das Kontrollkästchen "Absoluter XPath" aktiviert, wäre der Node `Location` als `/OrgChart/Office/Location` eingefügt worden.

- **Einfügen von XPath-Operatoren**

Im Bereich "Operator auswählen" werden je nach der für das SPS gewählten XSLT-Version (XSLT 1.0 oder XSLT 2.0) automatisch XPath 1.0 oder XPath 2.0 Operatoren aufgelistet. Um einen Operator in den XPath-Ausdruck einzufügen, doppelklicken Sie auf den gewünschten Operator.

- **Namespace-Informationen**

Die Schema-Struktur im Bereich "Schema-Node auswählen" enthält einen Namespace-Eintrag. Wenn Sie diesen Eintrag erweitern, werden alle im Stylesheet deklarierten Namespaces angezeigt. Diese Informationen sind oft hilfreich, wenn Sie die Präfixe eines Namespace überprüfen, den Sie in einem XPath-Ausdruck verwenden möchten.

- **Einfügen von XPath-Funktionen**

Der Bereich "Funktion auswählen" (*Abbildung unten*) befindet sich auf der rechten Seite des Dialogfelds "XPath-Ausdruck bearbeiten". Darin werden automatisch - je nachdem, welche XSLT-Version für das SPS ausgewählt wurde (XSLT 1.0 oder XSLT 2.0) - die XPath 1.0 oder XPath 2.0 Funktionen aufgelistet. Jede Funktion wird mit ihrer Signatur aufgelistet. Wenn eine Funktion mehrere Signaturen hat, wird sie so oft aufgelistet, wie sie Signaturen hat (siehe `adjust-date-to-timezone` in der Abbildung unten). Argumente in einer Signatur sind durch Kommas getrennt und können Instanzindikatoren haben (? bedeutet eine Sequenz von null oder einem Objekt des angegebenen Typs; \* bedeutet eine Sequenz von null oder mehr Objekten des angegebenen Typs). Außerdem sehen Sie in der Funktionsliste eine kurze Beschreibung der Funktion und welcher Art das Ergebnis der Funktion ist.

| Funktion                                   |               |                                           |
|--------------------------------------------|---------------|-------------------------------------------|
| Funktion                                   | Rückgabewerte | Beschreibung                              |
| <code>boolean(jeder)</code>                | boolean       | Konvertierung von Argument in Boolesch    |
| <code>ceiling(Zahl)</code>                 | Zahl          | Kleinste Ganzzahl, die größer oder gleich |
| <code>concat(string,string)</code>         | string        | Verkettung aller String-Argumente         |
| <code>concat(string,string,...)</code>     | string        | Verkettung aller String-Argumente         |
| <code>contains(string S,string SUB)</code> | boolean       | überprüft, ob String S den String SUB e   |
| <code>count(Nodes)</code>                  | Zahl          | Anzahl der Nodes im Argument              |
| <code>false()</code>                       | boolean       | Boolescher Wert "false"                   |
| <code>floor(Zahl)</code>                   | Zahl          | größte Ganzzahl, die kleiner oder gleich  |
| <code>id(jeder)</code>                     | Nodes         | wählt Elemente nach ihrer eindeutigen     |
| <code>lang(string)</code>                  | boolean       | Ob die durch das xml:lang-Attribut defin  |

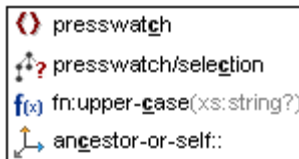
Um eine Funktion in den XPath-Ausdruck einzufügen, doppelklicken Sie auf die gewünschte Funktion.

- 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](#).

**Anmerkung:** **Java- und .NET-Erweiterungsfunktionen** werden in der Community Edition der Altova-Produkte mit Authentic-Ansicht nicht unterstützt. Sie benötigen dazu die Enterprise Edition dieser Produkte.

### Eingabeoptionen für XPath-Ausdrücke

Wenn Sie einen Ausdruck in das Textfeld "Ausdruck" eingeben, werden die verfügbaren Optionen in einem Popup-Fenster angezeigt (*Abbildung unten*).



Zur Auswahl stehen Elemente (wie z.B. in der Abbildung oben `presswatch`), untergeordnete Nodes (`presswatch/selection` in der Abbildung oben), XPath-Funktionen (`fn:upper-case` oben) und XPath-Achsen (`ancestor-or-self` oben). Die Liste der Variablenoptionen wird während der Eingabe von Ausdrücken in das Textfeld eingeschränkt.

### Das Kontrollkästchen "Andernfalls"

Das Kontrollkästchen "Andernfalls" unterhalb des Eingabefelds für den XPath-Ausdruck wird angezeigt, wenn eine zweite oder darauf folgende Bedingung zu einer Bedingungsvorlage hinzugefügt wird. Wenn Sie das Kontrollkästchen "Andernfalls" aktivieren, wird die optionale Bedingung "Andernfalls" einer Bedingungsvorlage eingefügt. Nähere Informationen zur Verwendung der Bedingung "Andernfalls" finden Sie unter [Bedingungsvorlagen](#).

### XPath-Ausdrücke, die Zeilenschaltungen (CR/LFs) enthalten

Sie können Zeilenschaltungen (CR/LFs) in den XPath-Ausdruck einschließen, damit ein Teil der Ausgabe auf separaten Zeilen ausgegeben wird. Damit CR/LF jedoch in der Ausgabe (allen Ausgaben mit Ausnahme von RTF-Ausgabe) zu sehen sind, muss die Komponente, die den XPath-Ausdruck enthält, in den Sonderabsatztyp `pre` eingeschlossen werden. Ein Beispiel dafür ist der XPath-Ausdruck:

```
translate('a;b;c', ' ','', codepoints-to-string(13))
```

## 19.3 Menü "Datei"

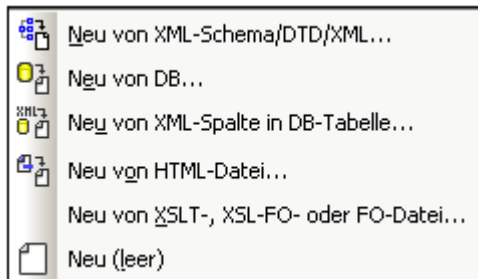
Das Menü **Datei** enthält Befehle für die Arbeit mit SPSs und verwandten Dateien. Es stehen die folgenden Befehle zur Verfügung:

- [Neu](#), zur Erstellung eines neuen SPS anhand von verschiedensten Quellen.
- [Öffnen](#), [Neu laden](#), [Schließen](#), [Alle schließen](#) zum Öffnen und Schließen und zum Neuladen der aktiven Datei.
- [Design speichern](#), [Design speichern unter](#), [Alles speichern](#): Befehle zum Speichern des aktiven SPS und aller geöffneten SPS Dateien.
- [XML-Authentic-Daten speichern](#), [Speichern unter](#): Dieser Befehl ist in der Authentic-Ansicht aktiv und dient zum Speichern der [XML-Arbeitsdatei](#).
- [Generierte Dateien speichern](#): Dient zum Speichern von Ausgabedateien, die mit Hilfe des SPS generiert werden können.
- [Web Design](#): Generiert alle zur Ausführung einer ASPX-Applikation erforderlichen Dateien im angegebenen Ordner.
  - [Eigenschaften](#): 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 (*Enterprise Edition*) eingebettet oder durch Verknüpfung eingefügt werden sollen.
  - [Druckvorschau](#), [Drucken](#): Ist in der Authentic-Ansicht und der Ausgabeansicht aktiv. Mit Hilfe dieser Befehle wird die Ansicht in den Vorschauen gedruckt.
  - [Zuletzt verwendete Dateien](#), [Beenden](#): Dient zum Öffnen einer zuletzt verwendeten Datei bzw. zum Beenden des Programms.



## 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 fünf Arten erstellen können:



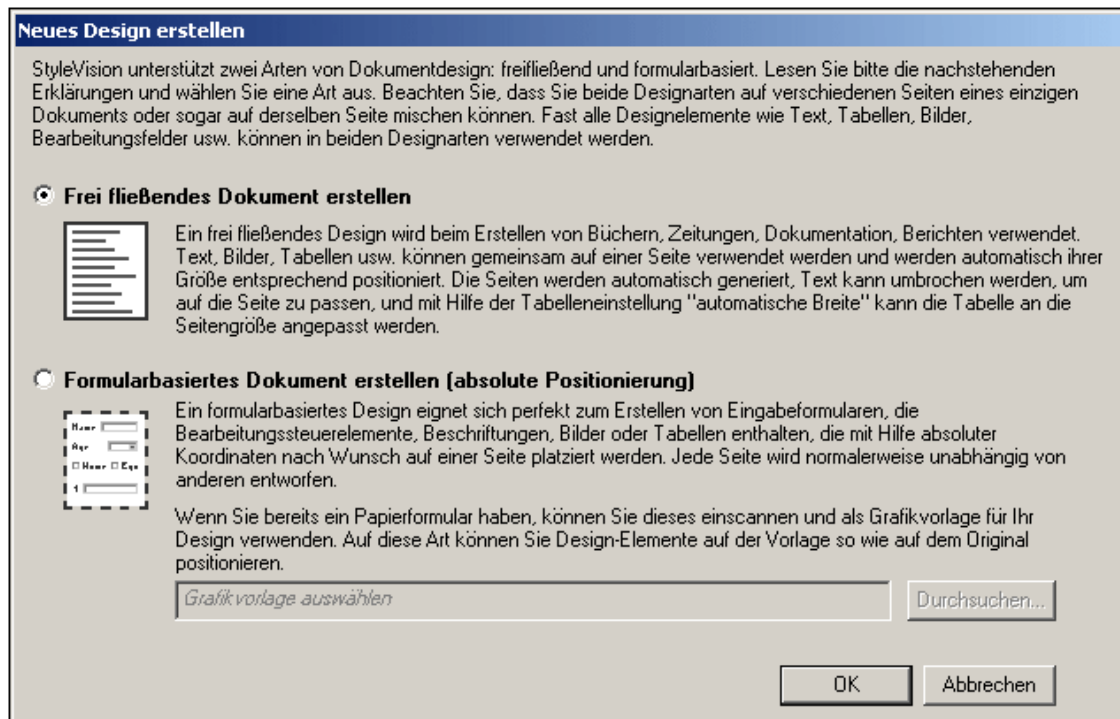
- Ein neues SPS basierend auf einem XML-Schema, einer DTD oder einem anhand einer XML-Datei generierten XML-Schema (**Neu von XML-Schema / DTD / XML**). Das ausgewählte Schema wird zur [Seitenleiste "Design-Übersicht"](#) hinzugefügt, zur Struktur der Schemaquellen wird (in der [Seitenleiste "Schema-Struktur"](#)) eine grafische Darstellung der Struktur hinzugefügt und in der [Design-Ansicht](#) wird ein SPS mit einer leeren Hauptvorlage erstellt. Ein neues SPS kann auch anhand einer Datei (Schema oder XML) über eine URL oder globale Ressource erstellt werden (*siehe unten*).
- Ein neues SPS basierend auf einem XML-Schema, das anhand einer von Ihnen ausgewählten Datenbank generiert wird (**Neu von DB** oder **Neu von XML-Spalte in IBM DB2**). Die Herstellung der Verbindung wird im Abschnitt [Herstellen einer Verbindung zu einer DB und Einrichten des SPS](#) beschrieben. Das SPS wird in der [Design-Ansicht](#) mit einer leeren Hauptvorlage erstellt.
- Ein neues SPS basierend auf einem benutzerdefinierten Schema, das Sie Node für Node anhand einer [HTML-Datei](#) erstellen (**Neu von HTML-Datei**). Das benutzerdefinierte Schema wird zu den Seitenleisten [Design-Übersicht](#) und [Schema-Struktur](#) hinzugefügt. In der Struktur der Schemaquellen hat es ein einziges Dokument-Element (Root-Element) und die HTML-Datei wird in die [Design-Ansicht](#) 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](#).
- Ein neues leeres SPS (**Neu (leer)**). Es wird kein Schema zur Seitenleiste "Design-Übersicht" oder der Struktur der Schemaquellen hinzugefügt. In der [Design-Ansicht](#) wird eine leere Hauptvorlage erstellt.

**Anmerkung:** Zum Definieren einer Datei oder Datenbankressource kann auch eine [globale Ressource](#) 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).

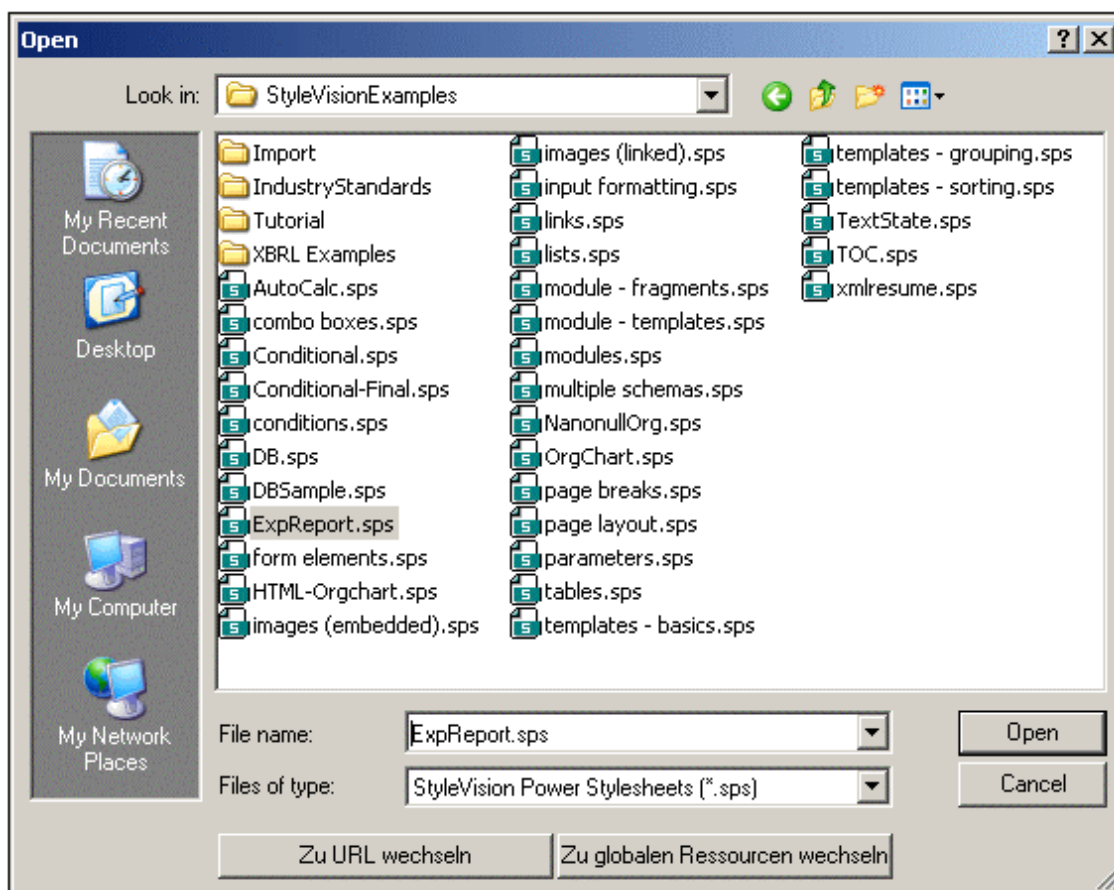


In einem frei fließenden Dokument ist das Dokument vom Layout her so angelegt, dass es den Maßen des Ausgabemediums entspricht (Papier oder Bildschirm). Elemente im Dokumentinhalt können nur relativ zueinander platziert werden, nicht aber an einer absoluten Position. Diese Art von Design eignet sich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

Ein in einem formularbasierten Dokument wird ein einziger [Layout Container](#) 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 orientiert soll, können Sie ein Originalformular als [Grafikvorlage](#) 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

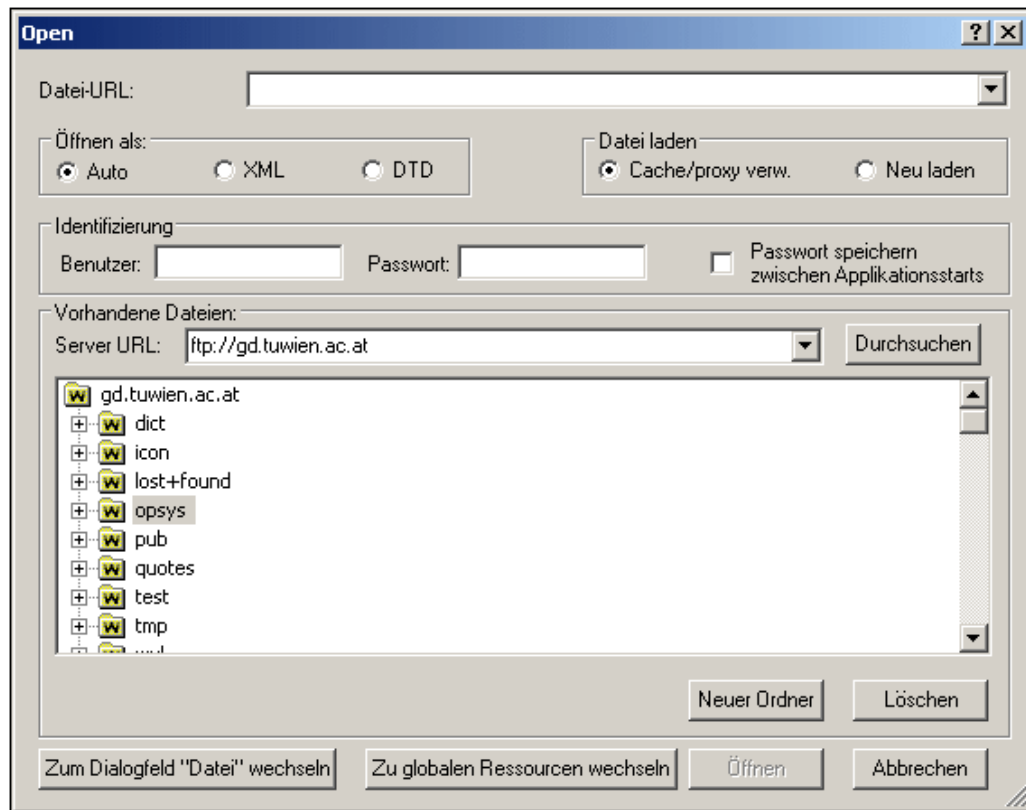
Es gibt mehrere Dialogfelder zum Öffnen und Speichern von Dateien, über die Sie die gewünschte Datei entweder über eine URL oder eine globale Ressource (*siehe Abbildung unten*) auswählen bzw. speichern können. Wählen Sie den Befehl **Zu URL wechseln** oder **Zu globalen Ressourcen wechseln**, um das entsprechende Dialogfeld aufzurufen.



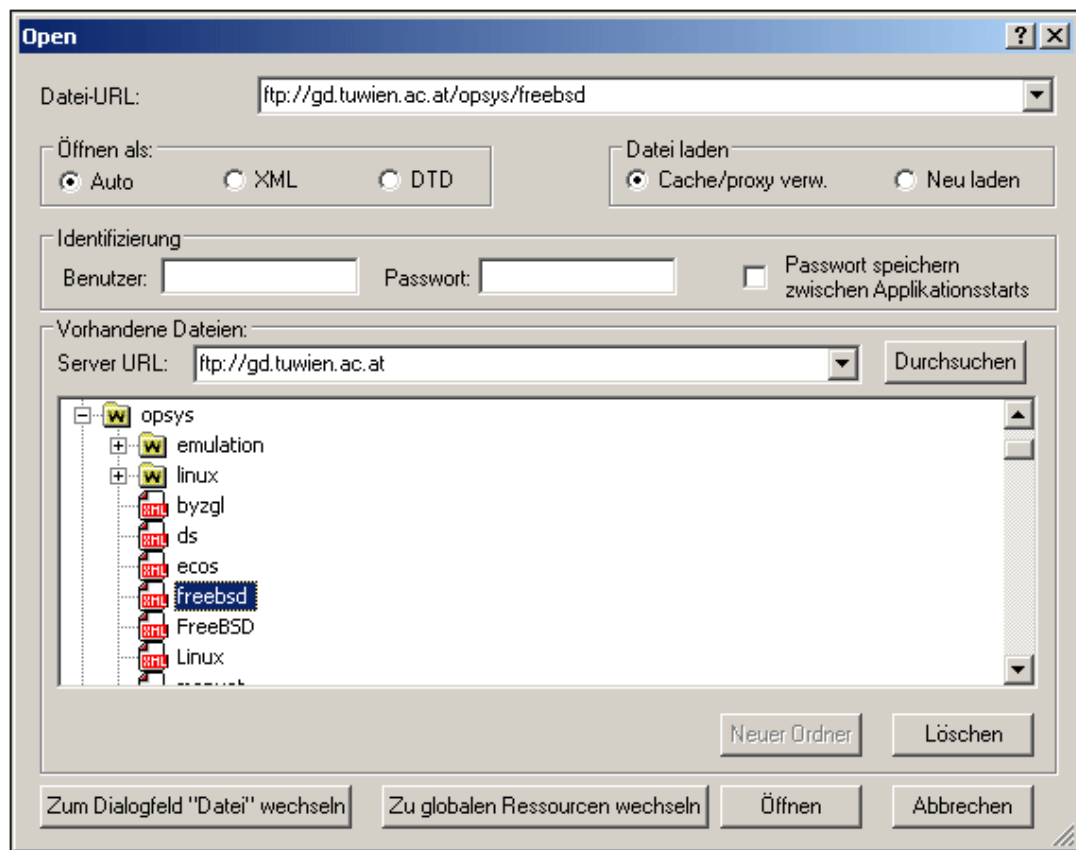
### Auswählen von Dateien über URLs

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.



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.

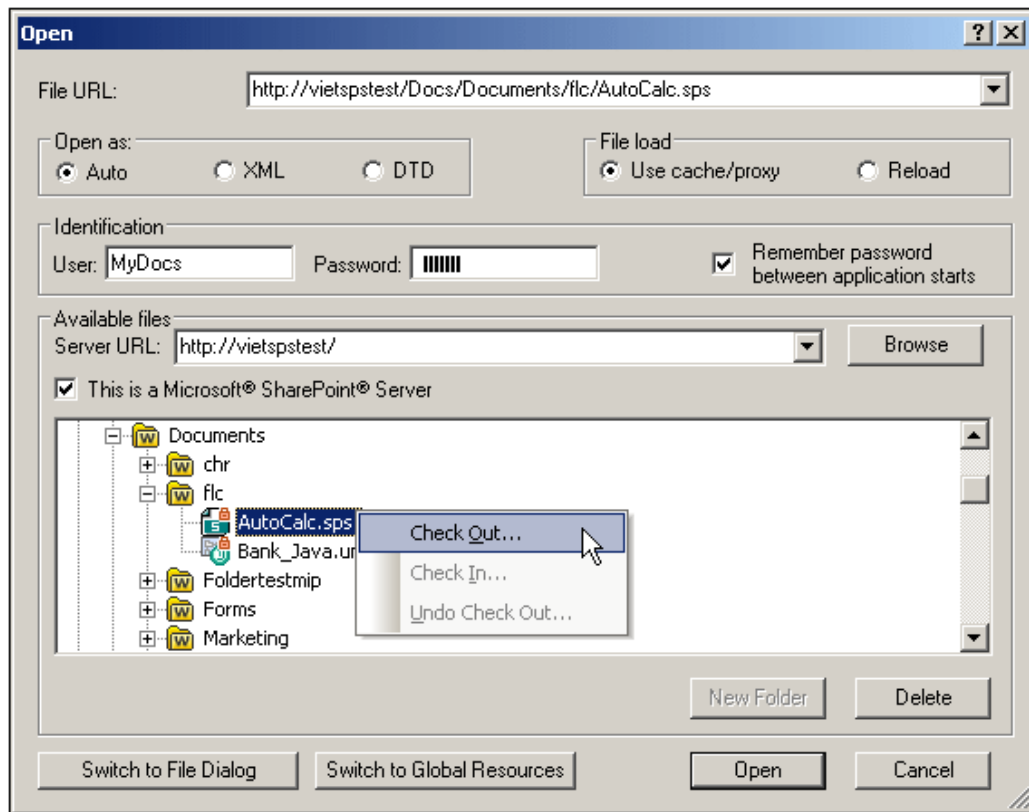
**Anmerkung:** Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.

**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.

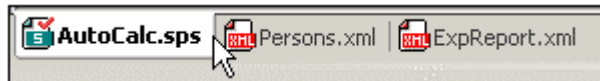


Wenn Sie mit der rechten Maustaste auf eine Datei klicken, erscheint ein Kontextmenü mit Befehlen, die für diese Datei zur Verfügung stehen (*Abbildung oben*).

- 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 auf das Dateiregister im Hauptfenster Ihrer Applikation klicken (*Abbildung unten*), einchecken.



- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht ausgecheckt 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 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.

#### **Öffnen und Speichern von Dateien über globale Ressourcen**

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

## Öffnen, Neu laden, Schließen, Alle schließen

Über den Befehl **Öffnen** (**Strg+O**)  können Sie eine vorhandene SPS Datei öffnen. Daraufhin wird das aus Windows bekannte [Dialogfeld "Öffnen"](#) 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](#) geschlossen werden. Wenn das Dokument nicht gespeicherte Änderungen enthält, werden Sie aufgefordert, diese Änderungen zu speichern.



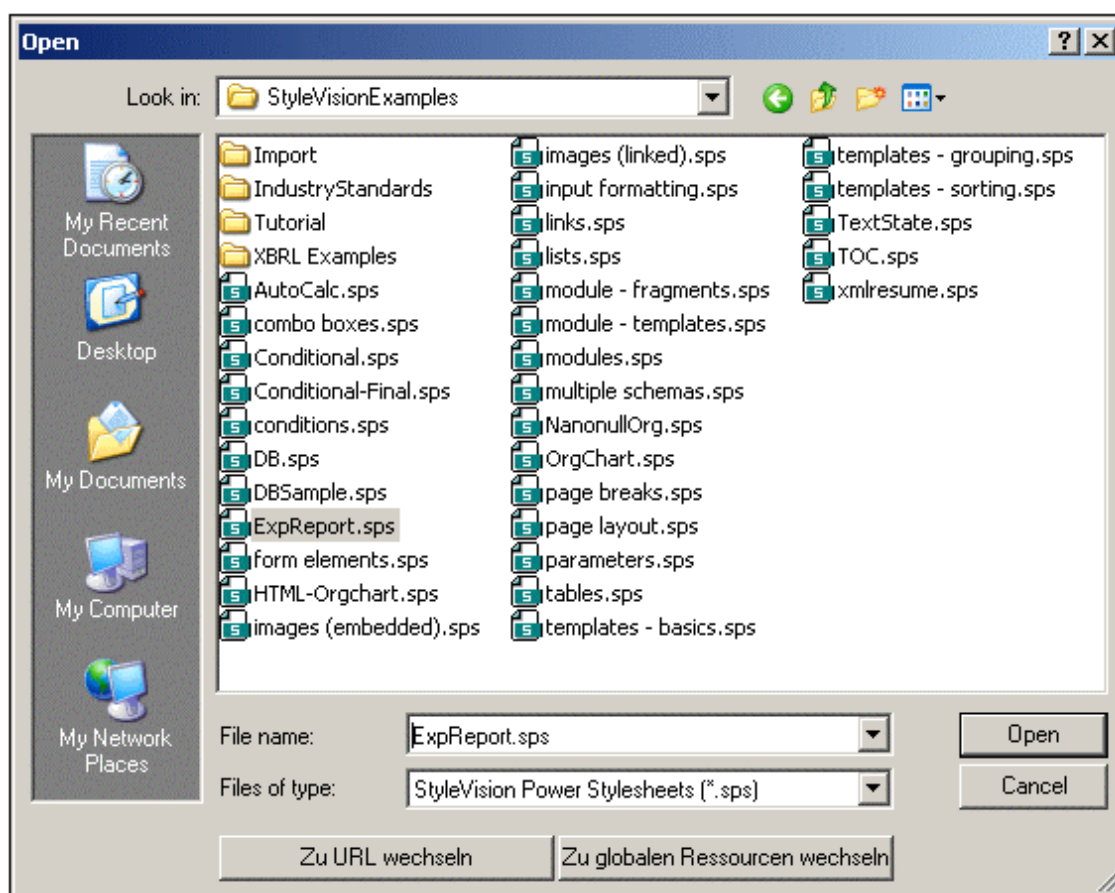
## Design speichern, Alles speichern

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 von Dateien über URLs und globale Ressourcen

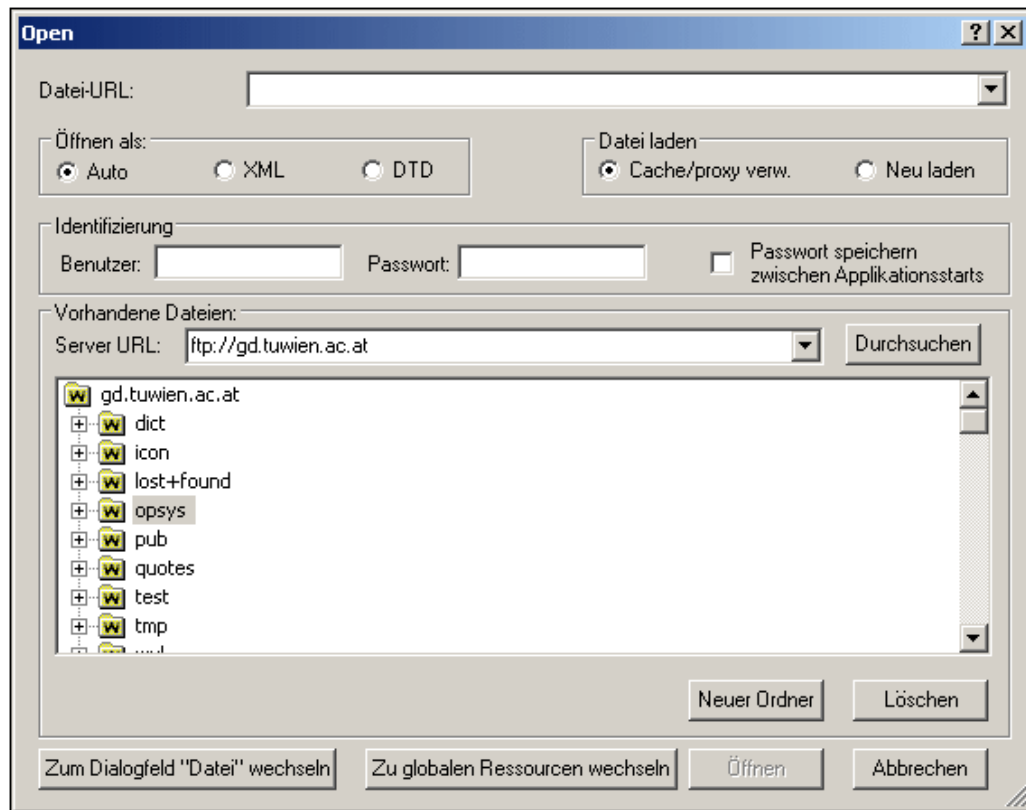
Es gibt mehrere Dialogfelder zum Öffnen und Speichern von Dateien, über die Sie die gewünschte Datei entweder über eine URL oder eine globale Ressource (siehe *Abbildung unten*) auswählen bzw. speichern können. Wählen Sie den Befehl **Zu URL wechseln** oder **Zu globalen Ressourcen wechseln**, um das entsprechende Dialogfeld aufzurufen.



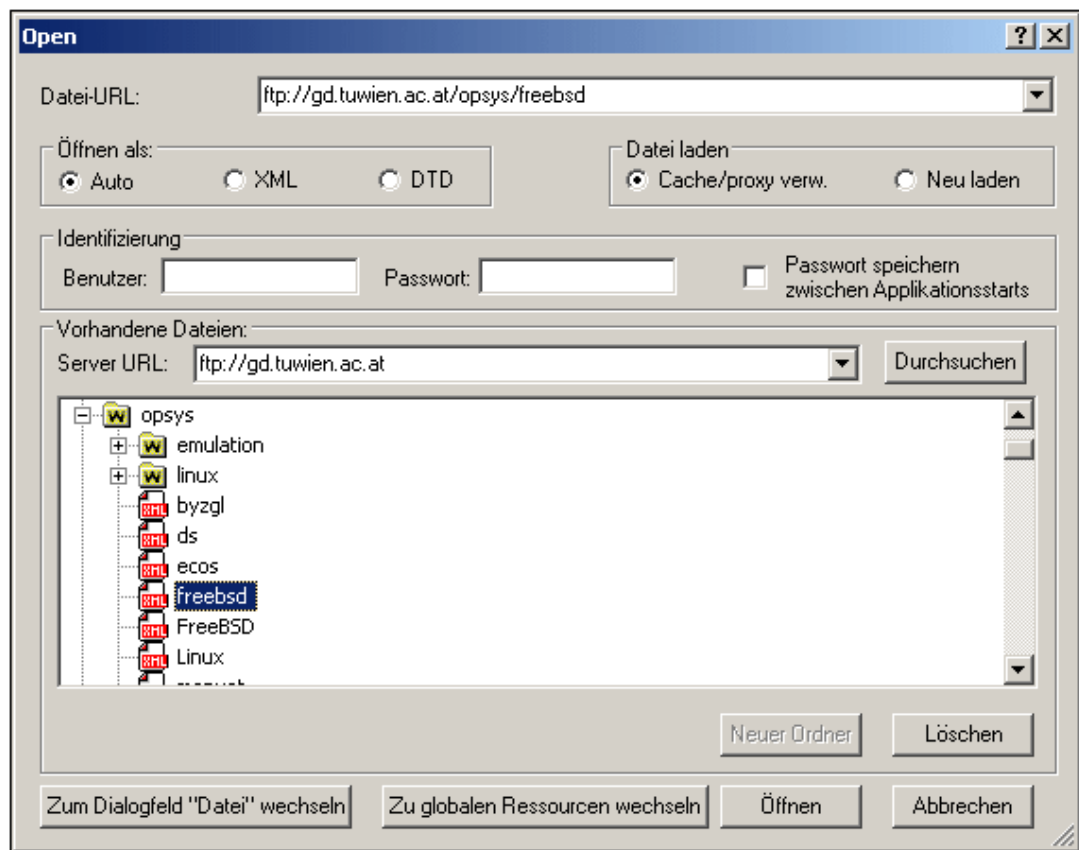
## Auswählen von Dateien über URLs

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.



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.

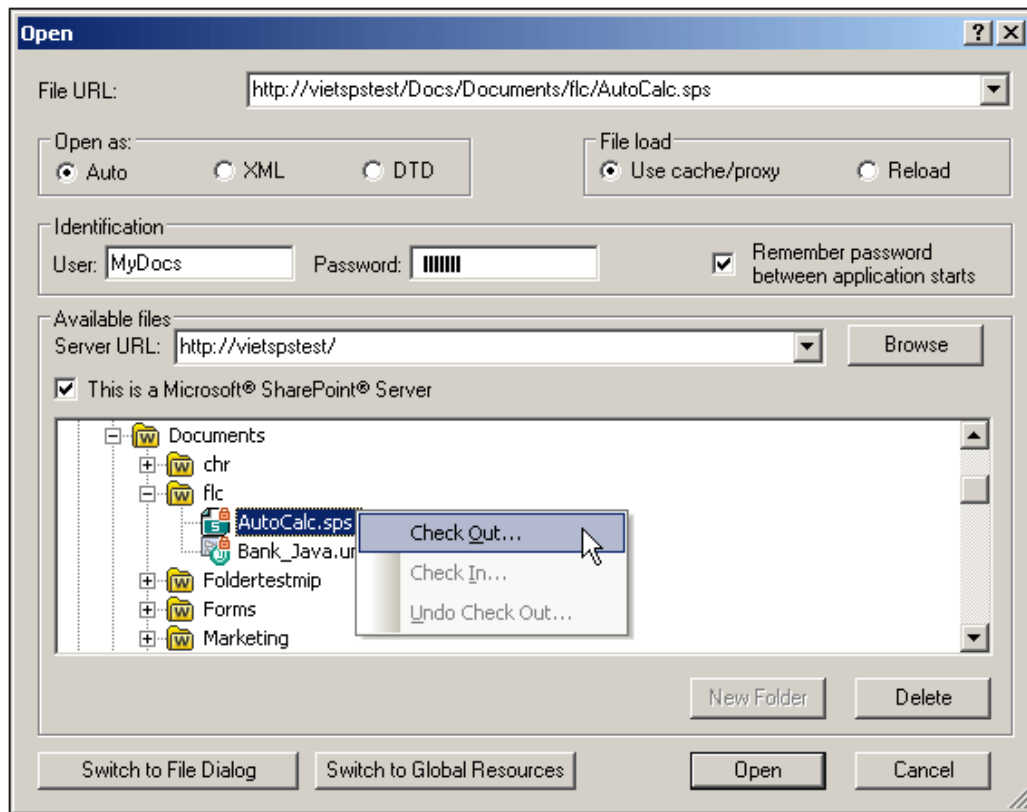
**Anmerkung:** Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.

**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.

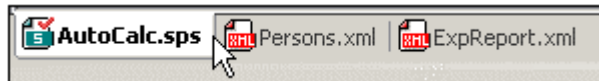


Wenn Sie mit der rechten Maustaste auf eine Datei klicken, erscheint ein Kontextmenü mit Befehlen, die für diese Datei zur Verfügung stehen (*Abbildung oben*).

- 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 auf das Dateiregister im Hauptfenster Ihrer Applikation klicken (*Abbildung unten*), einchecken.



- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht ausgecheckt 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 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.

#### **Öffnen und Speichern von Dateien über globale Ressourcen**

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

## 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 unter" (*Abbildung unten*) geöffnet. Wählen Sie das gewünschte Format aus und klicken Sie auf **OK**.



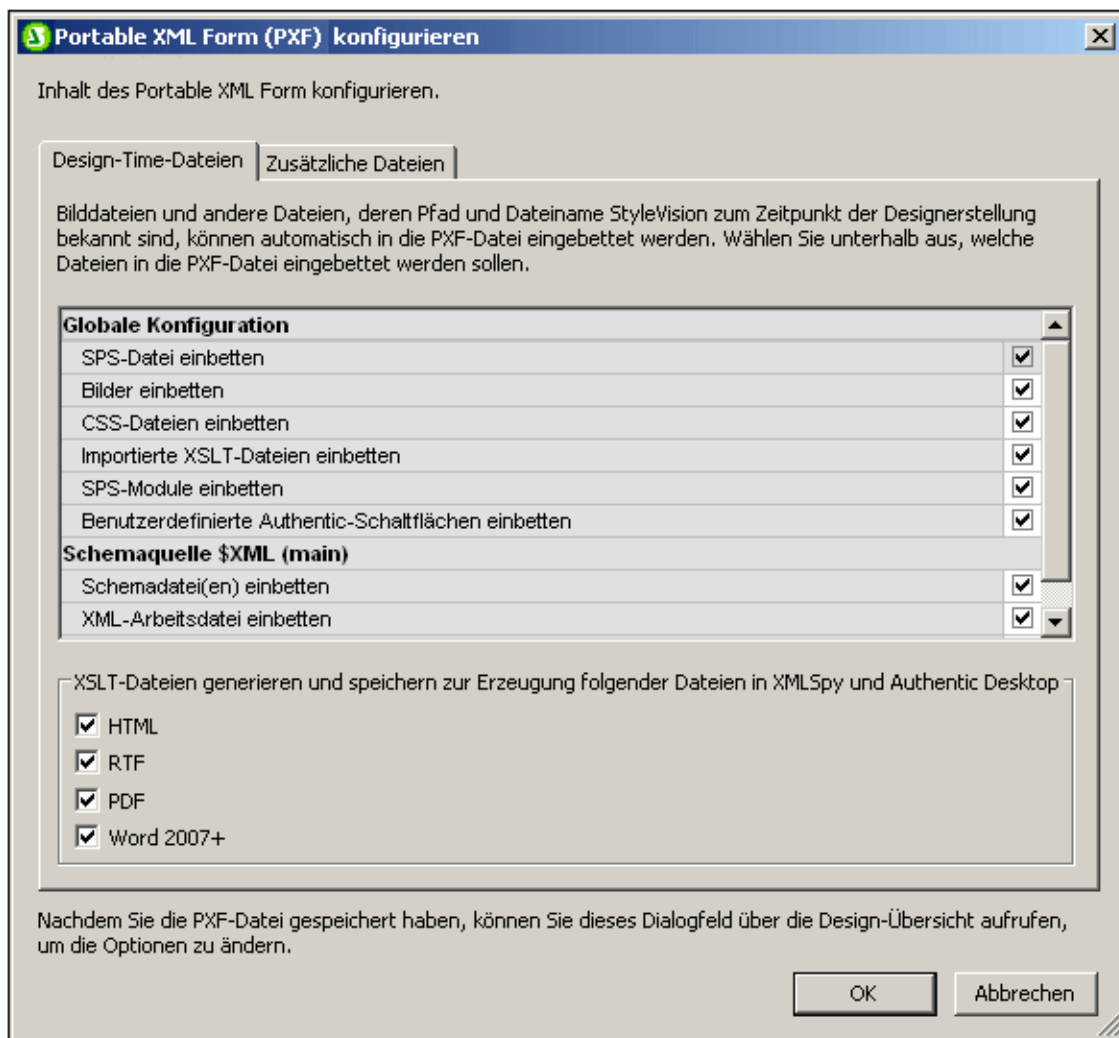
Das **SPS-Format** ist das Altova-Standardformat für StyleVision Designs. Das **PXF-Format** ist ein Altova-Format, mit dem alle mit dem Design in Zusammenhang stehenden Dateien (Schema-Dateien, XML-Dateien, Bilddateien, generierte XSLT-Stylehsheets, usw.) zusammen mit dem Design eingebettet werden können. Dieses Format eignet sich gut zum Transferieren aller Dateien, die zum Öffnen des Designs in der Authentic-Ansicht und/oder zum Generieren der HTML- und/oder RTF-Ausgabe anhand des Designs benötigt werden.

### Als SPS-Datei speichern

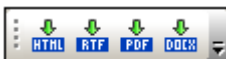
Bei Aufruf der SPS-Option wird das aus Windows bekannte Dialogfeld "Speichern unter" angezeigt. Das Speichern funktioniert genau so, wie für den Befehl [Design speichern](#) beschrieben. Der Vorteil des Befehls **Speichern unter** ist, dass Dateien, die bereits unter einem Dateinamen gespeichert wurden, unter einem anderen Dateinamen gespeichert werden können.

### Als PXF-Datei speichern

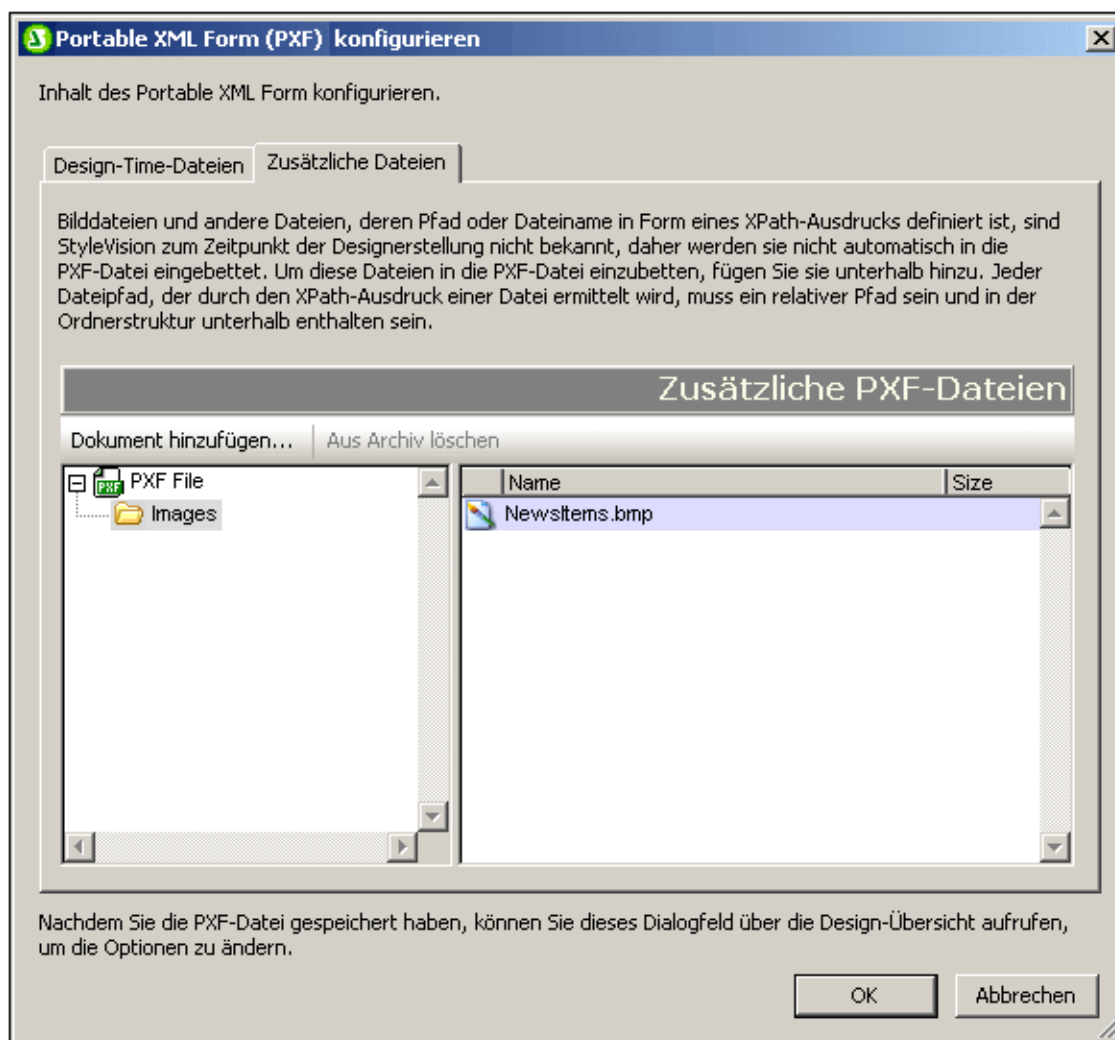
Bei Auswahl der PXF-Option wird das bekannte Windows-Dialogfeld "Speichern unter" angezeigt. Dieser Speicherbefehl funktioniert genau so, wie der [Befehl "Speichern unter"](#), mit dem zusätzlichen Schritt, dass die Dateien, die in die PFX-Datei inkludiert werden sollen, ausgewählt werden müssen. Nachdem 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. 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/NewsItem.bmp` referenziert werden.



## XML-Authentic-Daten speichern, Speichern unter

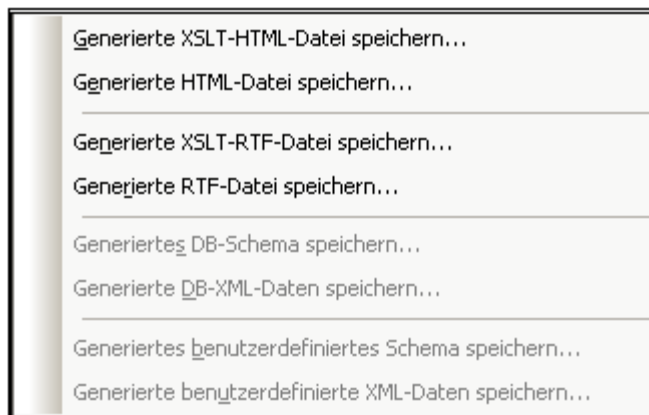


Sie können in der Authentic-Ansicht die XML-Arbeitsdatei oder die mit dem SPS verknüpfte Datenbank bearbeiten. Mit dem Befehl **XML-Authentic-Daten speichern** werden diese Änderungen an der XML-Arbeitsdatei oder der Datenbank gespeichert. Anstatt die XML-Datei in StyleVision zu bearbeiten, können Sie ein XML-Dokument oder eine DB auch in der Authentic-Ansicht von Altova XMLSpy oder Altova Authentic Desktop bearbeiten.

Mit dem Befehl **XML-Authentic-Daten speichern unter** können Sie das Authentic-XML-Dokument als eine andere Datei speichern.

## 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](#).



### 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. Anschließend können Sie mit Hilfe dieser XSLT-Datei ein XML-Dokument ins HTML-Format transformieren.

### Generierte HTML-Datei speichern

Mit diesem Befehl wird eine HTML-Datei generiert. Für diesen Vorgang sind zwei Eingabedateien erforderlich:

- die XML-Arbeitsdatei, die der derzeit aktiven SPS-Datei zugewiesen ist. Wenn keine XML-Arbeitsdatei zugewiesen wurde, ist der Befehl "Generierte HTML-Datei speichern..." deaktiviert. Bei Datenbank-basierten SPSs ist die automatisch generierte nicht editierbare XML-Datei die XML-Arbeitsdatei (siehe [Das DB-Schema und XML-DB-Dateien](#)); es wird keine XML-Arbeitsdatei zugewiesen.
- eine XSLT-Datei, die automatisch anhand der aktuell aktiven SPS-Datei generiert wird.

### Generierte XSLT-RTF-Datei speichern

Mit dem Befehl "Generierte XSLT-RTF-Datei speichern" wird anhand Ihres SPS eine XSLT-Datei für die RTF-Ausgabe generiert. Mit Hilfe dieser XSLT-Datei können Sie anschließend ein XML-Dokument ins RTF-Format transformieren.

### Generierte RTF-Datei speichern

Mit dem Befehl "Generierte RTF-Datei speichern" wird eine RTF-Datei generiert. Für diese Operation sind zwei Eingabedateien erforderlich:

- die XML-Arbeitsdatei, die der derzeit aktiven SPS-Datei zugewiesen ist. Wenn keine XML-Arbeitsdatei zugewiesen wurde, ist der Befehl deaktiviert. Bei Datenbank-basierten SPSs ist die automatisch generierte nicht editierbare XML-Datei die XML-Arbeitsdatei (siehe [Das DB-Schema und XML-DB-Dateien](#)); es wird keine XML-Arbeitsdatei zugewiesen.
- eine XSLT-für-RTF-Datei, die automatisch anhand der aktuell aktiven SPS-Datei

generiert wird.

**Generiertes DB-Schema speichern**

Beim Herstellen einer Verbindung zu einer Datenbank zur Erstellung eines DB-basierten SPS generiert und lädt StyleVision ein temporäres XML-Schema, das auf der Datenbankstruktur basiert. Mit dem Befehl "Generiertes DB-Schema speichern" können Sie dieses generierte XML-Schema speichern. Beachten Sie, dass StyleVision für XML-Datenbanken keine Schemadatei generiert, sondern eine Schema-Datei aus der DB oder einem anderen externen Ordner verwendet. Daher ist dieser Befehl für XML-Datenbanken nicht aktiv.

**Generierte DB-XML-Daten speichern**

Mit diesem Befehl wird eine XML-Datei generiert und gespeichert, die die Daten aus der Datenbank in einer XML-Struktur enthält, die der Struktur des XML-Schemas entspricht, das anhand der Datenbank erzeugt wurde. Wenn im StyleVision Power Stylesheet DB-Filter definiert wurden, werden diese auf den Datenimport angewendet. Beachten Sie, dass StyleVision für XML-Datenbanken keine XML-Datei generiert, sondern XML-Daten in den XML-Spalten der XML-DB verwendet. Daher ist dieser Befehl für XML-Datenbanken nicht aktiv.

**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.

## 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](#).

## Eigenschaften

Mit dem Befehl **Eigenschaften** wird das Dialogfeld "Eigenschaften" aufgerufen, 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 [Standard-Kodierung](#) für alle im Anschluss daran erstellten SPS Dokumente definieren; dies erfolgt im Dialogfeld "Optionen" über das Register "Kodierung".
- **HTML-Ausgabemodus:** 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, IE 9 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 IE 9. (Beachten Sie, dass IE 9 oder höher installiert sein muss, damit der IE 9 Kompatibilitätsmodus angewendet werden kann). Die entsprechende IE-Unterstützung steht sofort in der Authentic-Ansicht und in der HTML-Vorschau zur Verfügung. Beachten Sie, dass bei neuen SPS-Dokumente IE7-Kompatibilität ausgewählt ist. SPS-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](#)).

### XSLT

Auf dem Register "XSLT" kann die XSLT-Version für das aktive Dokument ausgewählt werden. Wenn Sie die Option "xsl:import Schemadeklaration verwenden" aktivieren, wird das `xsl:import` Element der XSLT 2-Spezifikation in das von StyleVision generierte XSLT 2.0-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.

### Bilder

Auf diesem Register können Sie wählen ob Bilder in der RTF-Ausgabe verknüpft oder eingebettet werden sollen. Einbetten ist bei Verwendung von XSLT 2.0 möglich. Diese Einstellung wird für jedes SPS einzeln vorgenommen. Um Bilder einzubetten, gehen Sie folgendermaßen vor:

1. Öffnen Sie, während das gewünschte SPS aktiv ist, das Dialogfeld "Eigenschaften" (**Datei | Eigenschaften**).
2. Aktivieren Sie das Kontrollkästchen "Bilder einbetten" (standardmäßig aktiviert). Beachten Sie, dass Bilder nur eingebettet werden, wenn als XSLT-Version des aktiven SPS XSLT 2.0 ausgewählt ist.
3. Klicken Sie auf **OK** und speichern Sie das SPS. Die Einstellung wird für das aktive SPS gespeichert.

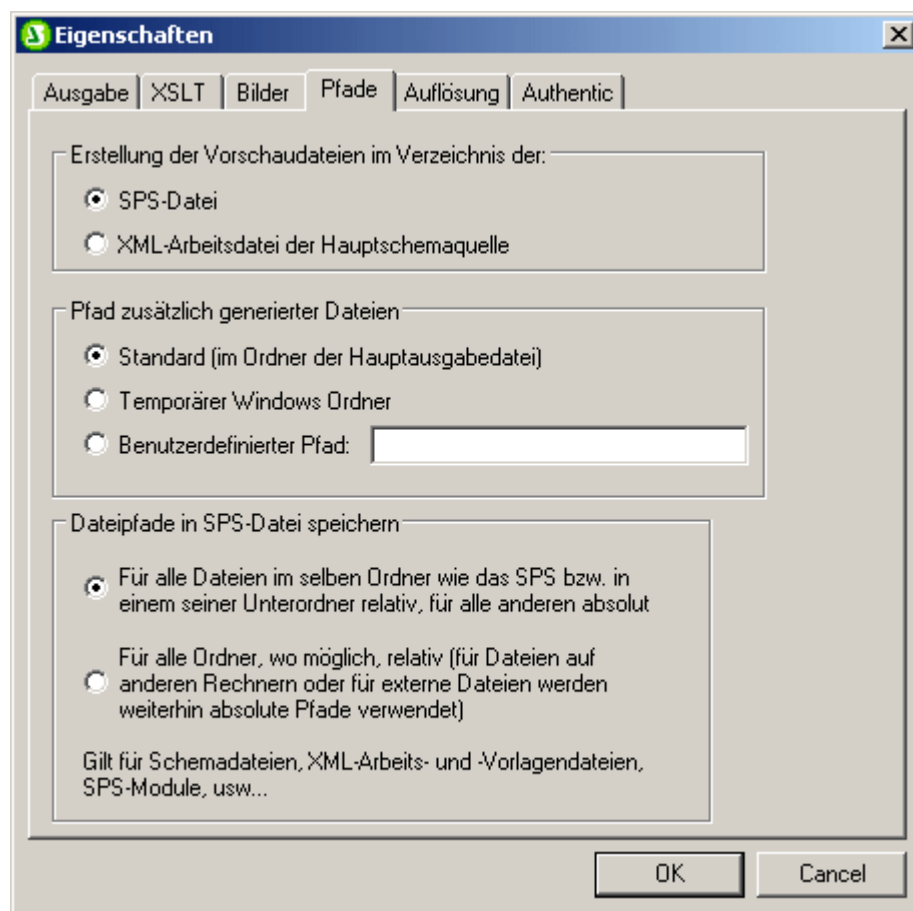
Falls das Kontrollkästchen "Bilder einbetten" deaktiviert ist, werden die Bilder gemäß dem in

den Bildeigenschaften definierten Bildpfad (wählen Sie das Bild aus und wählen Sie in der [Eingabehilfe "Eigenschaften"](#) die URL aus) verknüpft. Nähere Information zur Auflösung von Pfaden finden Sie im Abschnitt [Bild-URLs](#).

**Anmerkung:** Beachten Sie, dass die Einbettung von Bildern im RTF-Format nur für EMF-, JPG- und PNG-Dateien unterstützt wird.

### 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.



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.

### Auflösung

Der Auflösungsfaktor bestimmt, wie im Design in Pixeln definierte Längen in Längeneinheiten für den Druck konvertiert werden sollen. In der Dropdown-Liste der Auswahlliste stehen 72dpi, 96dpi oder 120dpi zur Verfügung. Je höher der dpi-Wert, desto kleiner sind die einzelnen Pixel und die entsprechende absolute Länge. Die Konvertierungsfaktoren (zur Berechnung der Anzahl der Punkte anhand der Anzahl der Pixel) für die einzelnen dpi-Werte sind 1,0, 0,75 bzw. 0,6. Eine ausführlichere Erklärung dazu finden Sie unter [Pixelauflösung](#).

### Authentic

Die folgenden Authentic-Einstellungen können definiert werden:

- **Standardeinfügemodus für Authentic:** Definiert, ob eine in der Zwischenablage gespeicherte Auswahl der Authentic-Ansicht standardmäßig als XML oder Text in die Authentic-Ansicht eingefügt werden soll. Wenn sie als XML eingefügt wird, wird sie mit Markup-Tags (XML-Tags) eingefügt, wenn der kopierte Ausschnitt Markup-Tags enthält. Andernfalls wird Textinhalt von Nodes im Standardeinfügemodus ohne Markup-Tags eingefügt. Sie können den Standardeinfügemodus in der Authentic-Ansicht außer Kraft setzen, indem Sie am Einfügepunkt auf die rechte Maustaste klicken und im angezeigten Kontextmenü je nach Bedarf den Befehl **Einfügen als | XML** oder **Einfügen als | Text** auswählen. Sie können den Standardeinfügemodus jederzeit beim Bearbeiten des SPS-Dokuments ändern.

Sie können auch festlegen, ob der Benutzer bei Text, der mit der Maus in die Authentic-Ansicht gezogen wird, auswählen kann, wie der Text eingefügt werden soll oder ob der Standardeinfügemodus verwendet werden soll. Damit der Benutzer die Wahl zwischen diesen Modi hat, aktivieren Sie das Optionsfeld *Bei jedem Drag & Drop-Vorgang fragen*; um den Standardeinfügemodus zu verwenden, ohne dass der Benutzer gefragt wird, aktivieren Sie das Optionsfeld *Standardmodus für "Einfügen" verwenden*.

- **Authentic Skripterstellung:** Dies sind Optionen für Skripts, die mit einem SPS gespeichert werden: (i) Skripts können aktiviert oder nicht aktiviert werden; (ii) Events im Skript können verarbeitet oder deaktiviert werden; (iii) Makros in den Skripts können außerhalb von StyleVision (also in der Authentic-Ansicht von anderen Altova-Produkten) im Debug-Modus ausgeführt werden oder deaktiviert sein. Diese Funktion ist nützlich, wenn Sie ein Makro außerhalb der StyleVision Umgebung testen und debuggen möchten.
- **Maximale Tiefe des Vorlagenaufrufs:** Definiert die maximale Anzahl von Vorlagenaufrufen, die in der Authentic-Ausgabe rekursiv gemacht werden dürfen. Wenn die Zahl der Vorlagenaufrufe die hier angegebene Ziffer übersteigt, wird eine Fehlermeldung ausgegeben. Diese Einstellung kann zur Vermeidung von Endlosschleifen in Authentic verwendet werden.

## 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 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.

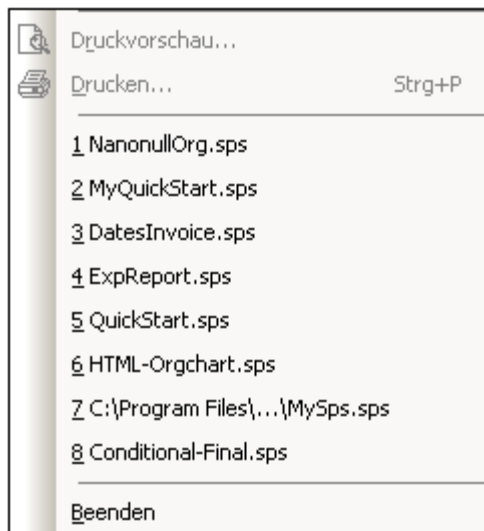
Der Befehl **Drucken**  ist in der Authentic-Ansicht und auf den Ausgabevorschauregistern aktiv. Sie können damit je nachdem, wie die Seite für diese Ansicht konfiguriert ist, die ausgewählte Ansicht der XML-Arbeitsdatei drucken. Beachten Sie, dass Sie die Seiteneinrichtung für die Authentic-Ansicht 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**.



## 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.

## 19.4 Menü "Bearbeiten"

Das Menü **Bearbeiten** enthält Befehle, die Ihnen bei der Bearbeitung von SPS und Dokumenten der Authentic-Ansicht helfen. Neben den Standardbearbeitungsbefehlen wie Ausschneiden (Umschalt+Entf), 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](#): Dienen zum Rückgängigmachen bzw. Wiederherstellen vorhergehender Aktionen und zum Auswählen des gesamten Inhalts des SPS.
- [Suchen, Weitersuchen, Ersetzen](#): Dienen zum Suchen von Text im SPS, in der Authentic-Ansicht und in XSLT-Stylesheet-Vorschauen und zum Ersetzen von Text in der Authentic-Ansicht.
- [Stylesheet Parameter](#): Dient zum Bearbeiten von global für das SPS deklarierten Parametern.
- [Markup reduzieren/erweitern](#): 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.

## 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.

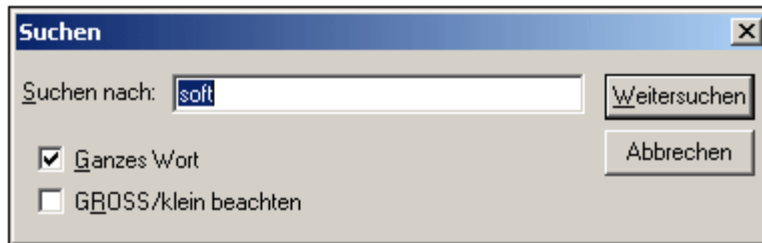
Mit dem Befehl **Alles markieren** wird der gesamte Inhalt des Design-Dokumentfensters markiert.

## Suchen, Weitersuchen, Ersetzen

Über den Befehl **Suchen (Strg+F)**  können Sie Wörter oder Wortfragmente in der Design-Ansicht, im JavaScript Editor, in der Authentic-Ansicht sowie im XSLT-für-HTML- und XSLT-für-RTF- Stylesheet suchen.

### Design-Ansicht, HTML-Vorschau und Authentic-Ansicht

Wenn Sie in der Design-Ansicht, HTML-Vorschau oder in der Authentic-Ansicht auf den Befehl **Suchen** klicken, wird das folgende Dialogfeld angezeigt:

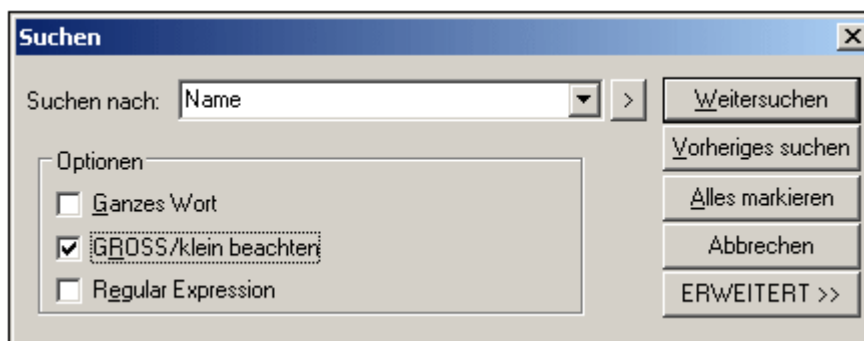


Beachten Sie bitte Folgendes:

- In der Design-Ansicht werden die statischen Daten durchsucht, nicht aber Node-Namen.
  - In der Authentic-Ansicht werden die dynamischen Daten (XML-Daten) durchsucht, nicht Text aus den statischen Eingabedaten (XSLT).
  - 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`.

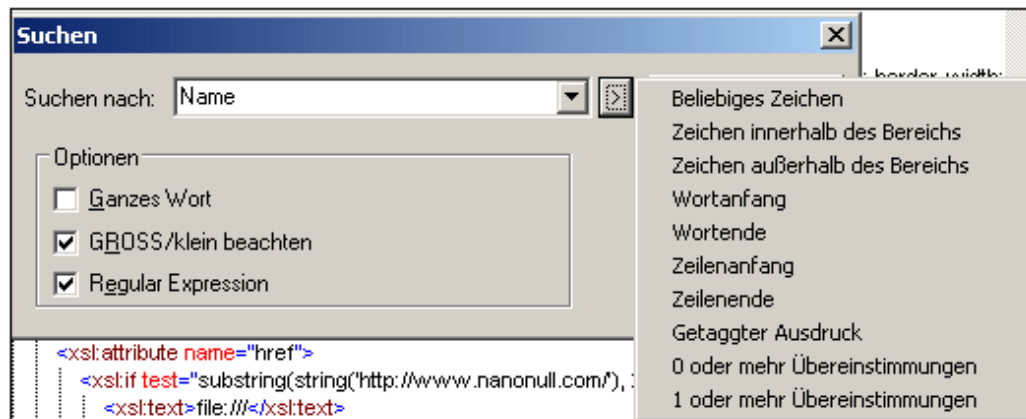
### XSLT-für-HTML und JavaScript Editor

Wenn Sie auf dem Register XSLT-für-HTML, XSLT-für-RTF oder JavaScript Editor den Befehl **Suchen** auswählen, erscheint das folgende Dialogfeld:

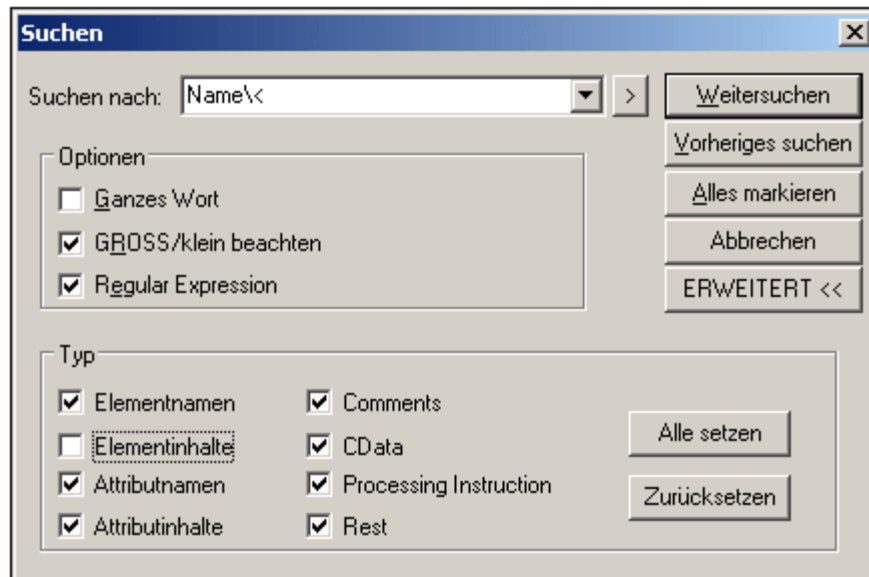


Beachten Sie bitte die folgenden Punkte:

- Um als Suchbegriff eine Regular Expression zu verwenden, aktivieren Sie das Kontrollkästchen "Regular expression". Sie können mit Hilfe eines Menüs, das angezeigt wird, wenn Sie mit der rechten Maustaste neben dem Eingabefeld für den Suchbegriff auf den Nach-rechts-Pfeil klicken, eine Regular Expression erstellen.



- Um einzuschränken, in welchem Bereich des Dokuments gesucht werden soll, klicken Sie auf die Schaltfläche "Erweitert". Es stehen Ihnen daraufhin mehr Suchoptionen zur Verfügung (*Abbildung unten*):



Wählen Sie die Art des Dokumentinhalts aus, den Sie durchsuchen möchten, indem Sie die entsprechenden Kontrollkästchen aktivieren.

### Befehl "Weitersuchen"

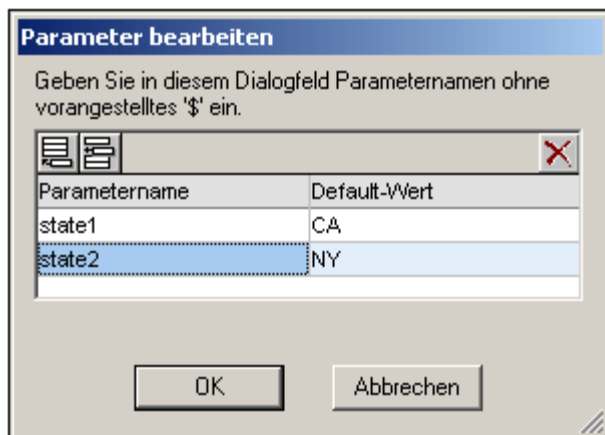
Mit dem Befehl **Weitersuchen** (F3)  wird der letzte Suchbefehl wiederholt und die nächste Instanz des gesuchten Texts wird gesucht. Eine Beschreibung zur Verwendung der Suchfunktion finden Sie unter [Suchen](#).

### Ersetzen (Strg+H)

Der Befehl **Ersetzen** ist in der Design-Ansicht, dem JavaScript Editor und der Authentic-Ansicht (*wird in der Basic Edition nicht unterstützt*) aktiv und ermöglicht Ihnen, nach einem Textstring zu suchen und ihn durch einen anderen Textstring zu ersetzen.

## 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.



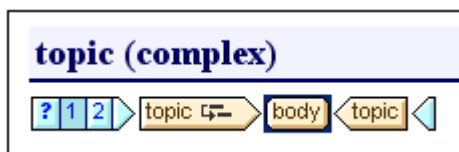
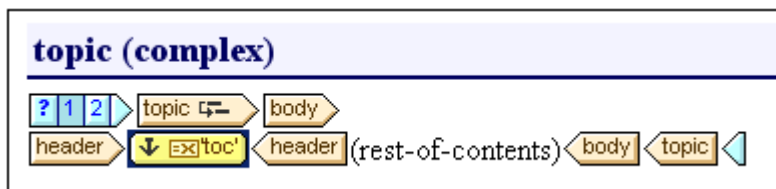
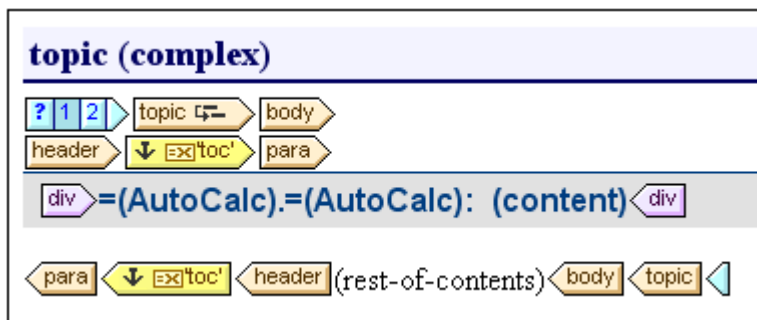
Beachten Sie bitte die folgenden Punkte:

- Sie können Parameter für das gesamte Stylesheet und für DB-Filter 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"](#) deklariert werden .

## 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.

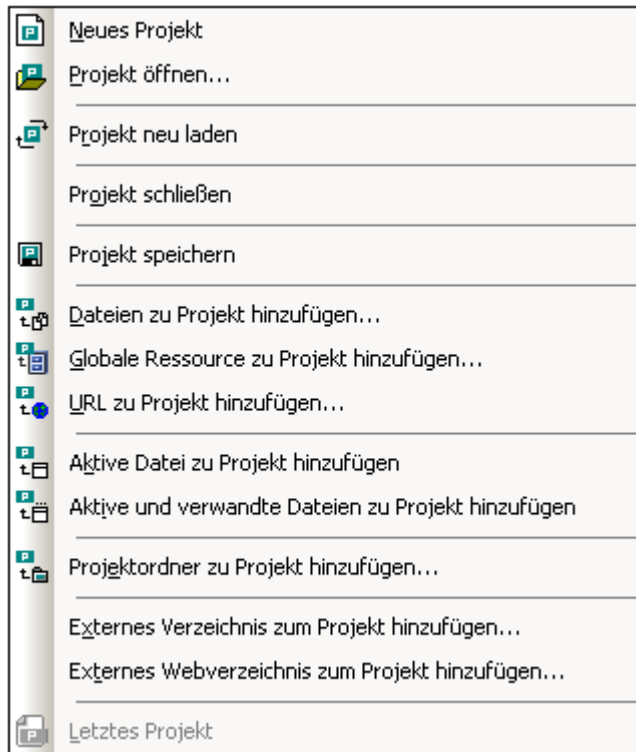
In der Abbildung unten sehen Sie, wie eine Reihe von Tags reduziert wird. Doppelklicken Sie auf einen reduzierten Tag, um ihn zu erweitern.



Durch Reduzieren von Tags können Sie die Anzeige Ihren Bedürfnissen bei der Bearbeitung entsprechend anpassen.

## 19.5 Menü "Projekt"

Über das Menü **Projekt** (Abbildung unten) können Sie Projekte erstellen, gliedern und ändern. Sie können ein Projekt im Handumdrehen einrichten, Dateien im Projekt definieren und Dateien nach Dateityp in separate Ordner legen. Ein Projekt wird in der Seitenleiste "Projekt" grafisch angezeigt. Hier können Sie Dateien zur Verwendung im SPS-Design aufrufen.



Das Menü **Projekt** enthält die folgenden Befehle, die in den Unterabschnitten dieses Abschnitts genau erklärt werden:

- [Neues Projekt](#), zum Erstellen eines neuen Projekts
- [Projekt öffnen](#), zum Öffnen eines bestehenden Projekts
- [Projekt neu laden](#), zum Aktualisieren eines Projekts in der Seitenleiste "Projekt"
- [Projekt schließen](#), zum Schließen eines Projekts in der Seitenleiste "Projekt"
- [Projekt speichern](#), zum Speichern und Benennen eines Projekts
- [Dateien zu Projekt hinzufügen](#), zum Hinzufügen von Dateien zu einem Projekt in der Seitenleiste "Projekt"
- [Globale Ressource zu Projekt hinzufügen](#), zum Hinzufügen von globalen Altova Ressourcen zu einem Projekt in der Seitenleiste "Projekt"
- [URL zu Projekt hinzufügen](#), zum Hinzufügen einer Datei zu einem Projekt in der Seitenleiste "Projekt" über eine URL
- [Aktive Datei zu Projekt hinzufügen](#), zum Hinzufügen der derzeit aktiven SPS-Datei zu einem Projekt
- [Aktive und verwandte Dateien zu Projekt hinzufügen](#), zum Hinzufügen der derzeit aktiven SPS-Datei und ihrer verwandten Dateien
- [Projektordner zu Projekt hinzufügen](#), zum Hinzufügen von Ordnern zu einem Projekt in der Seitenleiste "Projekt"
- [Externes Verzeichnis zum Projekt hinzufügen](#), zum Hinzufügen von lokalen Ordnern zu einem Projekt in der Seitenleiste "Projekt"



- [Externes Webverzeichnis zum Projekt hinzufügen](#), zum Hinzufügen von Ordnern zu einem Projekt in der Seitenleiste "Projekt" über eine URL

**Letztes Projekt**

Am unteren Rand des Menüs "Projekt" werden die Dateinamen der neun zuletzt verwendeten Projekte aufgelistet, über die Sie diese Dateien schnell aufrufen können.

**Drag-and-Drop**

Sie können einen Ordner im Projektfenster in einen anderen Ordner oder an eine andere Stelle im selben Ordner ziehen. Eine Datei kann in einen anderen Ordner gezogen, nicht aber im selben Ordner (in dem die Dateien in alphabetischer Reihenfolge geordnet sind) verschoben werden. Außerdem können Dateien und Ordner auch aus dem Windows Datei-Explorer in die Seitenleiste "Projekt" gezogen werden.

## Neues Projekt, Projekt öffnen, Projekt neu laden

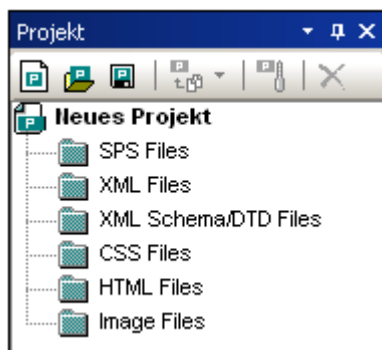
Mit dem Befehl **Neues Projekt**  wird ein neues Projekt erstellt. Das vorherige Projekt (falls vorhanden) wird in der Seitenleiste "Projekt" durch das neue ersetzt. Wenn das Projekt, an dem Sie gearbeitet haben, nicht gespeicherte Änderungen enthält, wird die Frage angezeigt, ob Sie die Änderungen am Projekt speichern möchten. Beachten Sie: Mit dem Befehl **Neues Projekt** wird nur ein neues Projekt erstellt. Es wird jedoch nicht gespeichert; Sie müssen das Projekt explizit mit dem Befehl [Projekt speichern](#) speichern.

Mit dem Befehl **Projekt öffnen**  wird ein bestehendes Projekt geöffnet und in der Seitenleiste "Projekt" angezeigt. Wenn in der Seitenleiste "Projekt" zuvor ein Projekt geöffnet war, wird dieses durch das geöffnete Projekt ersetzt. Wenn das vorherige Projekt, nicht gespeicherte Änderungen enthält, wird die Frage angezeigt, ob Sie die Änderungen am Projekt speichern möchten, bevor es in der Seitenleiste "Projekt" ersetzt wird.

Mit dem Befehl **Projekt neu laden**  wird das aktuelle Projekt neu geladen. Diese Befehl ist vor allem dann nützlich, wenn mehrere Benutzer an einem Projekt arbeiten und eventuell gleichzeitig am Projekt Änderungen vornehmen.

## Projekt schließen, Projekt speichern

Mit dem Befehl **Projekt schließen** wird das aktive Projekt geschlossen. Wenn das Projekt nicht gespeicherte Änderungen enthält, wird die Frage angezeigt, ob Sie die Änderungen am Projekt speichern möchten, bevor Sie es schließen. Ein Projekt mit nicht gespeicherten Änderungen wird durch ein Sternchen nach dem Projektnamen in der Seitenleiste "Projekt" gekennzeichnet (*Abbildung unten*).



Mit dem Befehl **Projekt speichern** wird das aktuelle Projekt gespeichert. Beachten Sie, dass ein Projekt seinen Namen beim ersten Speichern erhält. Ein Projekt kann nur außerhalb von StyleVision umbenannt werden; Sie können Sie Datei z.B. über den Windows Datei-Explorer umbenennen.

## Dateien / globale Ressource / URL zu Projekt hinzufügen

### Dateien zu Projekt hinzufügen

Mit dem Befehl **Dateien zu Projekt hinzufügen** werden Dateien zum aktuellen Projekt hinzugefügt. Mit dem Befehl wird das Dialogfeld "Öffnen" aufgerufen, in dem Sie eine einzelne Datei oder eine Gruppe der Dateien, die Sie zum Projekt hinzufügen möchten, auswählen. Die Datei(en) wird/werden entsprechend den für die einzelnen Unterordner definierten Dateierweiterungen zu den Unterordnern des Projekts hinzugefügt. Wenn dieselbe Dateierweiterung für mehr als einen Ordner definiert wurde, wird eine Datei mit dieser Dateierweiterung zum ersten Ordner (in der Projekt-Seitenleiste) hinzugefügt, für den diese Dateierweiterung definiert wurde.

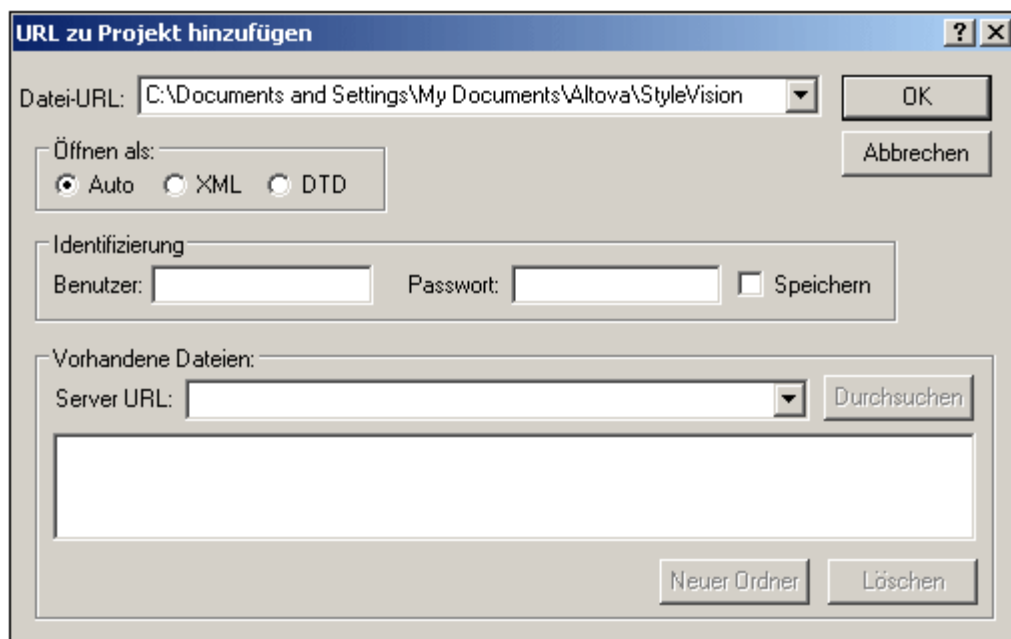
Um eine Datei zu einem Ordner oder einem Unterordner des Hauptprojektordners hinzuzufügen, klicken Sie mit der rechten Maustaste auf den Ordner, wählen Sie den Befehl **Dateien hinzufügen** und suchen Sie anschließend die gewünschte(n) Datei(en).

### Globale Ressource zu Projekt hinzufügen

Mit dem Befehl **Globale Ressource zu Projekt hinzufügen** wird ein Dialogfeld aufgerufen, in dem globale Ressourcen in der aktuell aktiven XML-Datei für globale Ressourcen aufgelistet sind und in dem Sie eine dieser Ressourcen auswählen können, um sie zum aktiven Projekt hinzuzufügen. Wählen Sie die gewünschte globale Ressource aus und klicken Sie auf **OK**.

### URL zu Projekt hinzufügen

Mit dem Befehl **URL zu Projekt hinzufügen** wird eine URL zum aktuellen Projekt hinzugefügt. Über URLs wird in einem Projekt das Zielobjekt der URL in das Projekt inkludiert. Bei Aufruf des Befehls erscheint das Dialogfeld **URL zu Projekt hinzufügen** (Abbildung unten).



Sie können entweder eine Datei-URL (mit oder ohne das `file: \\` Protokoll) oder eine Server-URL hinzufügen. Geben Sie bei einer Server-URL Ihren Benutzernamen und Ihr Passwort und anschließend die Server-URL ein. Klicken Sie auf **Durchsuchen**, um eine Verbindung zum Server herzustellen. Wählen Sie anschließend in der Liste der verfügbaren Dateien die gewünschte Datei aus.

Beachten Sie: URLs können auch zu Ordnern und Unterordnern des Hauptprojektordners hinzugefügt werden. Klicken Sie dazu mit der rechten Maustaste auf den Projektordner und wählen Sie den Befehl **URL hinzufügen**. Daraufhin wird das Dialogfeld "URL hinzufügen" aufgerufen. Fahren Sie fort, wie oben beschrieben.

#### **Drag-and-Drop**

Sie können eine Datei in einen anderen Ordner ziehen, nicht aber im selben Ordner (in dem die Dateien in alphabetischer Reihenfolge geordnet sind) verschieben. Außerdem können Dateien auch aus dem Windows Datei-Explorer in die Seitenleiste "Projekt" gezogen werden.

#### **Löschen von Dateien, Ressourcen und URLs**

Um eine Datei, Altova Ressource oder URL zu löschen, wählen Sie sie in der Projekt-Seitenleiste aus, klicken Sie auf die rechte Maustaste und wählen Sie im Kontextmenü den Befehl **Löschen**.

## Aktive (und verwandte) Dateien zu Projekt hinzufügen

### Aktive Datei zu Projekt hinzufügen

Mit dem Befehl **Aktive Datei zu Projekt hinzufügen** wird die aktive SPS-Datei zum aktuellen Projekt hinzugefügt. Diese Datei wird zum ersten Ordner hinzugefügt, der für die `.sps` Dateityperweiterung definiert wurde. Wenn nicht nur das SPS, sondern auch das dazugehörige Schema, die XML-Arbeitsdatei, die XML-Vorlagendatei und die CSS- und Bilddateien hinzugefügt werden sollen, verwenden Sie den Befehl **Aktive und verwandte Dateien zu Projekt hinzufügen** (siehe unten). Um die aktive Datei zu einem Ordner oder Unterordner des Hauptprojektordners hinzuzufügen, klicken Sie mit der rechten Maustaste auf den Ordner und wählen Sie den Befehl **Aktive Datei hinzufügen**.

### Aktive und verwandte Dateien zu Projekt hinzufügen

Mit dem Befehl **Aktive und verwandte Dateien zu Projekt hinzufügen** werden die gerade aktive SPS-Datei sowie die dazugehörigen Schema-Dateien und, falls vorhanden, die XML-Arbeitsdatei, die XML-Vorlagendatei, CSS- und Bilddateien zum Projekt hinzugefügt. Jede einzelne Datei wird zum ersten Ordner hinzugefügt, für den die entsprechende Dateierweiterung definiert wurde. Um die aktive und die dazugehörigen Dateien zu einem Ordner oder Unterordner des Hauptprojektordners hinzuzufügen, klicken Sie mit der rechten Maustaste auf den jeweiligen Ordner und wählen Sie den Befehl **Aktive und verwandte Dateien hinzufügen**.

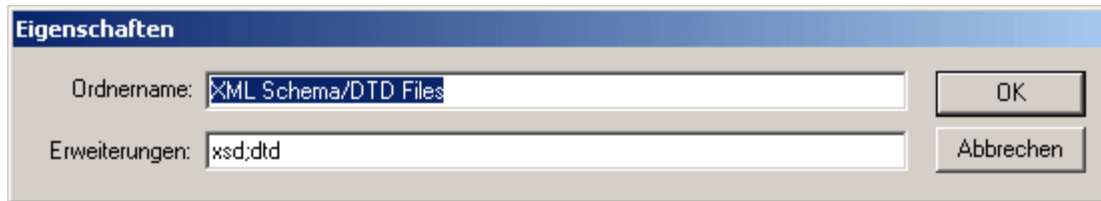
### Löschen von Dateien

Um eine Datei zu löschen, wählen Sie sie in der Projekt-Seitenleiste aus, klicken Sie auf die rechte Maustaste und wählen Sie im Kontextmenü den Befehl **Löschen**.

## Projekt und externe Verzeichnisse zu Projekt hinzufügen

### Projektordner zu Projekt hinzufügen

Mit dem Befehl **Projektordner zu Projekt hinzufügen** wird ein neuer Ordner zum aktuellen Projekt hinzugefügt. Bei Aufruf dieses Befehls wird das Dialogfeld "Eigenschaften" (*Abbildung unten*) angezeigt, in dem Sie den Namen und die Dateierweiterung für den Ordner eingeben können (Dateierweiterungen werden durch Semikola getrennt). Wenn eine Datei mit der für den Ordner definierten Dateierweiterung zum Projekt hinzugefügt wird, wird sie automatisch zu diesem Ordner hinzugefügt. Der neu hinzugefügte Projektordner wird zur Liste der Projektordner in der Projekt-Seitenleiste hinzugefügt.



Um einen Unterordner für einen beliebigen Projektordner zu erstellen, klicken Sie mit der rechten Maustaste auf den gewünschten Ordner und wählen Sie im Kontextmenü den Befehl **Projektordner hinzufügen**. Geben Sie im Dialogfeld "Eigenschaften" den Ordnernamen und die Dateierweiterung für den Ordner ein.

### Externes Verzeichnis zum Projekt hinzufügen

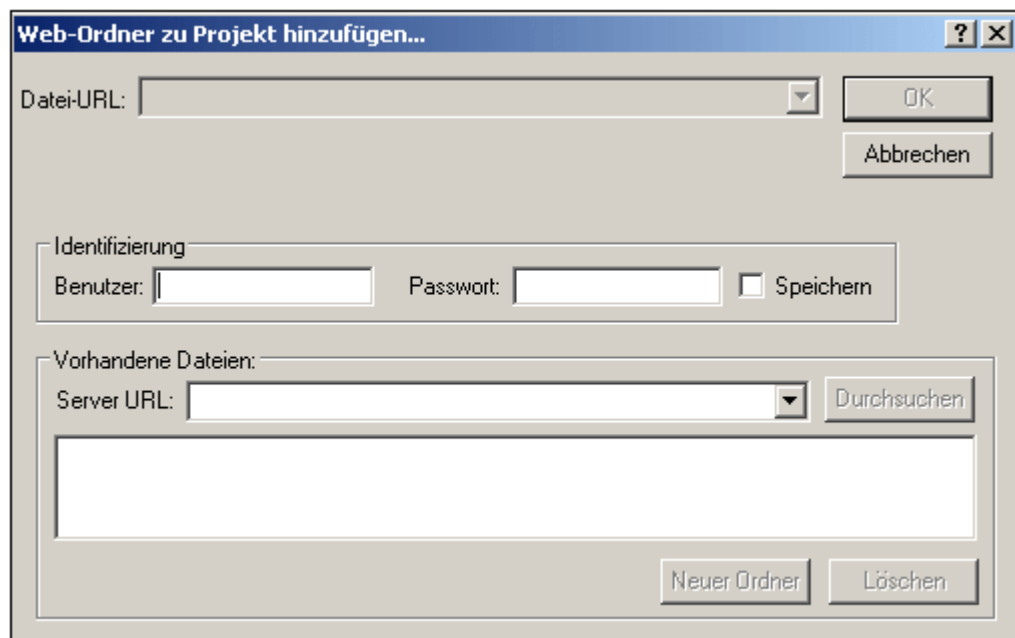
Mit dem Befehl **Externes Verzeichnis zum Projekt einfügen** wird ein neues externes Verzeichnis zum aktuellen Projekt hinzugefügt. Dabei wird ein lokales Verzeichnis oder ein Netzwerkordner mit seinem gesamten Inhalt zum aktuellen Projekt hinzugefügt. Der hinzugefügte externe Ordner kann erweitert oder reduziert werden. Um ein externes Verzeichnis als Unterverzeichnis zu einem Projektordner hinzuzufügen, klicken Sie mit der rechten Maustaste auf den Projektordner und wählen Sie im Kontextmenü den Befehl **Externes Verzeichnis einfügen**.

### Externes Webverzeichnis zum Projekt hinzufügen

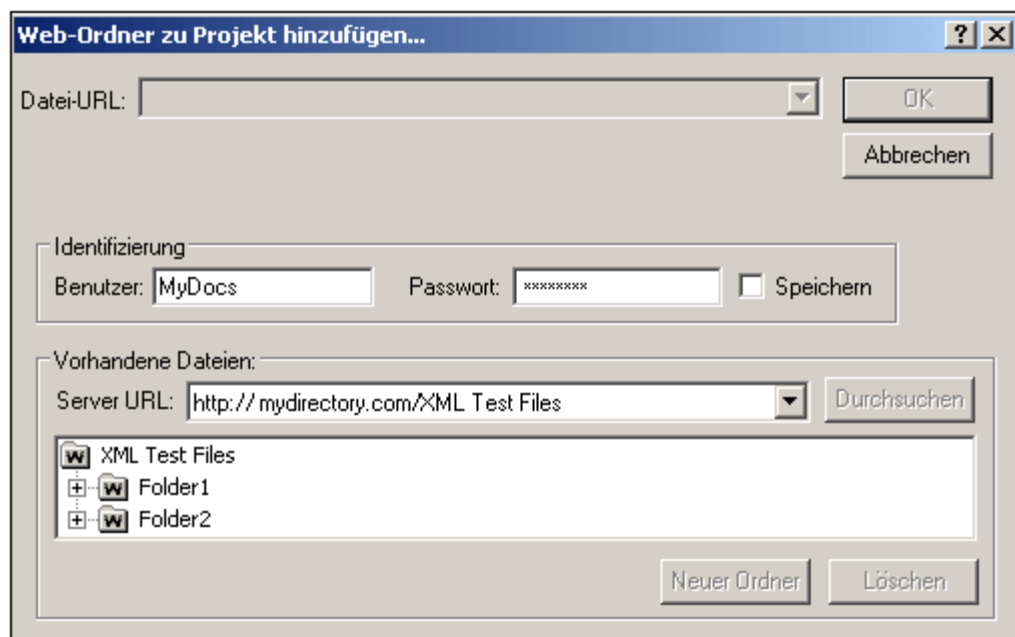
Mit dem Befehl **Externes Webverzeichnis zum Projekt hinzufügen** wird ein neues externes Verzeichnis über eine URL zum aktuellen Projekt hinzugefügt. Der hinzugefügte externe Ordner kann erweitert oder reduziert werden. Um ein externes Webverzeichnis als Unterverzeichnis zu einem Projektordner hinzuzufügen, klicken Sie mit der rechten Maustaste auf den Projektordner und wählen Sie im Kontextmenü den Befehl **Externes Webverzeichnis hinzufügen**.

Daraufhin wird das Dialogfeld "Webverzeichnis hinzufügen" (*Abbildung unten*) angezeigt. Gehen Sie folgendermaßen vor:

1. Klicken Sie auf das Feld "Server-URL", um die Server-URL einzugeben und geben Sie in die Felder "Benutzer" und "Passwort" die Login-ID und das Passwort ein.



2. Klicken Sie auf **Durchsuchen**, um eine Verbindung zum Server herzustellen und die Ordner darin anzuzeigen.



3. Klicken Sie auf den Ordner, den Sie zur Projektansicht hinzufügen möchten. Die **OK**-Schaltfläche steht erst dann zur Verfügung. Der Name des Ordners und die Server-URL werden nun im Feld "Datei-URL" angezeigt.
4. Klicken Sie auf **OK**, um den Ordner zum Projekt hinzuzufügen.
5. Klicken Sie auf das Plus-Symbol, um den Ordnerinhalt zu sehen.
6. Um festzulegen, welche Dateitypen für das Webverzeichnis angezeigt werden sollen, klicken Sie auf die rechte Maustaste, wählen Sie im Kontextmenü den Befehl **Eigenschaften** und geben Sie die gewünschten Dateierweiterungen ein.



**Drag-and-Drop**

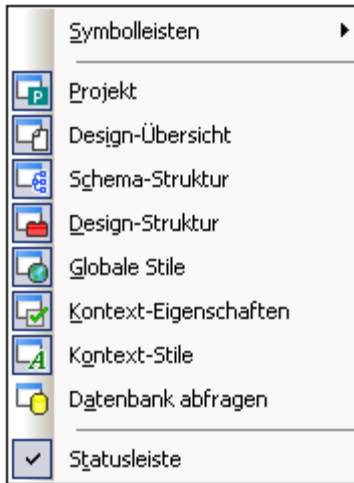
Sie können einen Ordner im Projektfenster in einen anderen Ordner oder an eine andere Stelle im selben Ordner ziehen. Außerdem können Ordner auch aus dem Windows Datei-Explorer in die Seitenleiste "Projekt" gezogen werden

**Löschen von Ordnern**

Um einen oder mehrere Ordner zu löschen, wählen Sie diese(n) in der Projekt-Seitenleiste aus, klicken Sie auf die rechte Maustaste und wählen Sie im Kontextmenü den Befehl **Löschen**.

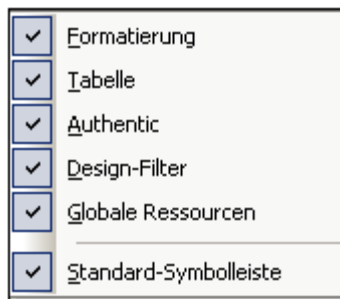
## 19.6 Menü "Ansicht"

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.



## 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.



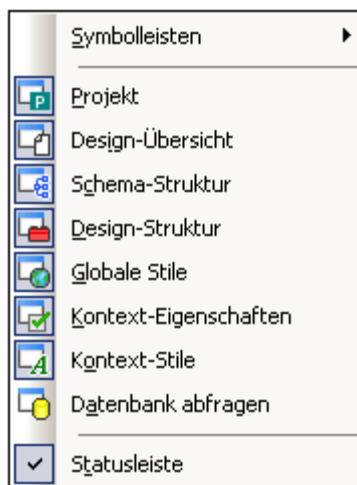
Wenn eine Symbolleiste mit einem Häkchen versehen ist, wird sie angezeigt. In der Abbildung oben werden alle Symbolleisten angezeigt. Um eine Symbolleiste ein- bzw. auszublenden, klicken Sie auf den entsprechenden Eintrag für die Symbolleiste. Eine ausführliche Beschreibung der Symbolleisten finden Sie im Abschnitt [Referenz | Symbolleisten](#).

### Statusleiste

Um die Statusleiste am unteren Rand des Applikationsfensters ein- oder auszublenden, klicken Sie auf den Eintrag "Statusleiste".

## 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, Authentic-Ansicht, Ausgabeansicht, kein Dokument geöffnet) definiert ist, wird so lange für dieser Ansichtskategorie beibehalten, bis sie geändert wird.

## 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                                                                                                                      |
|-------------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|    | <b>Immer nur eine Vorlage anzeigen</b>  | Zeigt nur die ausgewählte Vorlage an. Platzieren Sie den Cursor in eine Vorlage und klicken Sie, um nur diese Vorlage anzuzeigen. |
|    | <b>Alle Vorlagentypen anzeigen</b>      | Zeigt alle Vorlagen im SPS an (Hauptvorlage, globale Vorlagen, Design-Fragmente und Layout-Vorlagen) .                            |
|    | <b>Importierte Vorlagen anzeigen</b>    | Blendet importierte Vorlagen ein und aus.                                                                                         |
|    | <b>Hauptvorlage ein-/ausblenden</b>     | Blendet die Hauptvorlage ein und aus.                                                                                             |
|  | <b>Globale Vorlagen ein-/ausblenden</b> | Blendet die globalen Vorlagen ein und aus.                                                                                        |
|  | <b>Design-Fragmente ein-/ausblenden</b> | Blendet die Design-Fragmente ein und aus.                                                                                         |

Beachten Sie, dass diese Befehle auch als Schaltflächen in der Symbolleiste [Design-Filter](#) 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.

## 19.7 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](#) 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.

## 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"](#) 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.

## 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 **Einfügen | Restlicher Inhalt** ausgewählt wird.

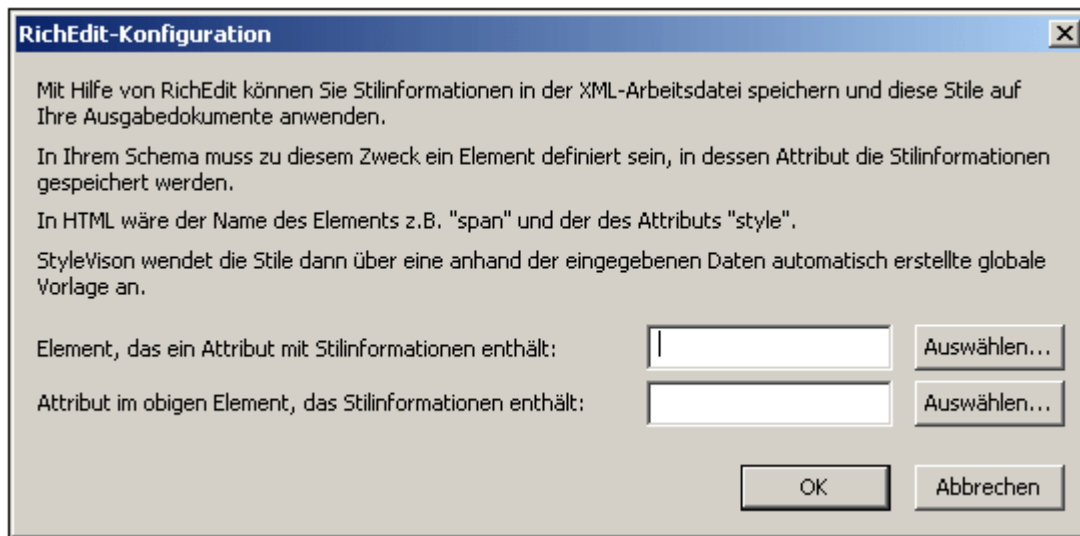
Verwenden Sie den Platzhalter (`restlicher Inhalt`) in Situationen, wenn ein Child-Element auf eine bestimmte Art verarbeitet und Vorlagen auf die gleichrangigen Elemente dieses Elements angewendet werden sollen. Es ist wichtig, dass Vorlagen auf gleichrangige Elemente angewendet werden, um zu vermeiden, 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.

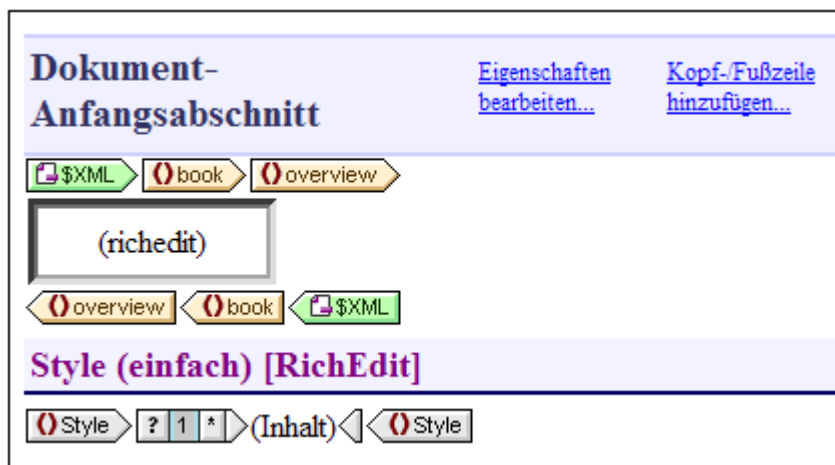


## RichEdit

Mit dem Befehl **RichEdit** wird eine RichEdit-Komponente an der Cursorposition eingefügt. Beim Einfügen der ersten RichEdit-Komponente wird das Dialogfeld "RichEdit-Konfiguration" (*Abbildung unten*) angezeigt. Diese RichEdit-Konfiguration gilt für alle RichEdit-Komponenten im Dokument, daher wird das Dialogfeld "RichEdit-Konfiguration" bei später eingefügt RichEdit-Komponenten nicht mehr angezeigt.



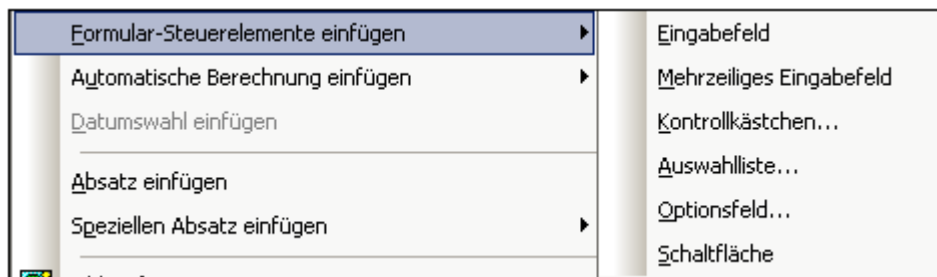
Geben Sie im Dialogfeld "RichEdit-Konfiguration" den Namen des Stilelements und seines Attributs an, das die RichEdit-Stileigenschaften enthalten soll. Sie können das erforderliche Element und Attribut aus der Schema-Struktur auswählen. Klicken Sie auf die entsprechende **Auswählen**-Schaltfläche, um die Schema-Struktur zu öffnen. Klicken Sie auf **OK** wenn Sie damit fertig sind. Daraufhin wird die RichEdit-Komponente erstellt (*siehe Abbildung unten*) und im Design wird eine nicht bearbeitbare RichEdit-Komponente angelegt, die den Namen des Stilelements (in der Abbildung unten *style*) hat.



Nähere Informationen zur RichEdit-Funktion und Ihrer Verwendung finden Sie unter [Flexibles Zuweisen von Textstilen in Authentic](#).

## 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](#)) (siehe *Abbildung unten*) angezeigt.



Eine Anleitung zum Erstellen der einzelnen Formular-Steuerelemente finden Sie im Abschnitt [Verwenden von Dateneingabeelementen](#). Nach Erstellung eines Formular-Steuerelements können Sie seine Eigenschaften bearbeiten. Wählen Sie das Element dazu aus und bearbeiten Sie die entsprechende Eigenschaft in der [Seitenleiste "Eigenschaften"](#).

Formular-Steuerelemente können auch ins Design eingefügt werden, indem Sie mit der rechten Maustaste auf den Einfügepunkt klicken und den Befehl **Einfügen | Inhalt** auswählen oder in der [Symbolleiste "Design-Element einfügen"](#) 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.

## DB-Steuerelement

Wenn Sie den Mauszeiger über den Befehl **DB-Steuerelement** platzieren, wird ein Untermenü angezeigt, das Befehle zum Einfügen von Steuerelementen in der Authentic-Ansicht enthält, mit Hilfe derer der Benutzer der Authentic-Ansicht in den Datensätzen in der Authentic-Ansicht navigieren und Datenbankabfragen durchführen kann. Diese Steuerelemente können in das Design eingefügt werden und werden im Authentic-Ansichtsdokument an der entsprechenden Stelle angezeigt.

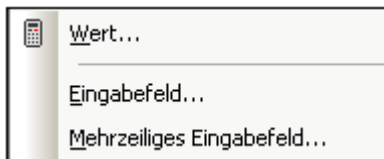
Dies ist die Liste der Befehle. Nähere Informationen zur Verwendung der Befehle finden Sie, wenn Sie auf die entsprechenden Links klicken.

- Navigation
- Navigation + Gehe zu
- Abfrageschaltfläche

Nähere Informationen zur Erstellung und Verwendung der Befehle finden Sie im Abschnitt [SPS Design Features für DB](#).

## Automatische Berechnung

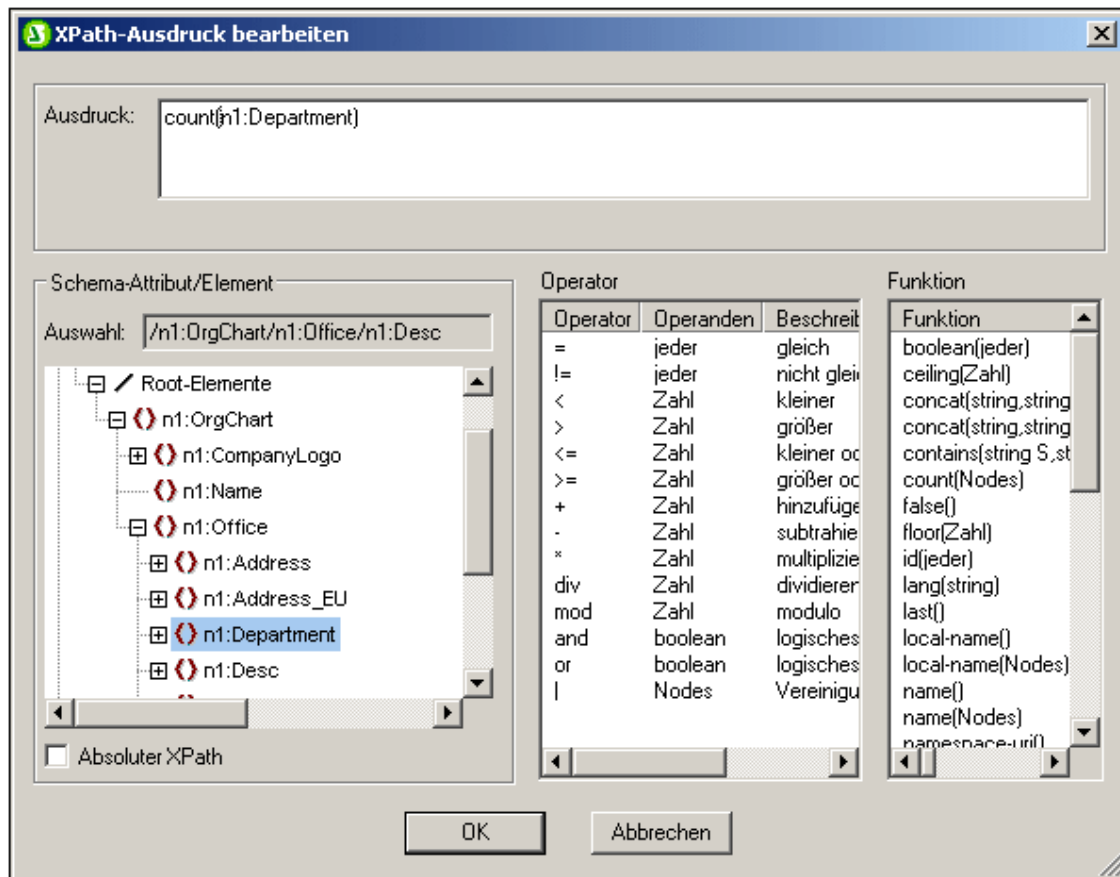
Bei einer **automatischen Berechnung** wird ein XPath-Ausdruck zum Berechnen eines Werts verwendet. Dieser Wert wird an der Stelle angezeigt, an der die automatische Berechnung eingefügt wird. Eine automatische Berechnung kann als Textwert, Eingabefeld oder mehrzeiliges Eingabefeld in das SPS eingefügt werden. Platzieren Sie den Cursor an die gewünschte Stelle und klicken Sie entweder auf die rechte Maustaste oder wählen Sie den Befehl über das 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"](#) auswählen.



Der Wert der automatischen Berechnung wird in der Authentic-Ansicht und 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"](#) (*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 "Absoluter XPath" deaktiviert ist) oder als absoluter Ausdruck beginnend mit den Dokument-Node eingefügt (wenn das Kontrollkästchen "Absoluter XPath" aktiviert ist).

Nachdem Sie den XPath-Ausdruck fertiggestellt haben, klicken Sie auf OK, um die automatische Berechnung in das Dokument einzufügen.

**Aktualisieren eines XML-Node mit einer automatischen Berechnung**

Ein Node in einem XML-Dokument kann mit dem Ergebnis einer automatischen Berechnung aktualisiert werden. Eine Beschreibung dazu finden Sie im Abschnitt [Aktualisieren von Nodes mit automatischen Berechnungen](#).

## Datumswahl

Mit dem Befehl **Datumswahl** wird eine Datumswahl an der aktuellen Cursorposition eingefügt. Der Befehl ist aktiv, wenn sich der Cursor innerhalb eines `xs:date` oder `xs:dateTime` Node befindet und wenn das Element als ( `Inhalt` ) oder als Eingabefeld erstellt wurde.

## Absatz, Spezieller Absatz

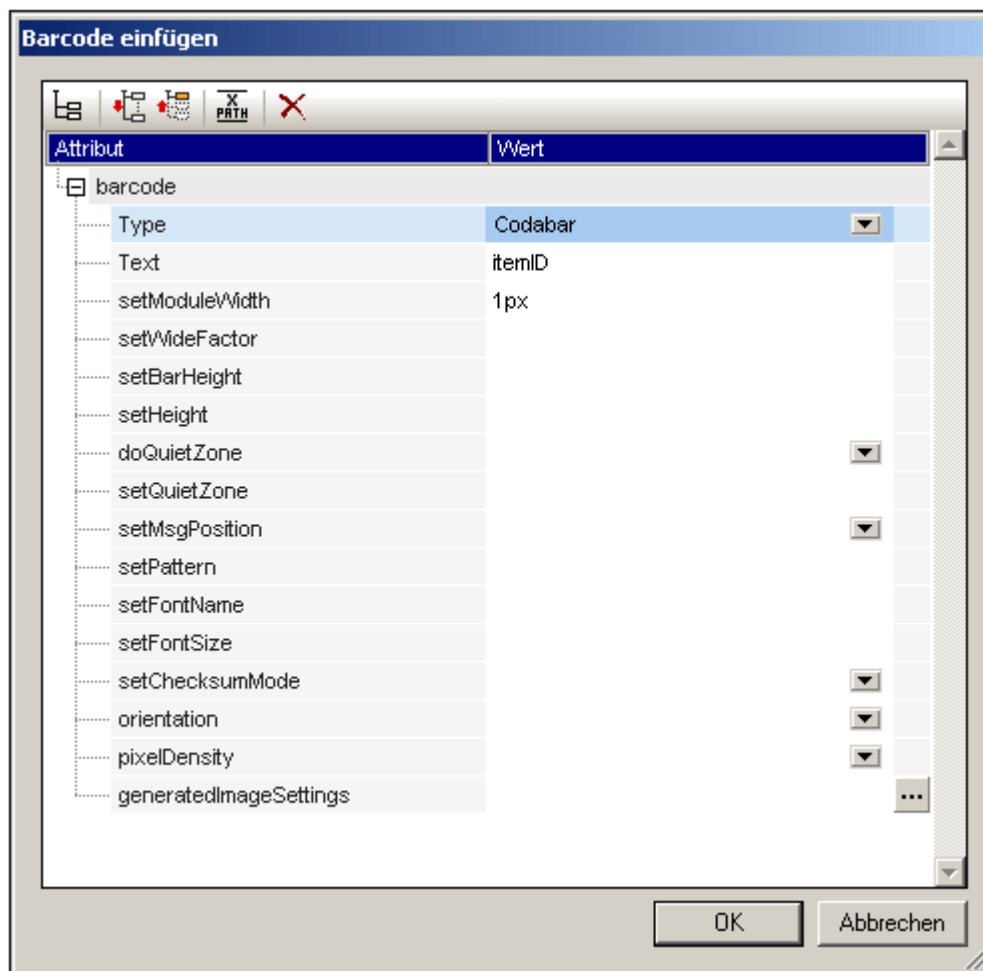
Mit dem Befehl **Absatz**  wird ein HTML-Absatzelement `<p>` rund um die ausgewählte Komponente eingefügt. Eine Komponente gilt als ausgewählt, wenn der gesamte Node ausgewählt ist (indem Sie auf einen seiner Tags klicken) oder wenn der statische Text ausgewählt ist. Wenn der Cursor in den statischen 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"](#) 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.

## Barcode

Der Befehl **Barcode einfügen** zeigt das Dialogfeld "Barcode einfügen" an (Abbildung unten).



Zwei Eigenschaften, *Type* und *Text*, sind Pflichtfelder, die anderen sind optional und/oder haben entsprechende Standardwerte. Nachdem Sie die Werte für die Pflichtfelder und weitere erwünschte Eigenschaften eingegeben haben, klicken Sie auf **OK**. Der Barcode, den Sie angegeben haben (wie dem für die unten abgebildete ISBN) werden in das Design eingefügt. Detaillierte Informationen über die verschiedenen Eigenschaften für Barcodes finden Sie im Abschnitt [Barcodes](#).



**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.

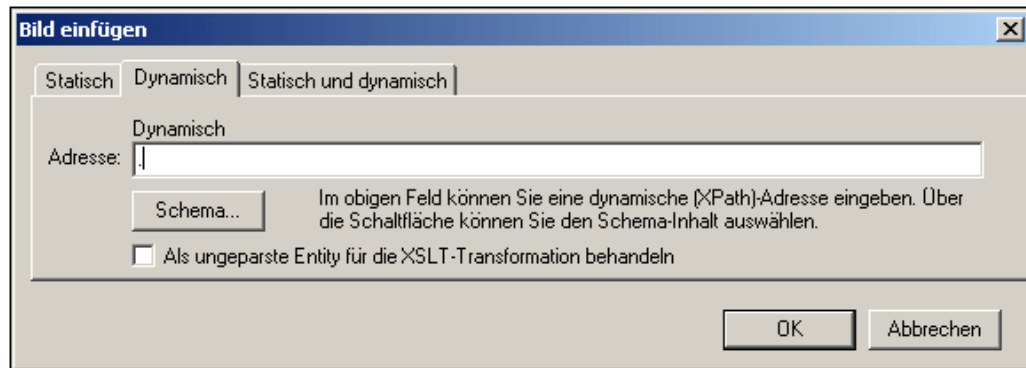


## Bild

Mit Hilfe des Befehls **Bild**  können Sie ein Bild einfügen, und zwar mittels einer Bildadresse, entweder aus der XML-Datei (dynamisch) oder durch direkte Eingabe des Pfads in das SPS (statisch).

Um ein Bild einzufügen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Einfügen | Bild** oder auf die entsprechende Symbolleiste-Schaltfläche. Daraufhin wird das Dialogfeld "Bild einfügen" (*Abbildung unten*) angezeigt. Ein Bild kann auch mittels der Schaltfläche **Bild einfügen** aus der [Symbolleiste "Design-Element einfügen"](#) eingefügt werden.



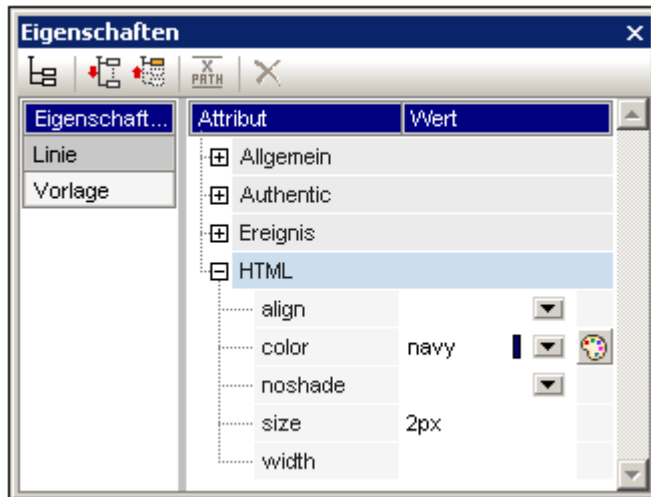
2. Wählen Sie das gewünschte Register aus (Statisch, Dynamisch oder Statisch und dynamisch) und geben Sie den Pfad zum Bild an und/oder geben Sie den XPath-Ausdruck ein, mit dem die Bildadresse im XML-Dokument gefunden wird. In der Abbildung oben sehen Sie, wie eine statische und dynamische Adresse eingegeben wird.

### Verwendung ungeparster Entities

Wenn das SPS auf einer DTD basiert und darin ungeparste Entities verwendet werden, dann kann für den dynamischen Teil einer Bildadresse die URI verwendet werden, die als Wert der ungeparsten Entity deklariert wurde. Nähere Informationen zur Verwendung von ungeparsten Entities finden Sie unter URIs von ungeparsten Entities.

## Horizontale Linie

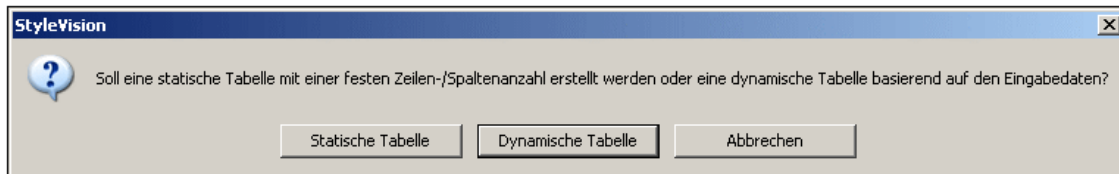
Mit Hilfe des Befehls **Horizontale Linie** wird eine horizontale Linie am Cursoreinfügapunkt eingefügt. Dieser Befehl steht nicht zur Verfügung, wenn eine SPS Komponente ausgewählt ist. Um Eigenschaften für die horizontale Linie zu definieren, wählen Sie die Linie im Design aus und wählen Sie in der 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).

## 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](#) und [Dynamische SPS-Tabellen](#).

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.

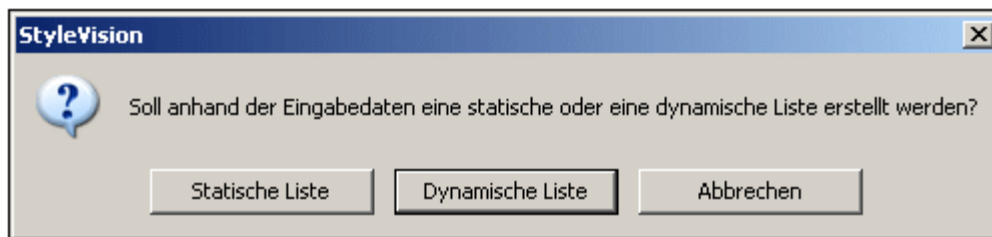
## Nummerierung und Aufzählungszeichen



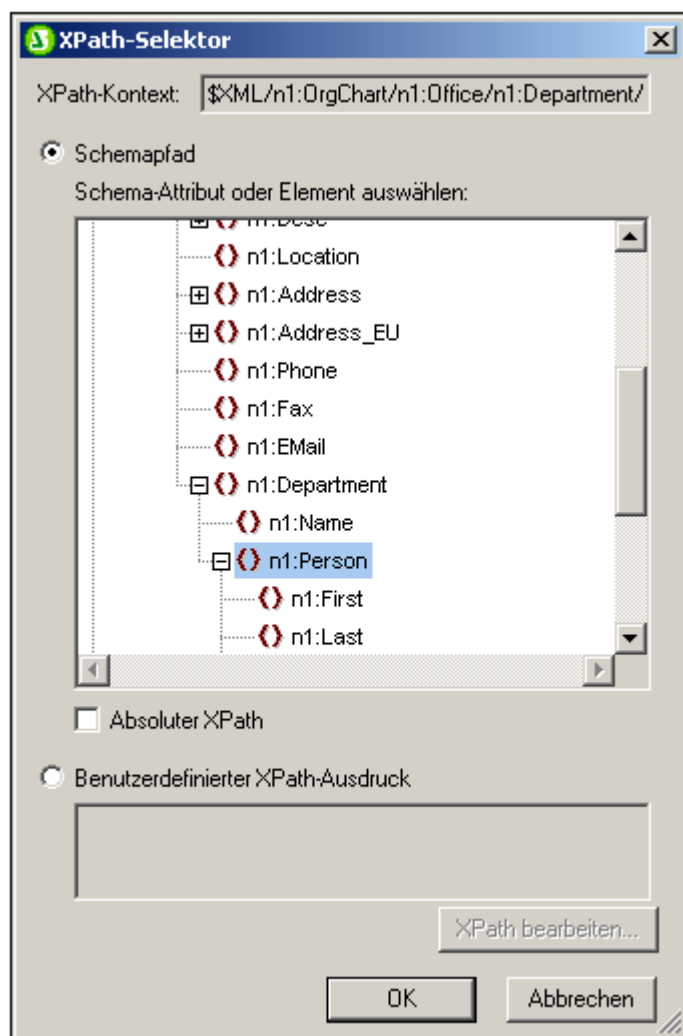
Mit Hilfe des Befehls **Nummerierung und Aufzählungszeichen** können Sie statische oder dynamische Listen erstellen. Die Einträge einer statischen Liste werden in das SPS eingegeben, während die Einträge dynamischer Listen aus gleichrangigen Nodes im XML-Dokument stammen.

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*).

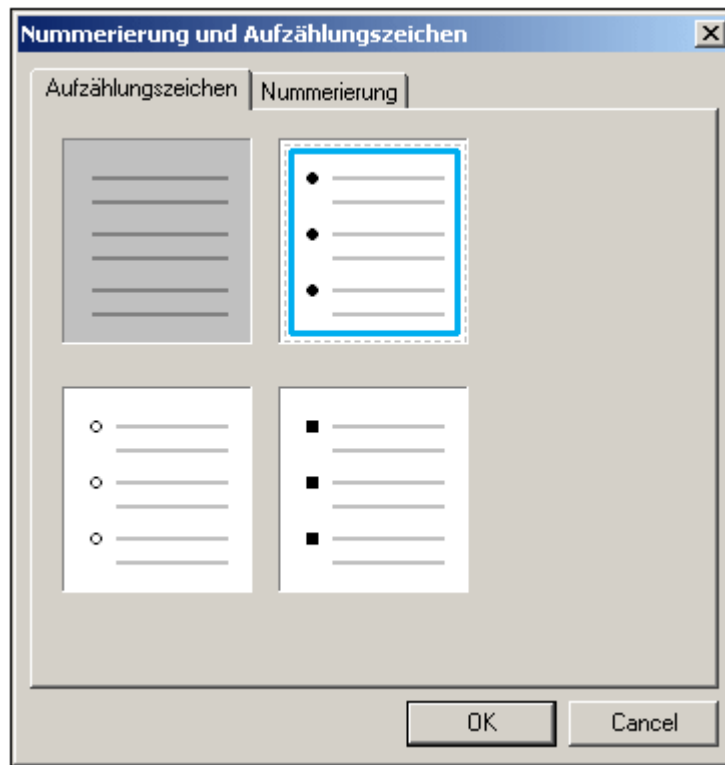


2. Wenn Sie auf **Statische Liste** klicken wird das in Schritt 3 beschriebene Dialogfeld angezeigt. Wenn Sie auf **Dynamische Liste** klicken, wird das Dialogfeld "XPath-Selektor" (*Abbildung unten*) angezeigt.
3. Beachten Sie, dass der XPath-Kontext im Dialogfeld "XPath-Selektor" der Kontext an der 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 `nl: Department` und der Node `nl: Person` wurde als der Node ausgewählt, der als Liste erstellt werden soll. D.h. der Inhalt der einzelnen `nl: 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 ausgewählten Variante zu erstellen).



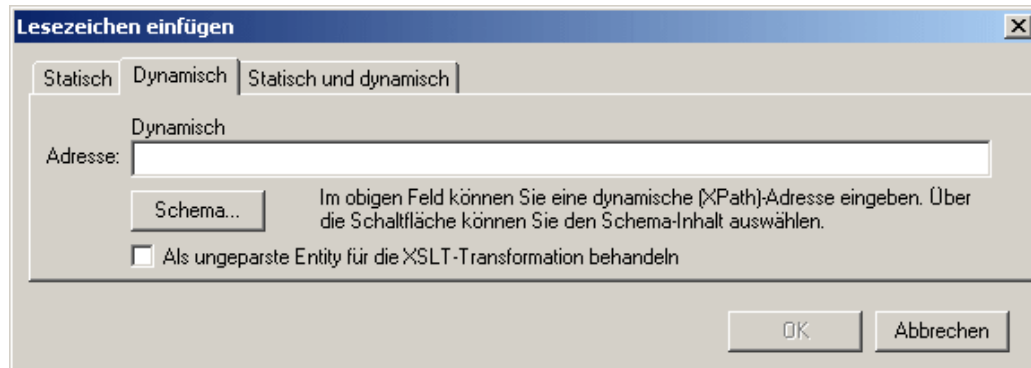
**Anmerkung:** Sie können die statische Liste auch durch Platzieren des Cursors an die gewünschte Stelle im Design und Klicken auf die Schaltfläche "Nummerierung und Aufzählungszeichen" in der [Symbolleiste "Design-Element einfügen"](#) erstellen. Eine dynamische Liste kann auch durch Ziehen eines Node aus der Schema-Struktur in das Design erstellt werden.

## Lesezeichen

Mit Hilfe des Befehls **Lesezeichen** können Sie ein Lesezeichen (Anchor) an jeder beliebigen Stelle im SPS einfügen. Ein Lesezeichen kann durch einen [Hyperlink](#) 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"](#), 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 Kontext-Node ist.
4. Klicken Sie auf OK. Das Lesezeichen wird definiert.

**Anmerkung:** Lesezeichen werden an der Stelle erstellt, die im Design definiert wird. Wenn es sich dabei um ein Element handelt, das sich wiederholt, so wird das Lesezeichen in jeder Instanz des sich wiederholenden Elements angelegt. Bei Definition eines statischen Namens, erhält jedes Lesezeichen denselben Namen. Daher empfiehlt es sich bei sich wiederholenden Elementen einen dynamischen Namen zu wählen, z.B. den Namen eines Child-Elements des Kontext-Node (des Elements, innerhalb von dem das Lesezeichen erstellt wird). Wenn der für den dynamischen Namen ausgewählte Node in mehreren Instanzen denselben Inhalt haben kann, können Sie die Eindeutigkeit des Lesezeichennamens dadurch sicherstellen, dass Sie zum Generieren des Namens die `generate-Id()` Funktion verwenden (*siehe Abbildung oben*). Zum Referenzieren des Lesezeichens kann dieselbe ID generiert werden, wie der `href` Wert eines [Hyperlinks](#).

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 **Lesezeichen bearbeiten**. Alternativ dazu können Sie auch in der Eigenschaftsleiste auf der Seite in der Eigenschaftsgruppe *Link* 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. Wählen Sie den Link alternativ dazu in der Eigenschaftsleiste aus und klicken Sie in der dazugehörigen Symbolleiste auf die Schaltfläche **Löschen**.

## Hyperlink



Mit Hilfe des Befehls **Hyperlink** können Sie einen Link einfügen, der von einer beliebigen Stelle im Ausgabedokument (HTML oder RTF) auf einen Anker innerhalb des Ausgabedokuments oder ein externes Dokument oder Dokumentfragment verweist. Beachten Sie, dass Links nur im Ausgabedokument erstellt werden; in der Authentic-Ansicht steht diese Funktion nicht zur Verfügung.

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"](#) eingefügt werden.
3. Definieren Sie im [Dialogfeld "Hyperlink einfügen"](#), 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](#).
  - 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 für den Node, den Sie über die Schaltfläche "Schema" auswählen, als relativ zum aktuell ausgewählten Dokument eingegeben.

### Verwendung von ungeparsten Entities

Falls Sie eine DTD verwenden, können Sie für den dynamischen Teil einer Hyperlink-Adresse eine 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.

### **Entfernen 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**

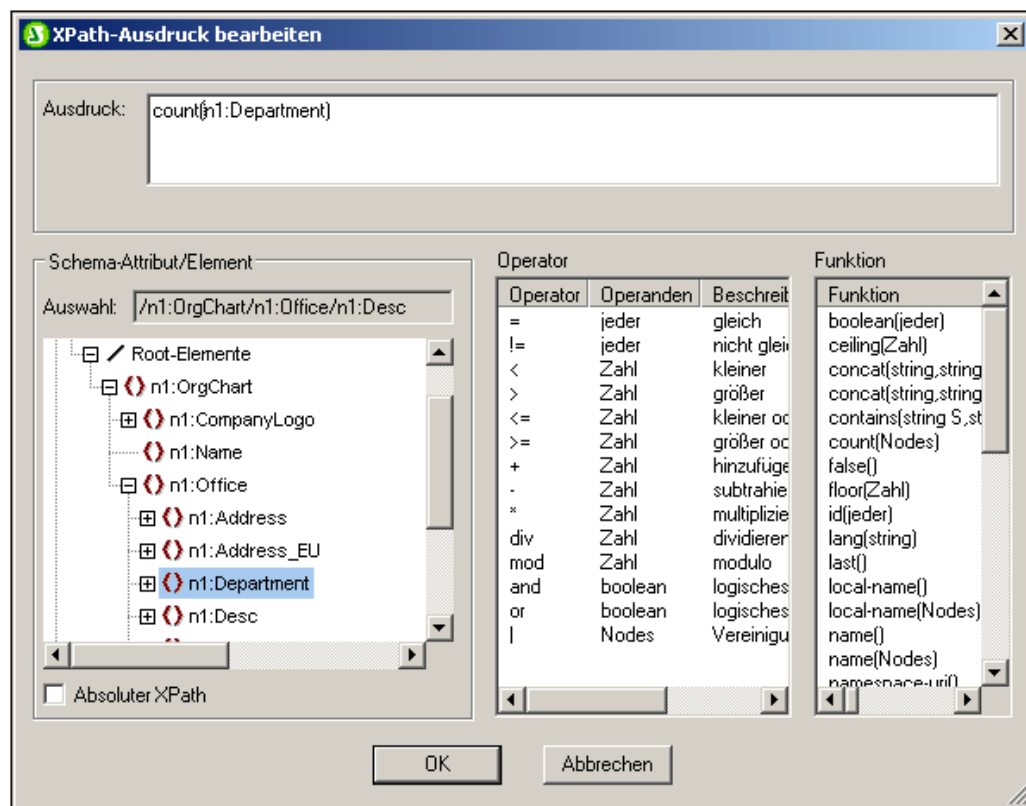
Um einen Hyperlink zu löschen, wählen Sie ihn im Design aus und drücken Sie die **Entf**-Taste. Wählen Sie den Link alternativ dazu in der Eigenschaftsleiste aus und klicken Sie in der dazugehörigen Symbolleiste auf die Schaltfläche **Löschen**.

## 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"**, 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 "Absoluter XPath" deaktiviert ist) oder als absoluter Ausdruck beginnend mit den Dokument-Node eingefügt (wenn das Kontrollkästchen "Absoluter XPath" aktiviert 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.

### **Ausgabe-basierte Bedingung einfügen**

Mit diesem Befehl wird eine Ausgabe-basierte Bedingung an der Cursorposition oder rund um die ausgewählte Komponente eingefügt. Jede Verzweigung der Bedingung steht für eine einzige Ausgabe (Authentic-Ansicht, RTF, oder HTML). Um festzustellen, welche Verzweigung für welche Ausgabe steht, positionieren Sie den Mauszeiger über den Verzweigungs-Tag oder überprüfen Sie den XPath-Ausdruck der ausgewählten Verzweigung (klicken Sie in der Seitenleiste "Eigenschaften" im Eintrag Bedingungsverzweigung auf die [Bearbeiten-Schaltfläche](#)). Wenn die Ausgabe-basierte Bedingung an einem Cursoreinfügepunkt erstellt wurde, sind alle Verzweigungen leer und der Inhalt muss für jede einzelne Verzweigung eingefügt werden. Wenn die Ausgabe-basierte Bedingung rund um eine Komponente erstellt wurde, enthält jede Verzweigung diese Komponente. Nähere Informationen zu Ausgabe-basierten Bedingungen finden Sie unter [Ausgabe-basierte Bedingungen](#). Sie können Ausgabe-basierte Bedingungen auf dieselbe Art bearbeiten, verschieben und löschen, wie Standard-Bedingungen.

### **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 oben*) angezeigt, in dem Sie den Ausdruck bearbeiten können. Klicken Sie anschließend auf OK.

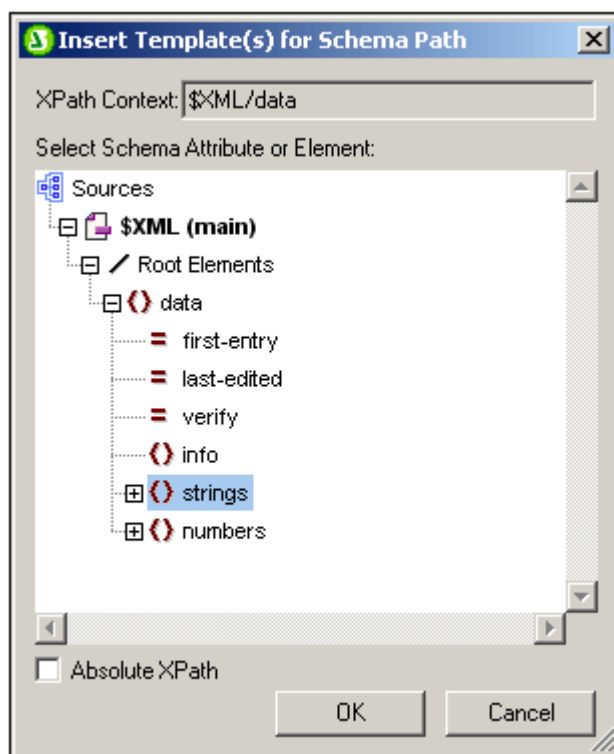
### **Hinzufügen von Verzweigungen, Ändern der Reihenfolge von Verzweigungen und Löschen von Verzweigungen**

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.

## Vorlage

Mit dem Befehl **Vorlage** wird am Cursoreinfügekpunkt 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.



3. Das XPath-Kontextfeld enthält den Kontext-Node des Cursor-Einfügekpunkts und der Kontext-Node für die Vorlage. Wählen Sie den Node, für den Sie die Vorlage erstellen möchten. In der Abbildung oben wurde der Node `strings` als der Node ausgewählt, für den die Vorlage erstellt werden soll.
4. Klicken Sie zum Fertigstellen auf **OK**.

Eine leere Vorlage für den ausgewählten Node wird nun erstellt (In der Abbildung unten wurde eine leere Vorlage für den Node `strings` erstellt).



## Benutzerdefinierte Vorlage

Mit dem Befehl **Benutzerdefinierte Vorlage** wird am Cursoreinfügapunkt eine leere Vorlage eingefügt, die einen vom Benutzer mit einem XPath-Ausdruck definierten Node auswählt. Gehen Sie beim Einfügen einer benutzerdefinierten Vorlage folgendermaßen vor:

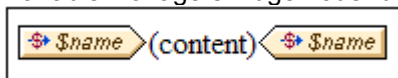
1. Platzieren Sie den Cursor an die Stelle im Design, an der die Vorlage eingefügt werden soll.
2. Klicken Sie auf den Befehl **Einfügen | Benutzerdefinierte Vorlage**. Daraufhin wird das Dialogfeld [XPath-Ausdruck bearbeiten](#) angezeigt.
3. Geben Sie den XPath-Ausdruck zur Auswahl des gewünschten 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](#).

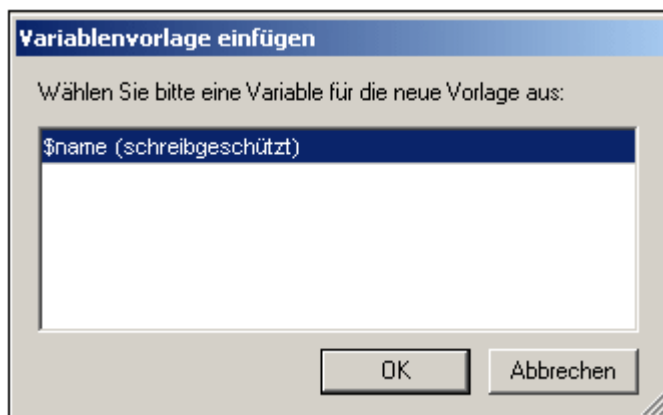
## Variablenvorlage

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.



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 den Befehl **Einfügen | Variablenvorlage**. Daraufhin wird das Dialogfeld "Variablenvorlage einfügen" angezeigt (*Abbildung unten*).



3. Das Dialogfeld enthält eine Liste aller im SPS definierten [benutzerdefinierten Parameter und Variablen](#). Wählen Sie die Variable aus, für die Sie eine Variablenvorlage hinzufügen möchten.
4. Klicken Sie zum Fertigstellen auf **OK**.

## 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.

## 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"](#) 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](#).



## 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.

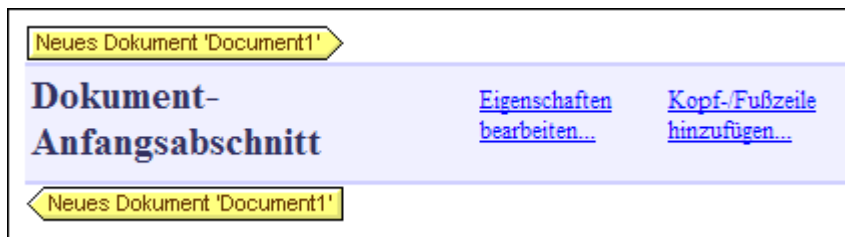
Dies ist die Liste der Befehle. Nähere Informationen zur Verwendung der Befehle finden Sie, wenn Sie auf die entsprechenden Links klicken.

- [Inhaltsverzeichnis einfügen](#)
- [Inhaltsverz.-Markierung](#)
- [Inhaltsverz.-Markierung \(Assistent\)](#)
- [Inhaltsverzeichnis-Referenz](#)
- [Inhaltsverz.-Referenz | Eingabetext / Punktierte Linie / Seitenverweis](#)
- [Hierarchische Nummerierung](#)
- [Sequenzielle Nummerierung](#)
- [Ebene](#)
- [Ebenenreferenz](#)
- [Vorlage dient als Ebene](#)

**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.

## 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-Dokumentausrage](#).

## Seite / Spalte / Dokumentabschnitt

Mit Hilfe des Befehls **Seite / Spalte / Dokumentabschnitt** können Sie bei der Ausgabe auf paginierten Medien einen Seitenumbruch (HTML-Ausdrucke und RTF-Ausgabe) und Seitennummern (RTF--Ausgabe) einfügen. Dies ist nur am Cursoreinfügekpunkt möglich.

### Neue Seite

Klicken Sie auf **Neue Seite**, um am Cursoreinfügekpunkt einen Seitenumbruch einzufügen. Der Seitenumbruch wird als gestrichelte Linie, die quer durch das ganze Design-Fenster verläuft, angezeigt. Während in der HTML-Ausgabe ein Seitenumbruch keine Auswirkung auf die Browser-Ansicht hat, wird beim Ausdruck der Browser-Ansicht der HTML-Datei ein Seitenumbruch gedruckt. In der PDF-, Word 2007+- und der RTF-Ausgabe wird an den angegebenen Stellen ein Seitenumbruch eingefügt.

### Seitennummer

Klicken Sie auf **Seite | Nummer**, um die aktuelle Seitennummer in der RTF-Ausgabe einzufügen. Die Seitennummer wird als Block angezeigt (d.h. als separate Linie) oder innerhalb einer Zeile (d.h. in den Text des Dokuments eingebettet) angezeigt, je nachdem, wo im Dokument die Seitennummer eingefügt wurde. Wenn die Seitennummer z.B. innerhalb eines Absatzelements eingefügt wird, so erscheint sie in einer Zeile innerhalb des Absatzes. Wenn die Seitennummer aber z.B. zwischen zwei Elemente eingefügt wird, wird sie auf einer eigenen Zeile angezeigt.

### Gesamtseitenzahl

Klicken Sie auf **Seite | Gesamt**, um in der PDF-Ausgabe die Gesamtseitenzahl einzufügen. Die Gesamtseitenzahl kann an einer beliebigen Stelle im Dokument-Design eingefügt werden, u.a. auch in Kopf- und Fußzeilen. Diese Option eignet sich besonders bei der Seitennummerierung. So kann die Gesamtseitenzahl z.B. folgendermaßen in einer Kopfzeile eingefügt werden:

Seite: (Seitennummer)/(Seite ges.) . In der Ausgabe würde diese folgendermaßen aussehen: Seite: 1/25.

### Neue Spalte

Die Anzahl der Spalten, die eine Seiten in einem bestimmten Abschnitt haben darf, wird in den Seiteneigenschaften dieses Abschnitts definiert. In der Ausgabe werden die Spalten auf einer Seite, die mehrere Spalten enthält, der Reihe nach mit Text befüllt. Damit Text zwingend in eine neue Spalte gesetzt wird, müssen Sie einen Spaltenumbruch (neue Spalte) einfügen. Klicken Sie dazu mit der rechten Maustaste auf die Stelle, an der Sie den Spaltenumbruch einfügen möchten und wählen Sie im Kontextmenü den Menübefehl **Neue Spalte**.

### Neuer Dokumentabschnitt

Ein Dokument besteht aus einem Anfangsabschnitt und optional dazu weiteren zusätzlichen Abschnitten. Jeder [Abschnitt](#) hat seine eigenen Seiteneigenschaften. Um einen neuen Abschnitt in ein Dokument einzufügen, platzieren Sie den Cursor an die gewünschte Stelle im Design und wählen Sie im Kontextmenü den Befehl **Neuer Dokumentabschnitt**.

### Löschen von Seitenumbrüchen

Um einen Seitenumbruch zu löschen, wählen Sie den Platzhalter aus und klicken Sie auf **Löschen**.

## Benutzerdefiniertes Element

Wenn Sie die Maus über den Befehl **Einfügen | Benutzerdefiniertes Element** platzieren, wird ein Untermenü angezeigt, das Befehle zum Einfügen eines [benutzerdefinierten Elements](#) oder eines [benutzerdefinierten XML-Textblocks](#) enthält. Eine Beschreibung zur Verwendung dieser beiden Komponenten finden Sie im Abschnitt [SPS-Datei: Inhalt | Benutzerdefinierte Elemente, XML-Textblöcke](#).

## 19.8 Menü "Einschließen in"

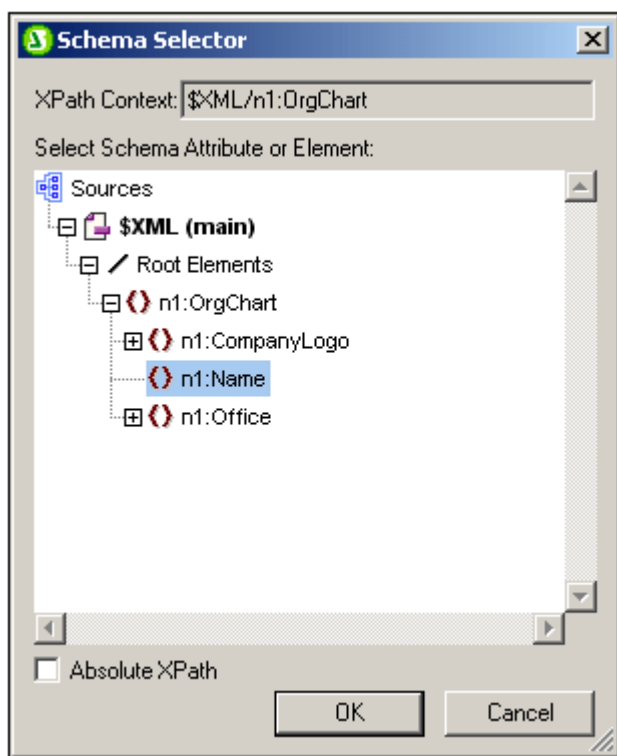
Das Menü **Einschließen in** enthält Befehle, über die Sie eine Auswahl im Design in die verschiedensten Design-Komponenten einschließen können. Einige dieser Befehle stehen in Form von [Symbolleisten-Schaltflächen](#) zur Verfügung, über die Sie die Komponente in das Design einfügen können (die entsprechenden Befehle stehen auch im Menü [Einfügen](#) 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.

## 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.

## 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](#) 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: Inhalt | Benutzerdefinierte Vorlagen und Design-Fragmente | Variablenvorlagen](#).

## 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](#) angezeigt.
3. Wählen Sie aus der Liste im Dialog die gewünschte Variable aus.
4. Klicken 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](#).



## 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"](#) eingefügt werden.

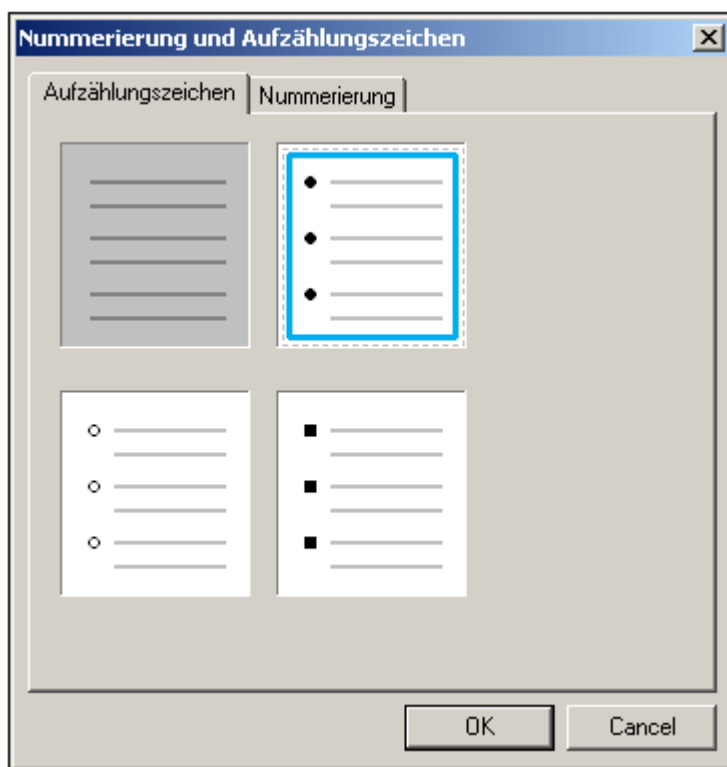
Mit dem Befehl **Speziellen Absatz einfügen** können Sie dem ausgewählten Node ein vordefiniertes Format zuweisen. Das verfügbare vordefinierte Format kann auch über die Auswahlliste in der Symbolleiste ausgewählt werden.

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.

## 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"](#) erstellen.

## Lesezeichen und Hyperlinks

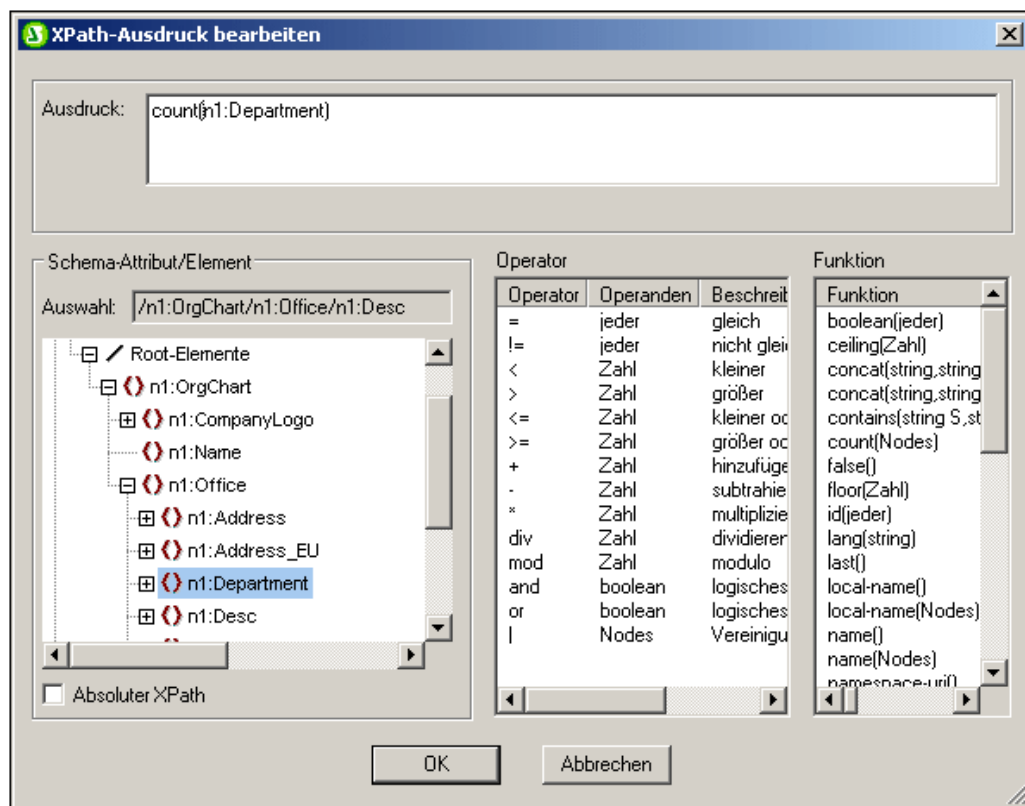
Die Befehle **Einschließen in | Lesezeichen** und **Einschließen in | Hyperlink** sind aktiviert, wenn im SPS-Design Text oder eine Komponente ausgewählt ist. Mit Hilfe dieser Befehle kann rund um die Auswahl ein Lesezeichen bzw. ein Hyperlink erstellt werden. Nähere Informationen zum Erstellen und zur Verwendung von Lesezeichen und Hyperlinks finden Sie im Abschnitt [Komplexe Funktionen | Inhaltsverzeichnis, Referenzieren, Lesezeichen](#).

## Bedingung, ausgabebasierte Bedingung

Mit Hilfe des Befehls **Bedingung** können Sie eine Bedingung am Cursoreinfügekpunkt 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"**, 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 "Absoluter XPath" deaktiviert ist) oder als absoluter Ausdruck beginnend mit den Dokument-Node eingefügt (wenn das Kontrollkästchen "Absoluter XPath" aktiviert 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.

### **Ausgabe-basierte Bedingung einfügen**

Mit diesem Befehl wird eine Ausgabe-basierte Bedingung an der Cursorposition oder rund um die ausgewählte Komponente eingefügt. Jede Verzweigung der Bedingung steht für eine einzige Ausgabe (Authentic-Ansicht, RTF, oder HTML). Um festzustellen, welche Verzweigung für welche Ausgabe steht, positionieren Sie den Mauszeiger über den Verzweigungs-Tag oder überprüfen Sie den XPath-Ausdruck der ausgewählten Verzweigung (klicken Sie in der Seitenleiste "Eigenschaften" im Eintrag Bedingungsverzweigung auf die [Bearbeiten-Schaltfläche](#)). Wenn die Ausgabe-basierte Bedingung an einem Cursoreinfügepunkt erstellt wurde, sind alle Verzweigungen leer und der Inhalt muss für jede einzelne Verzweigung eingefügt werden. Wenn die Ausgabe-basierte Bedingung rund um eine Komponente erstellt wurde, enthält jede Verzweigung diese Komponente. Nähere Informationen zu Ausgabe-basierten Bedingungen finden Sie unter [Ausgabe-basierte Bedingungen](#). Sie können Ausgabe-basierte Bedingungen auf dieselbe Art bearbeiten, verschieben und löschen, wie Standard-Bedingungen.

### **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 oben*) angezeigt, in dem Sie den Ausdruck bearbeiten können. Klicken Sie anschließend 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.

## Inhaltsverz.-Markierungen und Inhaltsverz.-Ebenen

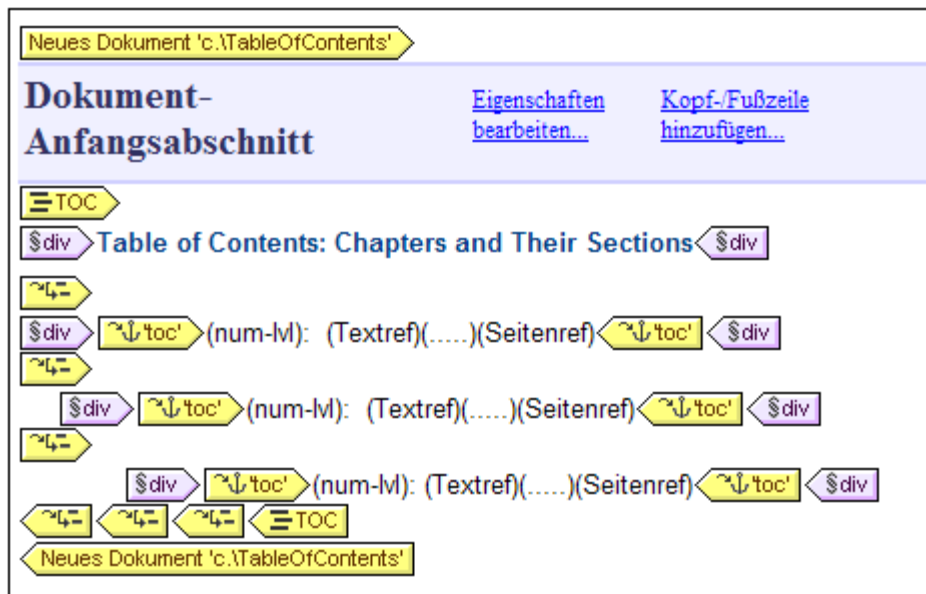
Wenn eine Komponente im Design ausgewählt ist, kann sie in eine oder mehrere Inhaltsverzeichnis-Komponenten eingeschlossen werden. Im Folgenden sehen Sie die Liste der Inhaltsverzeichnis-Befehle. Nähere Informationen zu ihrer Verwendung finden Sie unter den entsprechenden Links.

- [Inhaltsverz.-Markierung](#)
- [Inhaltsverz.-Markierung \(Assistent\)](#)
- [Ebene](#)
- [Ebenenreferenz](#)

**Anmerkung:** Diese Befehle stehen auch, je nachdem, an welche Stelle im Design Sie klicken, in Kontextmenüs zur Verfügung.

## 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-Dokumentaushabe](#).

## Benutzerdefiniertes Element

Mit dem Befehl **Einschließen in | Benutzerdefiniertes Element** wird rund um die Auswahl im Design ein [benutzerdefiniertes Element](#) erstellt. Eine Anleitung zur Verwendung von benutzerdefinierten Elementen finden Sie im Abschnitt [SPS-Datei: Inhalt | Benutzerdefinierte Elemente](#).



## 19.9 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](#) und [HTML-Eigenschaften für Tabellen und ihre Unterkomponenten](#) definiert.

Die Tabellenbefehle stehen im Menü "Tabelle" zur Verfügung (*siehe Liste unten*) und in Form von Symbolen in der [Symbolleiste "Tabelle"](#). Welche Tabellenbefehle jeweils verfügbar sind, hängt von der Position des Cursors ab. Eine Tabelle kann überall im SPS durch Auswahl des Befehls [Tabelle einfügen](#) 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"](#) und/oder der [Seitenleiste "Eigenschaften"](#) die gewünschte Eigenschaft für diese Tabellenkomponente.

Die folgenden Befehle stehen im Menü "Tabelle" zur Verfügung:

- [Tabelle einfügen, Tabelle löschen](#)
- [Kopf- / Fußzeilen für Tabellen einfügen](#)
- [Zeile/Spalte anhängen/einfügen](#)
- [Zeile, Spalte löschen](#)
- [Zelle links, rechts, unten, oben verbinden](#)
- [Zelle horizontal, vertikal teilen](#)
- [Zellenumrandung anzeigen / Tabellen-Markup-Code anzeigen](#)
- [Tabelleneigenschaften](#)
- [Vertikale Ausrichtung des Zelleninhalts](#)

### 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.

## 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.

- 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](#)

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.

## 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 platziert 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.

## 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.

## 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.

## 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.

## 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.

## 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.



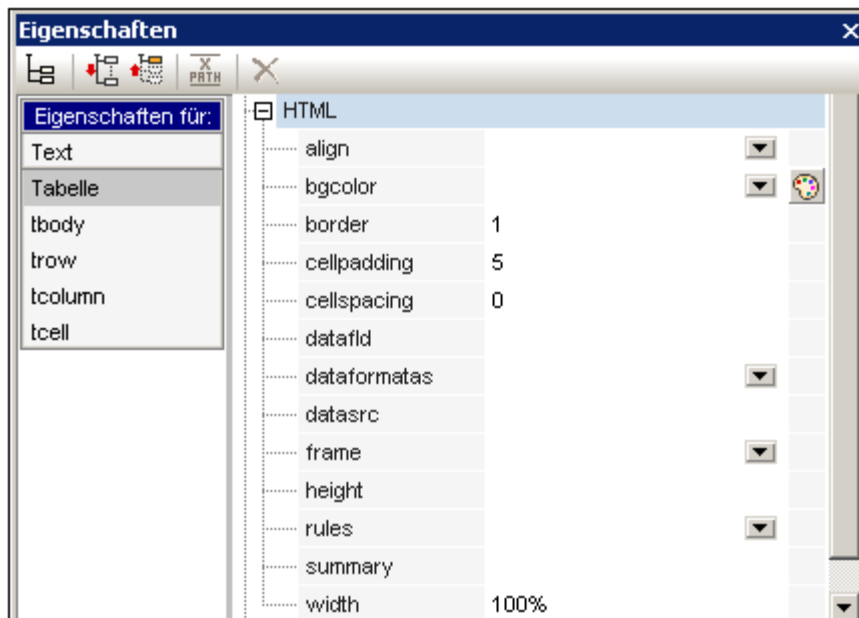
Mit dem Befehl **Tabellen-Markup-Code anzeigen** können die blauen Spalten- und Zeilenmarkierungen ein- und ausgeblendet werden.



## Tabelleneigenschaften



Der Befehl **Tabelleneigenschaften** ist aktiv, wenn der Cursor sich innerhalb einer [statischen](#) oder [dynamischen Tabelle befindet](#). Bei Aufruf des Befehls wird die Seitenleiste "Eigenschaften" angezeigt, wobei die Komponente "Tabelle" ausgewählt ist (*Abbildung unten*).

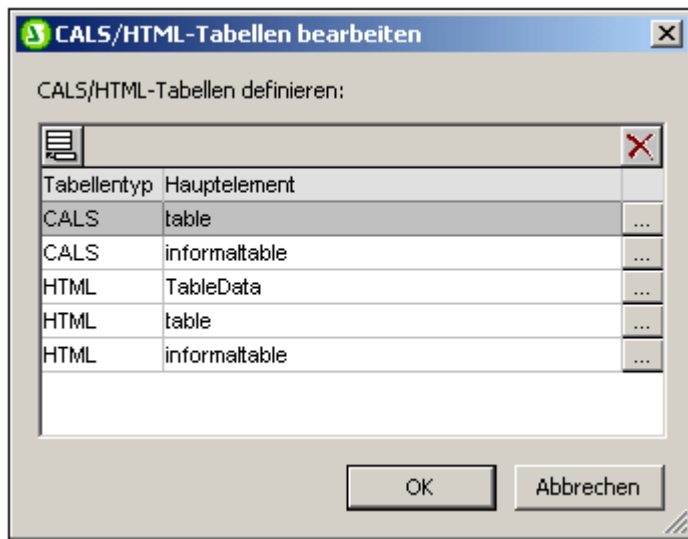


Sie können Die Eigenschaften der Tabelle nun bearbeiten. Klicken Sie anschließend auf OK.

## CALS/HTML-Tabellen bearbeiten

Mit dem Befehl **CALS/HTML-Tabellen bearbeiten** können Datenstrukturen im XML-Dokument, die dem CALS- bzw. dem HTML-Tabellenmodell entsprechen, in der Ausgabe als Tabellen generiert werden. Der Tabellen-Markup-Code in den Ausgabeformaten wird direkt aus dem XML-Dokument übernommen. Sie können jedoch über das SPS zusätzliche Tabellenformatierungsstile hinzufügen.

Bei Auswahl dieses Befehls wird das Dialogfeld "CALS/HTML-Tabellen" (*Abbildung unten*) aufgerufen.



Nähere Informationen zu CALS/HTML-Tabellen finden Sie im Abschnitt [Tabellen](#).

## 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.

## 19.10 Menü "Authentic"

Das Menü **Authentic** enthält Befehle für die folgenden Zwecke:

- zum Anpassen von Aspekten der Authentic-Ansicht eines XML-Dokuments, das unter Verwendung des SPS angezeigt wird.
- zum Bearbeiten von Dokumenten in der Authentic-Vorschau von StyleVision.

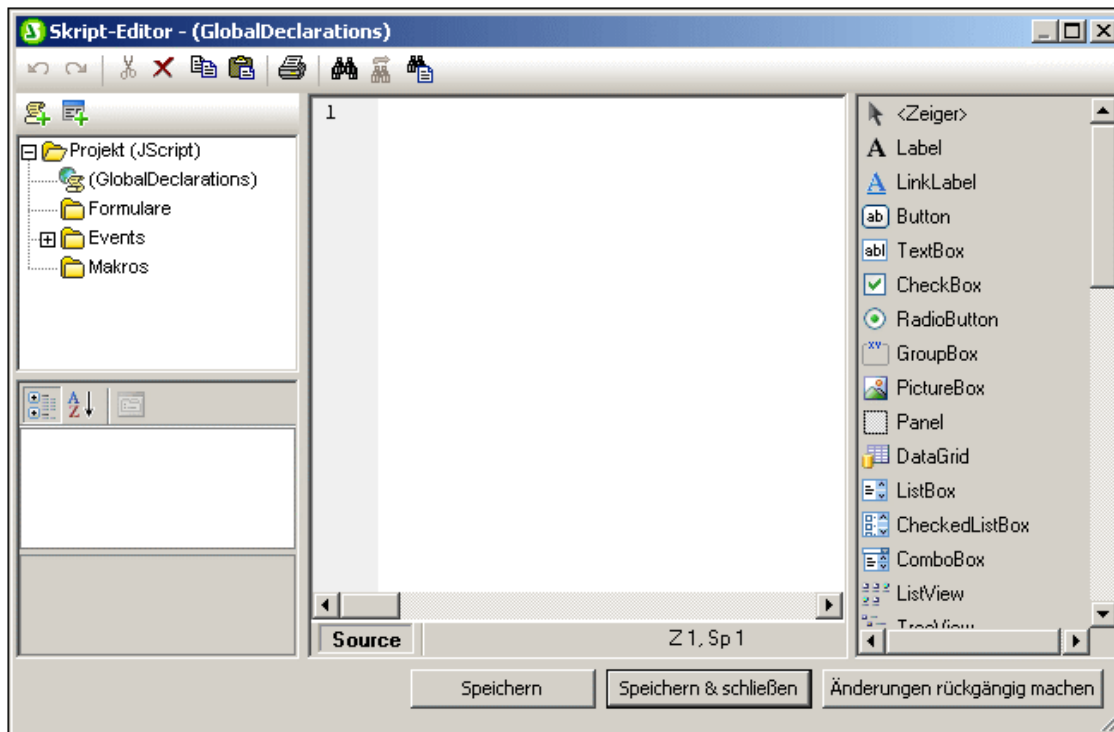
Nachfolgend sind die Befehle im Menü "Authentic" aufgelistet:

- [Authentic Skripts bearbeiten](#)
- [Benutzerdefinierte Symbolleisten-Schaltflächen](#)
- [Makro-Referenzen überprüfen](#)
- [Datumswahl automatisch hinzufügen](#)
- [DB-Steuerelemente automatisch hinzufügen](#)
- [Neu laden, XML validieren](#)
- [Neue Zeile mit XML-Daten für Bearbeitung auswählen](#)
- XML-Signatur
- [XML Entities definieren](#)
- [Markup-Befehle](#)
- [Zeilenbefehle \(einer dynamischen Tabelle\)](#)

Jeder dieser Befehle wird in den Unterabschnitten dieses Abschnitts näher beschrieben.

## Authentic Skripts bearbeiten

Mit diesem Befehl wird der StyleVision Skript-Editor (Abbildung unten) aufgerufen, in dem Sie Formulare, Events und Makros für die Verwendung in der Authentic-Ansicht erstellen können.

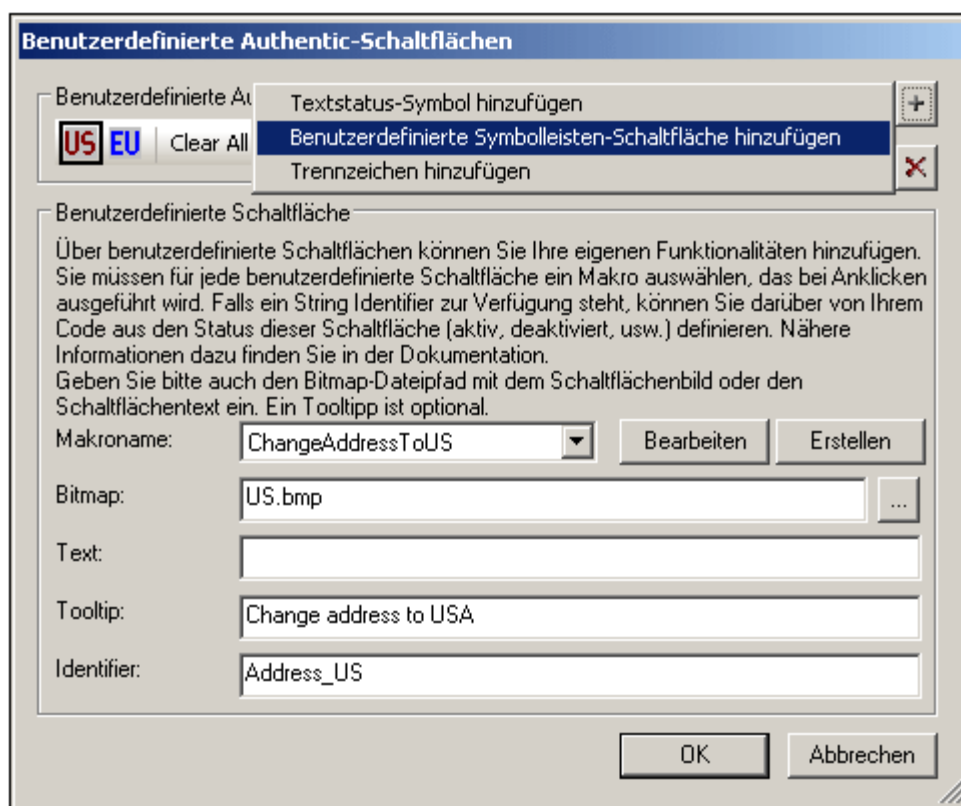


Eine Übersicht über die Verwendung von Skripts in der Authentic-Ansicht finden Sie im Abschnitt [Authentic-Skripterstellung](#). Eine Beschreibung des Skript-Editors finden Sie in der [Referenz für Programmierer](#) im Abschnitt [Skripterstellung](#).

## Benutzerdefinierte Symbolleisten-Schaltflächen

Wenn Sie auf den Befehl **Benutzerdefinierte Symbolleisten-Schaltflächen** klicken, wird das gleichnamige Dialogfeld (*Abbildung unten*) angezeigt, in dem Sie eine benutzerdefinierte Authentic-Symbolleiste erstellen können. Nachdem eine Authentic-Symbolleiste mit einem SPS gespeichert wurde, wird die Symbolleiste in der Authentic-Ansicht der Enterprise und der Professional Edition von Altova-Produkten angezeigt, wenn eine XML-Datei, die mit diesem SPS verknüpft ist, in der Authentic-Ansicht bearbeitet wird.

**Anmerkung:** Altova-Produkte mit einer Authentic-Ansicht sind: die Enterprise und Professional Edition von XMLSpy und StyleVision und die Enterprise Edition von Authentic Desktop und Authentic Browser.



### Hinzufügen einer Schaltfläche

Um eine Schaltfläche oder eine Trennlinie zur Authentic-Symbolleiste hinzuzufügen, klicken Sie rechts oben im Dialogfeld auf die Schaltfläche **Hinzufügen**. Daraufhin erscheint ein Menü, in dem Sie den gewünschten Befehl auswählen können: (i) ein Textstatus-Symbol, (ii) eine benutzerdefinierte Schaltfläche zur Ausführung eines Makros oder (iii) eine Trennlinie in der Symbolleiste, die Schaltflächengruppen voneinander trennt. [Textstatus-Symbole](#) und [benutzerdefinierte Schaltflächen](#) werden weiter unten näher beschrieben.

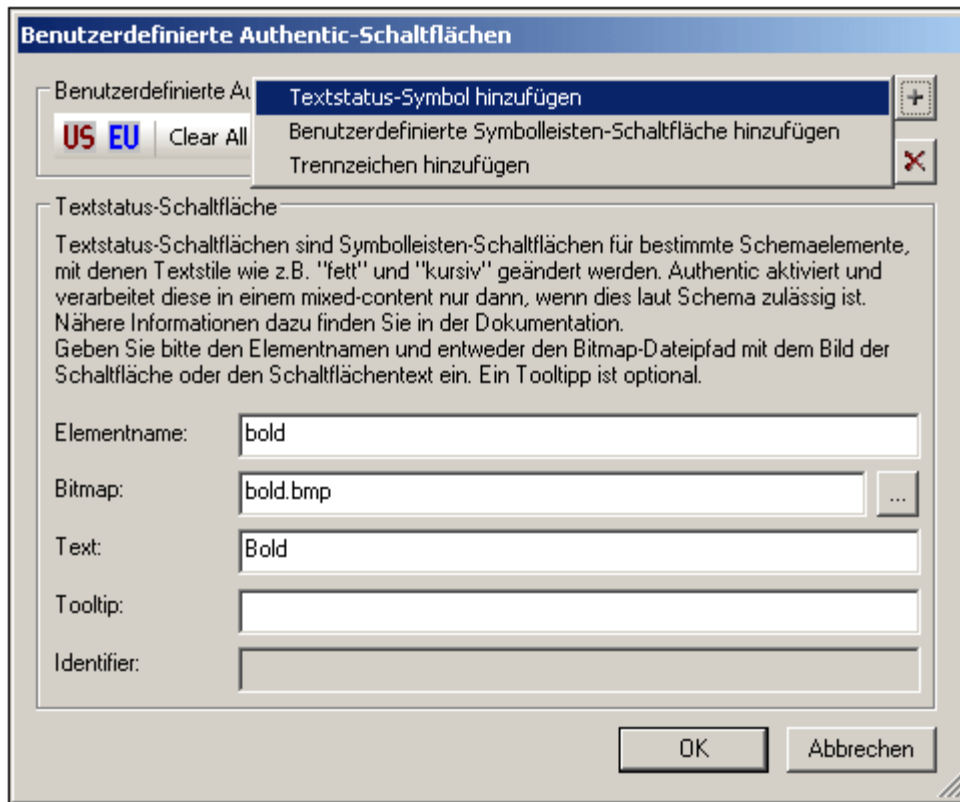
### Verschieben und Löschen einer Schaltfläche

Um eine Schaltfläche oder eine Trennlinie an eine andere Stelle in der Symbolleiste zu verschieben, wählen Sie sie aus und ziehen Sie sie an die neue Stelle. Um eine Schaltfläche oder eine Trennlinie zu löschen, wählen Sie sie aus und klicken Sie rechts oben im Dialogfeld auf die Schaltfläche **Löschen**.

### Textstatus-Symbole

Mit einem Textstatus-Symbol wird ein Symbol für ein globales Element definiert. Wenn Sie in der Authentic-Ansicht Text im Authentic-Dokument auswählen und auf ein Textstatus-Symbol klicken, so wird das Element, das mit diesem Symbol definiert wird, rund um den ausgewählten Text eingefügt. Textstatus-Symbole sind für Elemente gedacht, die Inline-Formatierungen wie z.B. fett und kursiv definieren.

Um ein Textstatus-Symbol zur Authentic-Symboleiste hinzuzufügen, klicken Sie rechts oben im Dialogfeld "Benutzerdefinierte Authentic-Schaltflächen" (siehe Abbildung unten) auf die Schaltfläche **Hinzufügen** und wählen Sie den Befehl **Textstatus-Symbol hinzufügen**. Geben Sie den Namen des XML-Elements ein, für das diese Schaltfläche erstellt wird, navigieren Sie anschließend zu einer Bitmap-Datei für die Schaltfläche oder geben Sie einen Text für die Schaltfläche ein. Als Hilfe für den Benutzer der Authentic-Ansicht können Sie optional einen Tooltip eingeben, der angezeigt wird, wenn der Benutzer die Maus über die benutzerdefinierte Schaltfläche platziert. Klicken Sie auf **OK**, um die Schaltfläche zur Authentic-Symboleiste hinzuzufügen.



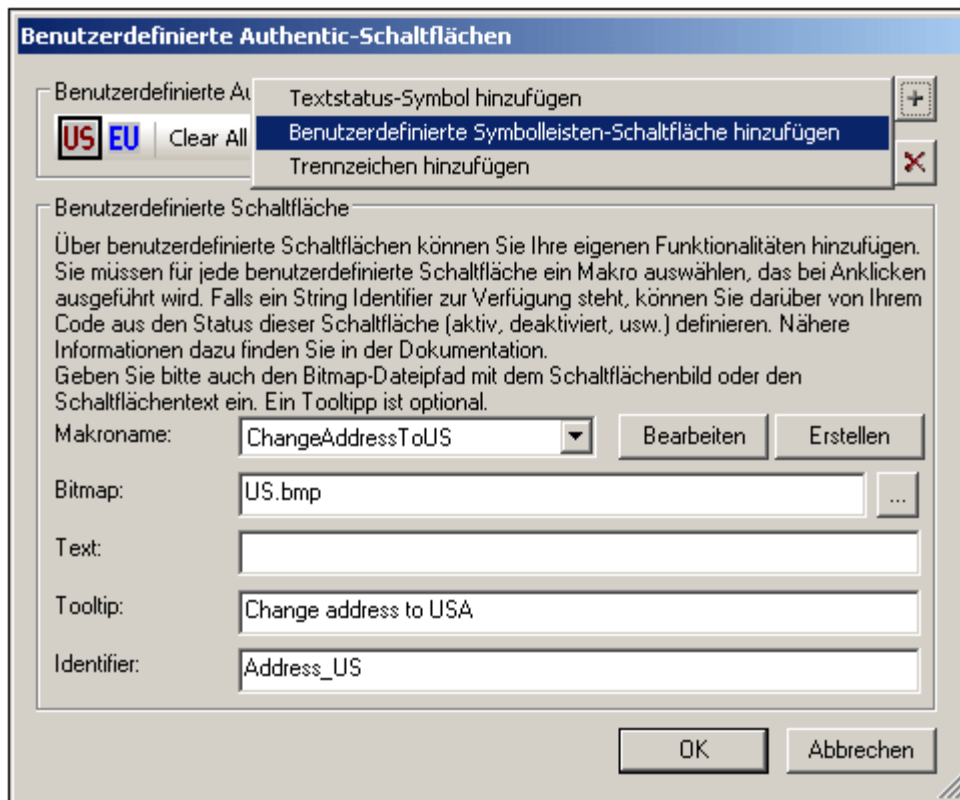
In der Abbildung oben sehen Sie, dass das Textstatus-Symbol für das Element `bold` ausgewählt ist. Für dieses Textstatus-Symbol wird ein Bild namens `bold.bmp` verwendet. Zusätzlich dazu kann Text für die Schaltfläche eingegeben werden. Wenn das Bild für das in der Abbildung oben definierte Textstatus-Symbol nicht gefunden wird, so wird als Schaltflächentext der Text `Bold` verwendet.

### Benutzerdefinierte Schaltflächen

Wenn ein Benutzer in der Authentic-Ansicht in der Authentic-Symboleiste auf eine

benutzerdefinierte Schaltfläche klickt, wird ein Makro ausgeführt. Sie können im SPS-Design eine benutzerdefinierte Schaltfläche erstellen und definieren, welches Makro damit aufgerufen werden soll.

Um eine benutzerdefinierte Schaltfläche zur Authentic-Symbolleiste hinzuzufügen, klicken Sie rechts oben im Dialogfeld "Benutzerdefinierte Authentic-Schaltflächen" (siehe Abbildung unten) auf die Schaltfläche **Hinzufügen** und wählen Sie den Befehl **Benutzerdefinierte Symbolleisten-Schaltfläche hinzufügen**.



Benutzerdefinierte Schaltflächen erhalten die folgenden Parameter zugewiesen:

- Den Speicherpfad eines Bilds für die Schaltfläche (im Feld "Bitmap") oder Schaltflächentext (im Feld "Text").
- Wählen Sie in der Auswahlliste "Makroname" ein Makro aus der Dropdown-Liste aus. Hier sind die Makros aufgelistet, die mit dem SPS gespeichert wurden. Wenn Sie auf die Schaltfläche **Neu** klicken, wird der [Skript-Editor](#) von StyleVision in einem eigenen Fenster geöffnet. Hier können Sie ein Makro schnell und einfach erstellen und es mit dem SPS speichern. Wenn Sie auf die Schaltfläche **Bearbeiten** klicken, wird das ausgewählte Makro für die Bearbeitung im [Skript-Editor](#) geöffnet.
- Als Hilfe für den Benutzer der Authentic-Ansicht können Sie optional einen Tooltip eingeben, der angezeigt wird, wenn der Benutzer die Maus über die benutzerdefinierte Schaltfläche platziert.
- Geben Sie im Feld "Identifier" einen Textstring ein, der als Identifier für die benutzerdefinierte Schaltfläche verwendet wird. Dieser Identifier kann im Skript-Code verwendet werden.

In der Abbildung oben sehen Sie die benutzerdefinierte Schaltfläche `ChangeAddressToUS`. Für diese benutzerdefinierte Schaltfläche wird ein Bild mit dem Namen `US.bmp` verwendet. Zusätzlich dazu kann Text für die Schaltfläche eingegeben werden. Es wurde auch ein Tooltip



eingegeben und die benutzerdefinierte Schaltfläche hat den Identifier `Address_US`. Dieses Beispiel stammt aus der Datei `ToolbarButtons.sps`, das sich im Projektfenster im Ordner `Authentic\Scripting` des Projekts `Examples` befindet.

**Makro-Referenzen überprüfen**

Mit diesem Befehl wird überprüft, ob Referenzen von Symbolleisten-Schaltflächen und Skripts zu Makros korrekt sind. Wenn fehlerhafte Referenzen gefunden werden, wird eine Fehlermeldung angezeigt.

## Datumswahl automatisch hinzufügen



Dies ist ein Ein/Aus-Befehl zum Ein- und Ausschalten der automatischen Datumswahlfunktion. Wenn der Befehl aktiviert ist, wird bei jedem Element vom Datentyp `xs:date` oder `xs:dateTime`, das als Inhalt oder Eingabefeld erstellt wird, die Datumswahl innerhalb der Element-Tags und hinter dem Platzhalter `Inhalt` oder dem Eingabefeld automatisch eingefügt.

## DB-Steuerelemente automatisch hinzufügen

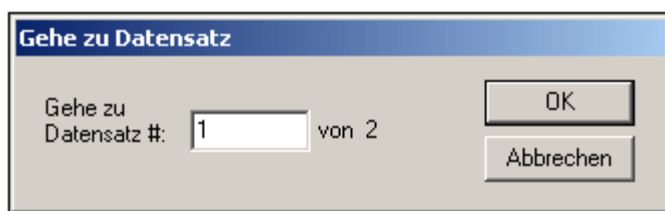


Dies ist ein Ein/Aus-Befehl zum Aktivieren und Deaktivieren der Funktion zum automatischen Hinzufügen von DB-Steuerelementen

Wenn diese Funktion aktiviert ist, wird jedes Mal, wenn ein DB-Tabellenelement in das Design gezogen wird, direkt vor dem Child-Element `Row` dieses DB-Tabellenelements die Leiste mit den DB-Steuerelementen (*siehe unten*) eingefügt.



Mit Hilfe der DB-Steuerelementleiste kann der Benutzer der Authentic-Ansicht in der Authentic-Ansicht zwischen den Zeilen der DB-Tabelle navigieren. Mit Hilfe der ersten Schaltfläche (ganz links) können Sie zum ersten Datensatz navigieren; mit der zweiten Schaltfläche gelangen Sie vom vorhergehenden Datensatz; die dritte Schaltfläche ist die "Gehe zu" Schaltfläche, über die das Dialogfeld zur Eingabe der gewünschten Datensatznummer (*Abbildung unten*) aufgerufen wird; über die vierte Schaltfläche gelangen Sie zum nächsten Datensatz; und über die fünfte Schaltfläche gelangen Sie zum letzten Datensatz.



Wenn die Funktion zum automatischen Hinzufügen von DB-Steuerelementen deaktiviert ist, wird die DB-Steuerelementleiste **nicht** eingefügt, wenn eine DB-Tabelle in das Design-Dokument gezogen wird.

**Anmerkung:** Sie können Navigationsschaltflächen auch manuell einfügen. Platzieren Sie dazu den Cursor an eine beliebige Stelle zwischen dem Start- und End-Tag der DB-Tabelle und wählen Sie die gewünschte Option aus dem Untermenü **Einfügen | DB-Steuerelemente**. Beachten Sie: Die DB-Steuerelementleiste kann in diesem Untermenü entweder mit vier Navigationsschaltflächen oder mit vier Navigationsschaltflächen plus der Schaltfläche zum Aufrufen des Dialogfelds "Gehe zu" aufgerufen werden.

### Authentic-Ansicht neu laden, XML validieren

Mit dem Befehl **Neu laden** wird die XML-Authentic-Datei neu geladen. Dies kann nützlich sein, wenn die Datei außerhalb von StyleVision bearbeitet wurde, v.a. wenn ein anderer Benutzer die Datei auf einem anderen Rechner geändert hat.

Mit dem Befehl **XML validieren (F8)**  wird die Gültigkeit der XML-Datei gegenüber dem damit verknüpften Schema überprüft. Außerdem werden auch alle zusätzlichen Validierungskriterien, die Sie für einzelne Nodes (**Authentic | Node-Einstellungen**) definiert haben, überprüft. Das Ergebnis der Validierungsüberprüfung wird in einem Popup-Meldungsfeld angezeigt.

## Neue Zeile mit XML-Daten für Bearbeitung auswählen

Mit diesem Befehl können Sie eine neue Zeile in der entsprechenden Tabelle einer XML-Datenbank wie z.B. IBM DB2 auswählen. Diese Zeile wird nun in der Authentic-Ansicht angezeigt, kann hier bearbeitet und wieder in der Datenbank gespeichert werden.

Bei Verwendung einer XML-Datenbank als XML-Datenquelle handelt es sich bei den in der Authentic-Ansicht angezeigten XML-Daten um das XML-Dokument, das in einer der Zellen der XML-Datenspalte enthalten ist. Mit dem Befehl **Neue Zeile mit XML-Daten für Bearbeitung auswählen** können Sie ein XML-Dokument aus einer anderen Zelle (oder Zeile) dieser XML-Spalte auswählen. Bei Auswahl des Befehls **Neue Zeile...auswählen** wird das Dialogfeld "XML-Feld auswählen" aufgerufen (*Abbildung unten*), in dem die Tabelle, die die XML-Spalte enthält angezeigt wird.

**XML-Feld auswählen**

Bitte wählen sie das Quellfeld aus der Tabelle 'CUSTOMER' die XML-Daten enthält, aus:

Filter (Where-Klausel):

| CID  | INFO                                            | HISTORY |
|------|-------------------------------------------------|---------|
| 1003 | <?xml version="1.0" encoding="UTF-8" ?><cust... | [NULL]  |
| 1004 | <?xml version="1.0" encoding="UTF-8" ?><cust... | [NULL]  |
| 1005 | <?xml version="1.0" encoding="UTF-8" ?><cust... | [NULL]  |

Bitte beachten Sie: Nur die ersten 100 Zeilen werden angezeigt!

Sie können für diese Tabelle einen Filter eingeben. Dabei sollte es sich um eine SQL WHERE-Klausel handeln (nur die Bedingung ohne das WHERE-Schlüsselwort wie z.B.: CID>1002). Klicken Sie auf **Aktualisieren**, um das Dialogfeld zu aktualisieren. In der Abbildung oben sehen Sie das Ergebnis einer gefilterten Ansicht. Wählen Sie als nächstes die Zelle, die das gewünschte XML-Dokument enthält aus, und klicken Sie auf **OK**. Das XML-Dokument in der ausgewählten Zelle (Zeile) wird in die Authentic-Ansicht geladen.

## XML Entities definieren



- Der Befehl **XML Entities definieren** steht nur in der Authentic-Ansicht zur Verfügung. Mit diesem Befehl können Sie in der Authentic-Ansicht Entities definieren, die Sie zu Ihrem **XML-Dokument** hinzufügen möchten. Nachdem eine Entity definiert wurde, können Sie diese in das XML-Dokument einfügen, indem Sie mit der rechten Maustaste an die Stelle klicken, an der die Entity eingefügt werden soll. Wählen Sie anschließend im Kontextmenü den Befehl **Entity einfügen** und anschließend den Namen der gewünschten Entity.

Bei Entities, die Sie mit diesem Befehl definieren, kann es sich um die folgenden drei Arten von Entities handeln:

- Interne geparte Entity. Der Wert der Entity ist ein Textstring, der gewöhnlich häufig im Dokument vorkommt. Durch Verwendung einer Entity stellen Sie sicher, dass alle Instanzen den hier definierten Wert enthalten.
- Externe geparte Entity. Dies ist eine externe XML-Datei, durch die die einzelnen Instanzen der Entity ersetzt werden. Der Wert der Entity ist die URI der externen XML-Datei.
- Externe ungeparte Entity. Dies ist eine externe Ressource, die bei der Verarbeitung der Entity aufgerufen wird. Der Wert der Entity ist die URI der externen Ressource.

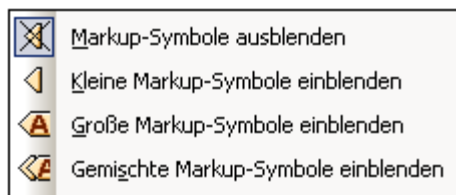
Wenn Sie auf den Befehl klicken, wird das Dialogfeld "Entities definieren" (siehe unten) aufgerufen.

| Name     | Type     | PUBLIC | Value/Path               | NDATA |
|----------|----------|--------|--------------------------|-------|
| nano_dc  | Internal |        | Nanonull, Inc            | ...   |
| Nano_eu  | Internal |        | Nanonull Europe, AG      | ...   |
| Nano_ma  | Internal |        | Nanonull Partners, Inc   | ...   |
| website  | Internal |        | http://www.nanonull.com/ | ...   |
| branches | SYSTEM   |        | branches.xml             | ...   |
| logo     | SYSTEM   |        | nanonull.gif             | ...   |

Eine Beschreibung zur Verwendung dieses Dialogfelds finden Sie in der Dokumentation zur Authentic-Ansicht unter [Entities definieren](#).

## Markup-Code anzeigen

Der Befehl **Markup-Code anzeigen** hat ein Untermenü mit Optionen, mit denen Sie auswählen können, wie Markup-Code im XML-Authentic-Dokument angezeigt werden soll. Mit den vier **Markup-Befehlen** (*Abbildung unten*) können Sie zwischen den Modi "Markup-Symbole ausblenden" und den verschiedenen Arten der Anzeige von Markup-Symbolen auswählen.

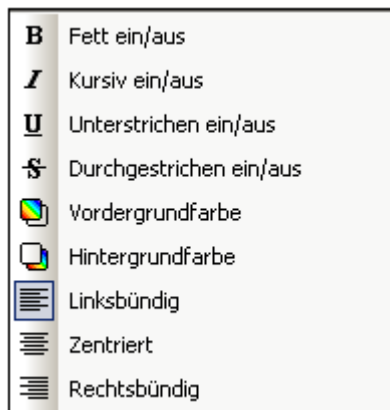


Die Optionen für Markup-Symbole beziehen sich auf die Art, wie verschiedene Node-Tags in der Authentic-Ansicht angezeigt werden. Es kann immer nur eine Option ausgewählt werden. Bei Auswahl der Option "Markup-Symbole ausblenden" werden ausgewählte Node-Tags nicht angezeigt. Bei Auswahl von "Kleine Markup-Symbole" werden nur öffnende und schließende Tags ohne Node-Namen angezeigt. Bei Auswahl von "Große Markup-Symbole" werden die öffnenden und schließenden Node-Tags mit ihren entsprechenden Node-Namen angezeigt. Die Option "Gemischte Markup-Symbole" bezieht sich auf die Markup-Symbole, die in den [Authentic Node-Eigenschaften](#) einzelner Nodes definiert wurden. Da die Standard-Markup-Eigenschaft für einzelne Nodes "Markup-Symbole ausblenden" ist, werden keine Markup-Symbole (weder große noch kleine) angezeigt, es sei denn Sie haben kleine oder große Markup-Symbole (als [Node-Eigenschaft](#)) für einige Nodes definiert.



## RichEdit

Wenn Sie den Mauszeiger über einen RichEdit-Befehl platzieren, wird ein Untermenü mit den RichEdit-Markup-Befehlen (*Abbildung unten*) angezeigt. Die Menübefehle in diesem Untermenü sind nur in der Authentic-Ansicht und nur, wenn der Cursor sich innerhalb eines als RichEdit-Komponente erstellten Elements befindet, aktiv.



Wenn Sie auf einen RichEdit-Befehl klicken, werden die Textstileigenschaften des RichEdit-Menüs auf den ausgewählten Text angewendet. Der Benutzer der Authentic-Ansicht kann die Schriftart, Schriftbreite, den Schriftstil, die Schriftverzierung, die Farbe, Hintergrundfarbe und Ausrichtung des ausgewählten Texts definieren.

Nähere Informationen zur Verwendung der RichEdit-Funktion finden Sie unter [Flexibles Zuweisen von Textstilen in Authentic](#).

### (Dynamische Tabelle) Zeilenbefehle

Die **Zeilenbefehle (für dynamische Tabellen)** sind in der Authentic-Ansicht aktiviert, wenn der Cursor in eine Zeile einer dynamischen Tabelle platziert wird. Mit Hilfe dieser Befehle können Sie die Zeilen einer dynamischen Tabelle bearbeiten. Sie können Zeilen anhängen, einfügen, kopieren und löschen und Sie können die ausgewählte Zeile im Verhältnis zu den anderen Zeilen einer Tabelle nach oben oder unten verschieben. Da es sich bei einer Zeile in einer dynamischen Tabelle um eine feste Datenstruktur handelt, arbeitet der Benutzer der <%AUTH-VIEW%-Ansicht mit Einheiten einer Datenstruktur im Kontext der durch die dynamische Tabelle dargestellten Datenstruktur.



Eine Zeile wird ausgewählt, indem Sie den Cursor hineinplatzieren. Sie können nun vor der ausgewählten Zeile eine leere Zeile einfügen oder hinter dieser eine Zeile anhängen. Eine Zeile kann kopiert werden. Dabei wird eine Kopie der Zeile mitsamt ihrem Inhalt nach der ausgewählten Zeile eingefügt. Eine Zeile kann auch im Verhältnis zu benachbarten Zeilen nach oben oder nach unten verschoben werden.

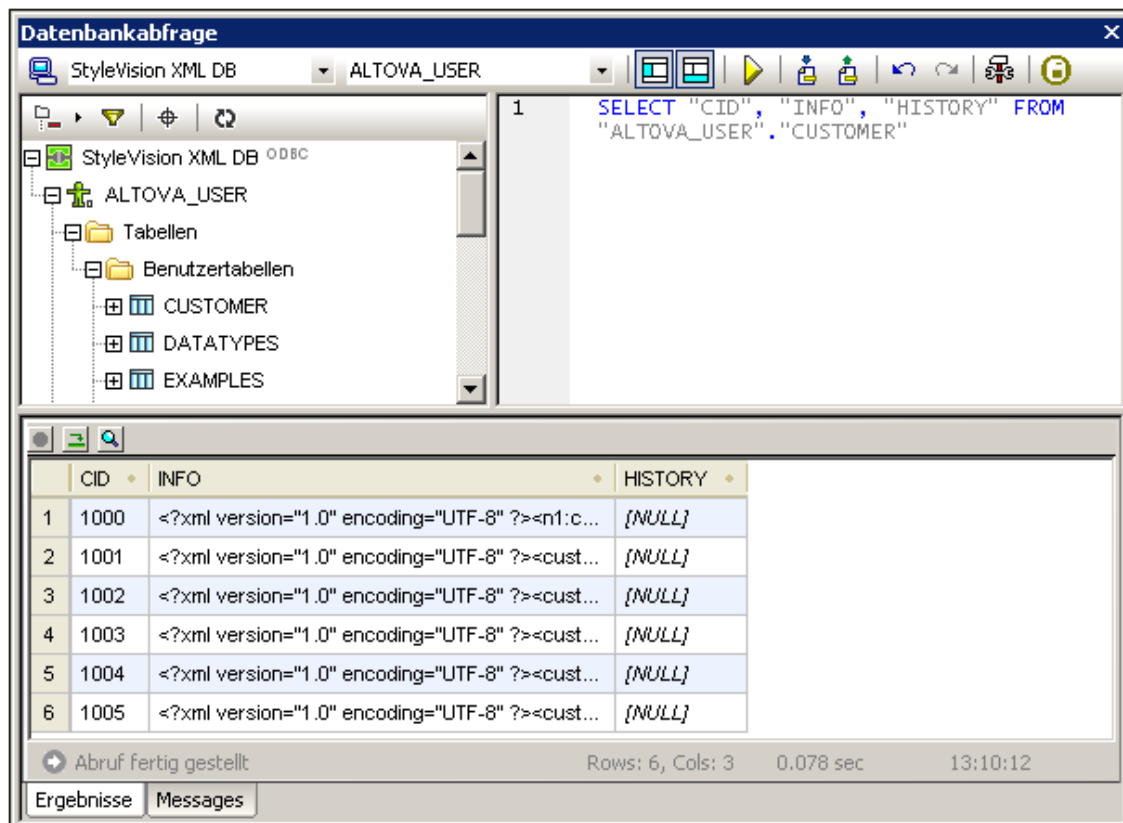
## 19.11 Datenbank

Das Menü **Datenbank** enthält Befehle zum Abfragen der verbundenen Datenbank und zum Bearbeiten und Löschen der auf die angebundene Datenbank angewendeten Filter.

- [Datenbank abfragen](#) startet den Vorgang zum [Herstellen einer Datenbankverbindung](#) und öffnet das Fenster "Datenbankabfrage".
- [DB-Filter bearbeiten](#), [DB-Filter löschen](#): Dient zum Aufrufen des Dialogfelds "DB-Filter bearbeiten" bzw. zum Löschen von Datenbankfiltern.

## Datenbank abfragen

Mit dem Befehl **Datenbank abfragen**  wird das Fenster "[Datenbankabfrage](#)" (*Abbildung unten*) geöffnet. Über dieses Dialogfeld können Sie eine Verbindung zu einer Datenbank herstellen und diese abfragen. Eine Beschreibung zum Verwenden des Fensters "Datenbankabfrage" finden Sie im Abschnitt [Datenbank abfragen](#).



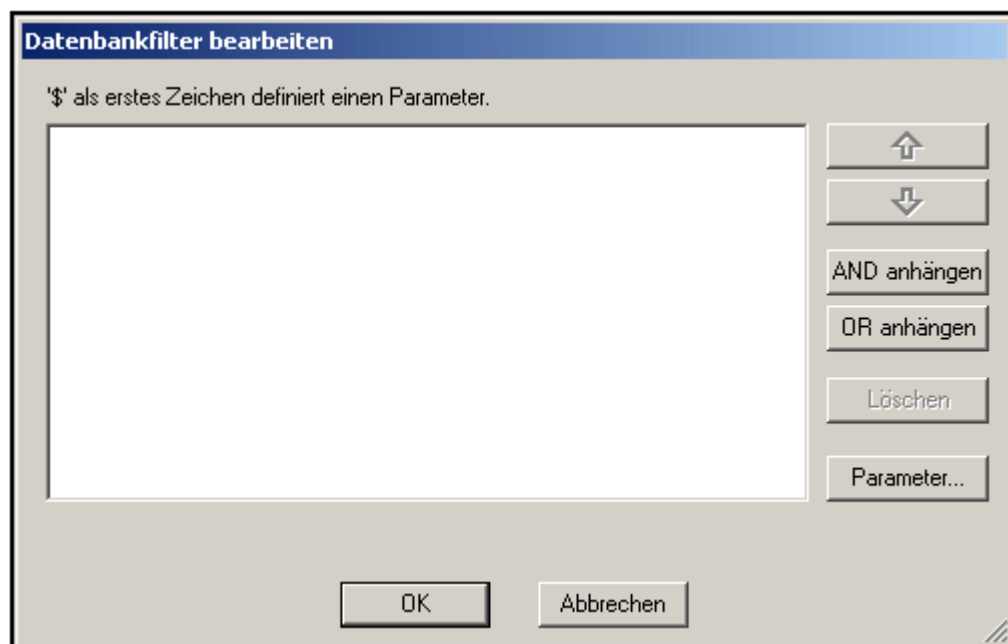
Das Datenbankabfragefenster wird durch Klicken auf den Befehl **Ansicht | Datenbank abfragen** ein und ausgeblendet.

## DB-Filter bearbeiten, DB-Filter löschen

Mit Hilfe des Befehls **DB-Filter bearbeiten**  können Sie einen Filter für eine Datenbanktabelle (eine DB-Filter) erstellen und bearbeiten. Ein DB-Filter legt fest, welche Daten aus der ausgewählten Datenbanktabelle importiert und angezeigt werden sollen. Ein DB-Filter besteht aus einem oder mehreren Filterkriterien. Wenn Sie Kriterien angeben, verwenden Sie einen Ausdruck, welcher eine Kombination aus Operatoren (= oder >) und Werten (Text oder Zahlen) darstellt. Zusätzlich dazu können Sie mit Hilfe der logischen Operatoren **AND** oder **OR** Kriterien miteinander verbinden.

Um einen DB-Filter zu erstellen oder zu bearbeiten, gehen Sie folgendermaßen vor:

1. Wählen Sie das oberste Datentablenelement aus, für das ein DB-Filter erstellt oder bearbeitet werden soll. Klicken Sie dazu entweder auf den Element-Tag in der Design-Ansicht oder auf dem Elementnamen in der Schemastruktur.
2. Wählen Sie den Befehl **Bearbeiten | DB-Filter bearbeiten** oder klicken Sie auf die entsprechende Symbolleisten-Schaltfläche. Daraufhin wird das Dialogfeld "DB-Filter bearbeiten" angezeigt.



3. Zum Hinzufügen von Kriterien verwenden Sie die Schaltflächen **AND anhängen** und **OR anhängen**. Um ein Kriterium nach oben oder unten zu verschieben, verwenden Sie die Pfeilschaltflächen. Um ein Kriterium zu löschen, verwenden Sie die Schaltfläche "Löschen".
4. Definieren Sie das Kriterium für den DB-Filter. Jedes Kriterium besteht aus drei Teilen: **Feldname** + **Operator** + **Wert**. Die Optionen für Feldnamen und Operatoren stehen in Auswahllisten zur Verfügung. Der Wert für den Ausdruck muss über die Tastatur eingegeben werden und kann ein Parameter sein (wird durch ein vorangestelltes \$ Zeichen gekennzeichnet).

### Befehl "DB-Filter löschen"

Mit dem Befehl **DB-Filter löschen**  wird der Filter nach einer Bestätigungsaufforderung gelöscht.

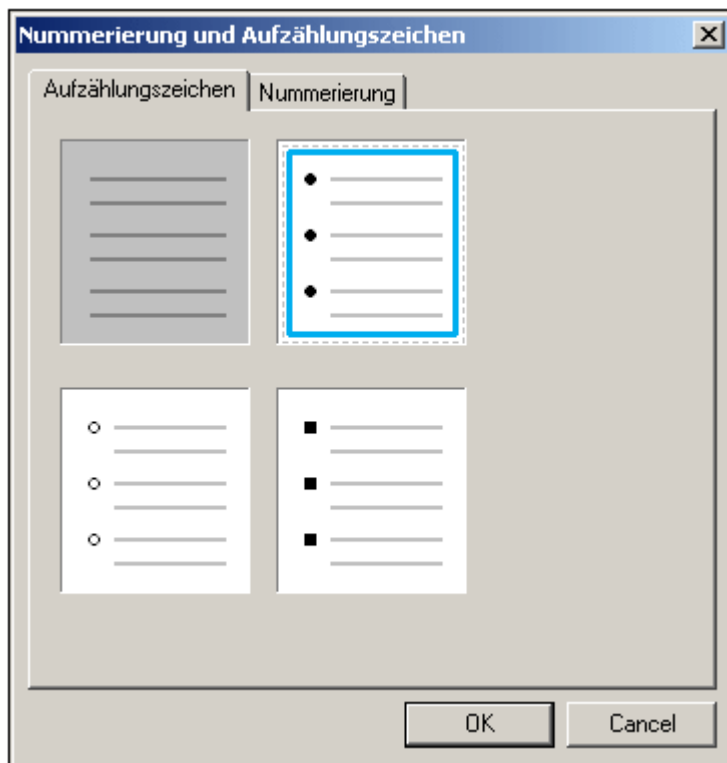
## 19.12 Menü "Eigenschaften"

Das Menü **Eigenschaften** enthält Befehle zum Einfügen von Listen und Definieren von Datentyp-Formaten für die Funktion [Werteformatierung](#). Die Beschreibung der Befehle ist in die folgenden Unterabschnitte gegliedert:

- der Befehl [Nummerierung und Aufzählungszeichen](#) zum Einfügen von Listen.
- der Befehl [Vordefinierte Formatierungsstrings](#) zum Definieren von numerischen Datentypformaten für ein bestimmtes SPS.

## 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.



## Vordefinierte Wertformatierungsstrings

Jedem Platzhalter ( *Inhalt* ), jedem Eingabefeld, jeder automatischen Berechnung vom Datentyp *numeric*, *date*, *time*, *dateTime* oder *duration* kann über das Dialogfeld [Wertformatierung](#) ein benutzerdefiniertes Format zugewiesen werden. Im Dialogfeld "Wertformatierung" können Sie ein Format entweder direkt erstellen oder es aus einer Dropdown-Liste vordefinierter Formate auswählen. Diese Liste besteht aus zwei Arten von vordefinierten Formaten:

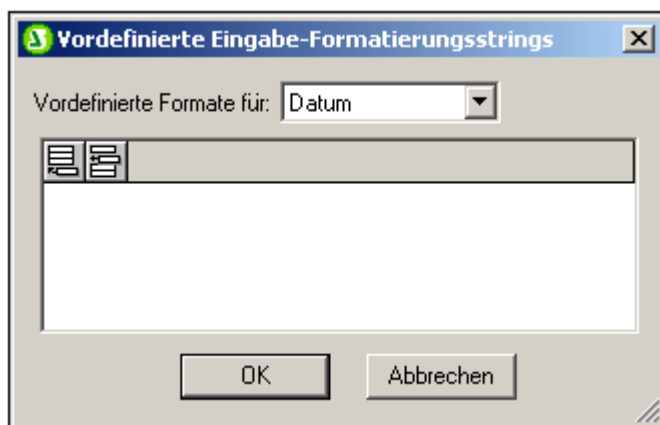
In der Dropdown-Liste stehen zwei Arten von vordefinierten Formaten zur Verfügung:

- 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 Wertformate 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.

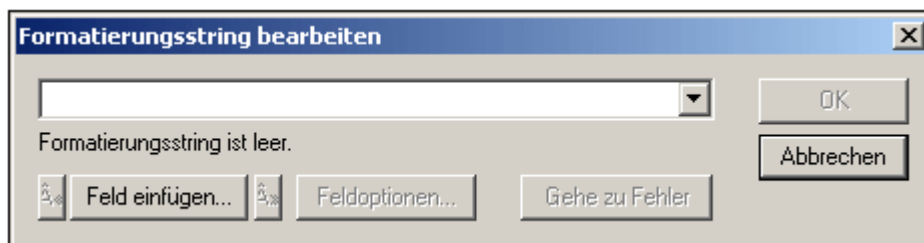
### Erstellen eines vordefinierten Wertformatierungsstrings

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:

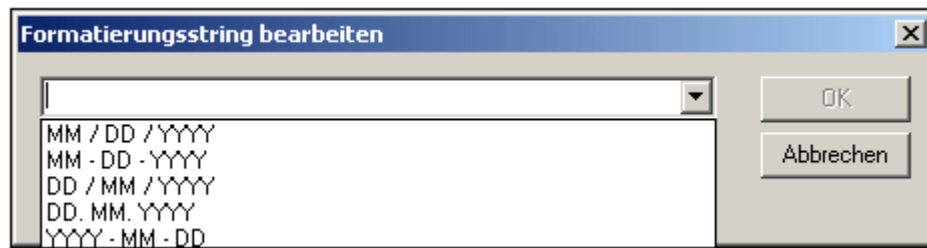


2. Wählen Sie aus der Dropdown-Auswahlliste einen Datentyp aus und klicken Sie anschließend je nach Bedarf auf das Symbol **"Anhängen"** oder **"Einfügen"**. Daraufhin wird das Dialogfeld "Formatierungsstring bearbeiten" angezeigt:



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](#) 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.

**Anmerkung:**

- 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".

## 19.13 Menü "Extras"

Das Menü **Extras** enthält den Befehl zur Rechtschreibprüfung und Befehle zum Anpassen von StyleVision.

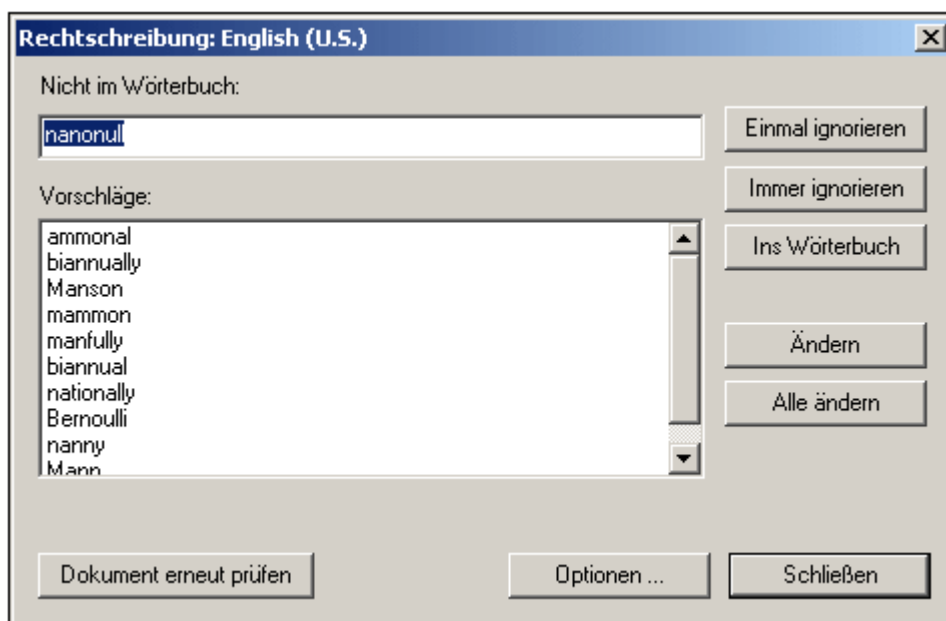
Die Beschreibung zu den Befehlen des Menüs "Extras" ist in die folgenden Unterabschnitte gegliedert:

- [Rechtschreibung](#)
- [Rechtschreibung - Optionen](#)
- [Globale Ressourcen](#)
- [Aktive Konfiguration](#)
- [Anpassen](#)
- [Optionen](#)

## Rechtschreibung

Mit dem Befehl **Rechtschreibung (Umschalt+F7)** wird - je nachdem, welches Dokument aktiv ist - eine Rechtschreibprüfung am SPS (in der Design-Ansicht) oder am Dokument in der Authentic-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 Lizenz einschränkungen wie z.B. im Rahmen von [GPL](#) Lizenzen weitergegeben. Viele dieser Wörterbücher stehen als Teil eines separaten Installationsprogramms unter <http://www.altova.com/dictionaries> zur Verfügung. 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 Words 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 Words ignorieren (**Einmal ignorieren**) oder *alle Instanzen* dieses unbekannten Words 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.

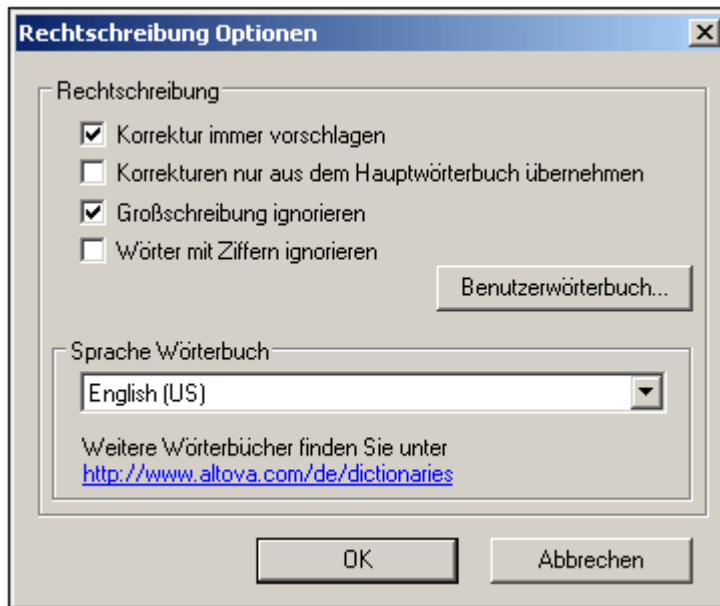
Nachdem alle nicht im Wörterbuch gefundenen Wörter nacheinander angezeigt wurden und für jedes Wort eine Aktion durchgeführt wurde, wird die folgende Meldung angezeigt: "Rechtschreibprüfung abgeschlossen". Sie können das Dokument anschließend entweder erneut von Anfang an überprüfen (**Dokument erneut prüfen**) oder das Dialogfeld schließen (

**Schließen).**

Über die Schaltfläche **Optionen** wird das Dialogfeld [Rechtschreibung Optionen](#) geöffnet. Hier können Sie Optionen für die Rechtschreibprüfung definieren.

## Rechtschreibung - Optionen

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 keine 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.

### *Wörter in Großschreibung ignorieren:*

Bei Auswahl dieser Option werden alle Wörter in Großschreibung ignoriert.

### *Wörter mit Ziffern ignorieren:*

Bei Auswahl dieser Option werden alle Wörter, die Ziffern enthalten, ignoriert.

### *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](http://www.altova.com/de/dictionaries) 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:

*Auf Windows 7:* C:\ProgramData\Altova\SpellChecker\Lexicons

*Auf Windows XP:* C:\Documents and Settings\All Users\Application Data\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)` ) auf einem Windows 7-Rechner folgendermaßen gespeichert:

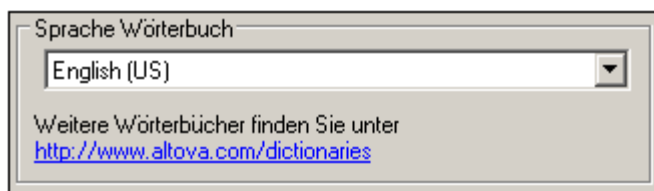
```
C:\ProgramData\Altova\SpellChecker\Lexicons\English (British)\en_GB.aff
C:\ProgramData\Altova\SpellChecker\Lexicons\English (British)\en_GB.dic
C:\ProgramData\Altova\SpellChecker\Lexicons\English (US)\en_US.dic
C:\ProgramData\Altova\SpellChecker\Lexicons\English (US)\en_US.dic
```

Die Sprachwörterbücher werden im Dialogfeld "Rechtschreibung Optionen" 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 <http://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.

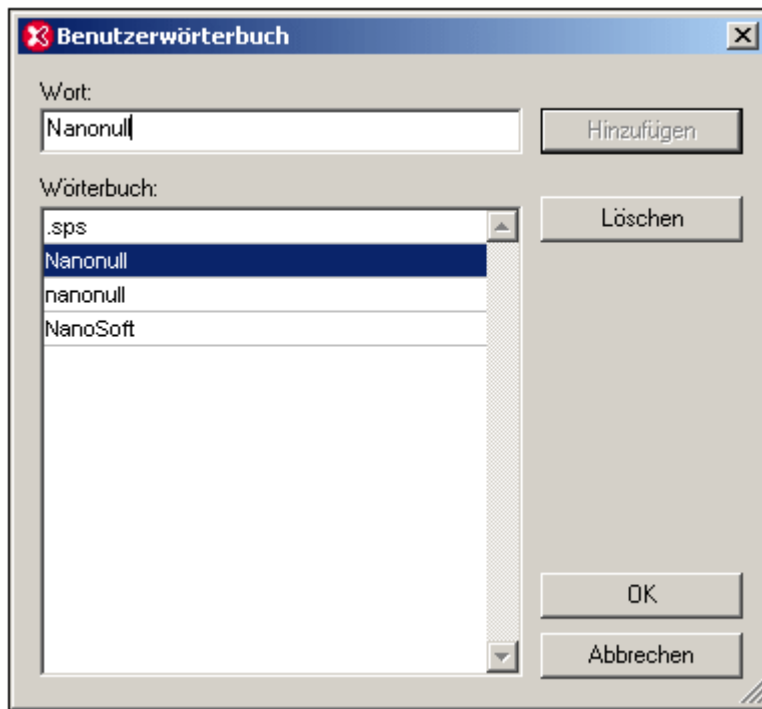


**Anmerkung:** Es bleibt Ihnen überlassen, ob Sie mit den für das jeweilige Wörterbuch geltenden Lizenzbedingungen einverstanden sind und ob das Wörterbuch sich für die Verwendung mit der auf Ihrem Rechner installierten Software eignet.

### 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 "Rechtschreibung Optionen" aufgerufen (*siehe Abbildung im obersten Bereich*

dieses Abschnitts).



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 [Rechtschreibung](#) 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:

*Auf Windows 7:* C: \Users\<user>\Documents\Altova\SpellChecker\Lexicons\user.dic

*Auf Windows XP:* C: \Documents and Settings\<user>\My Documents\Altova\SpellChecker\Lexicons\user.dic

---

**Siehe auch:**

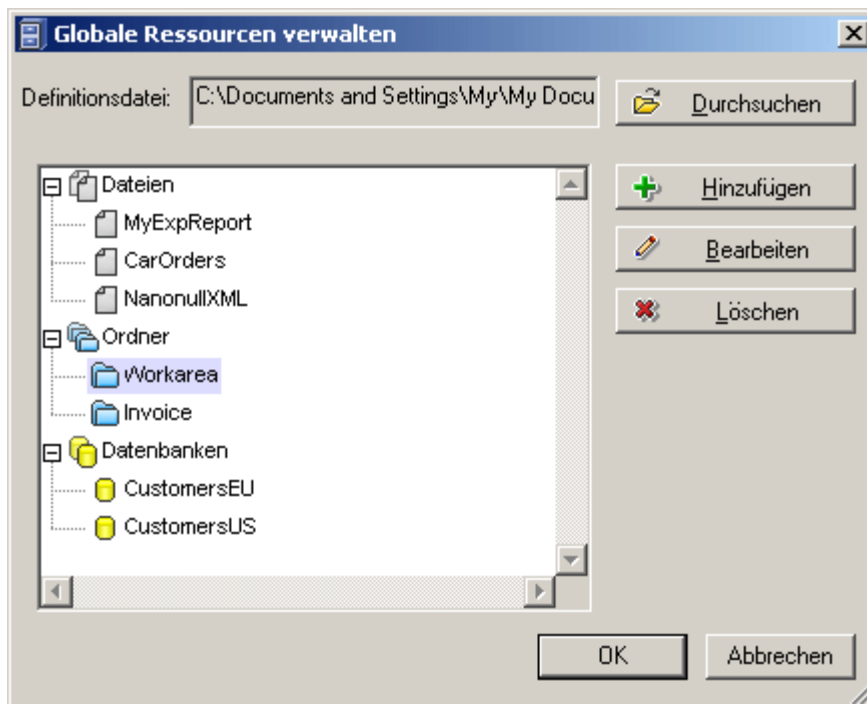
[Rechtschreibung](#): Hier finden Sie eine Beschreibung zum Befehl zum Durchführen einer Rechtschreibprüfung am Design- und am Authentic-Ansicht-Dokument.

---

## Globale Ressourcen

Mit dem Befehl **Globale Ressourcen** wird das Dialogfeld "Globale Ressourcen" aufgerufen (Abbildung unten), in dem Sie Folgendes tun können:

- Angabe der XML-Datei für globale Ressourcen.
- Hinzufügen von globalen Ressourcen (oder Aliassen) vom Typ Datei, Ordner und Datenbank
- Angabe verschiedener Konfigurationen für die einzelnen globalen Ressourcen (Aliasse). Jede Konfiguration wird einer bestimmten Ressource zugeordnet.



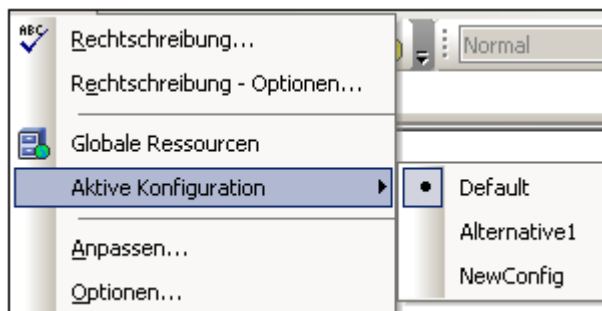
Eine Beschreibung, wie man globale Ressourcen definiert, finden Sie im Abschnitt [Definieren globaler Ressourcen](#).

**Anmerkung:** Das Dialogfeld "Globale Ressourcen" kann auch über die [Symbolleiste "Globale Ressourcen"](#) aufgerufen werden (**Ansicht | Symbolleisten | Globale Ressourcen**).



## Aktive Konfiguration

Wenn Sie die Maus über den Menübefehl **Aktive Konfiguration** platzieren, wird ein Untermenü ausgerollt, das alle in der aktiven [XML-Datei für globale Ressourcen](#) definierten Konfigurationen enthält. (Abbildung unten).



Die gerade aktive Konfiguration ist durch einen Punkt gekennzeichnet. In der Abbildung oben ist die gerade aktive Konfiguration `Default`. Um die aktive Konfiguration zu wechseln, wählen Sie die Konfiguration aus, die aktiv gemacht werden soll.

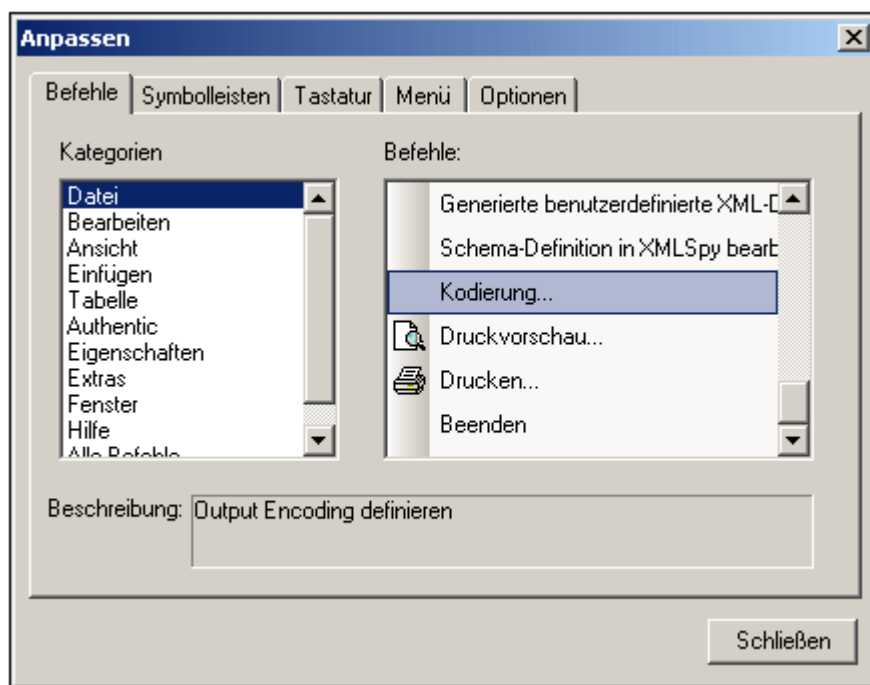
**Anmerkung:** Die aktive Konfiguration kann auch über die [Symbolleiste "Globale Ressourcen"](#) ausgewählt werden (**Ansicht | Symbolleisten | Globale Ressourcen**).

## Anpassen

Mit Hilfe des Befehls "Anpassen" können Sie StyleVision Ihren persönlichen Bedürfnissen entsprechend anpassen.

### 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".

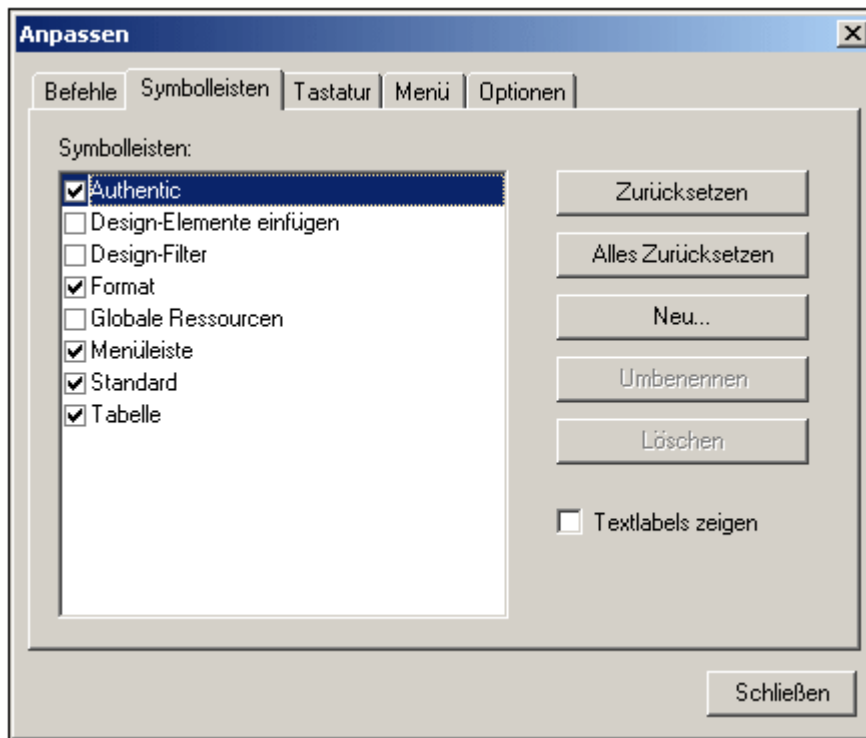
### Anmerkung:

- 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.

### 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.



Auf der Benutzeroberfläche von StyleVision werden eine fixe Menüleiste sowie mehrere optionale Symbolleisten (Authentic, 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.

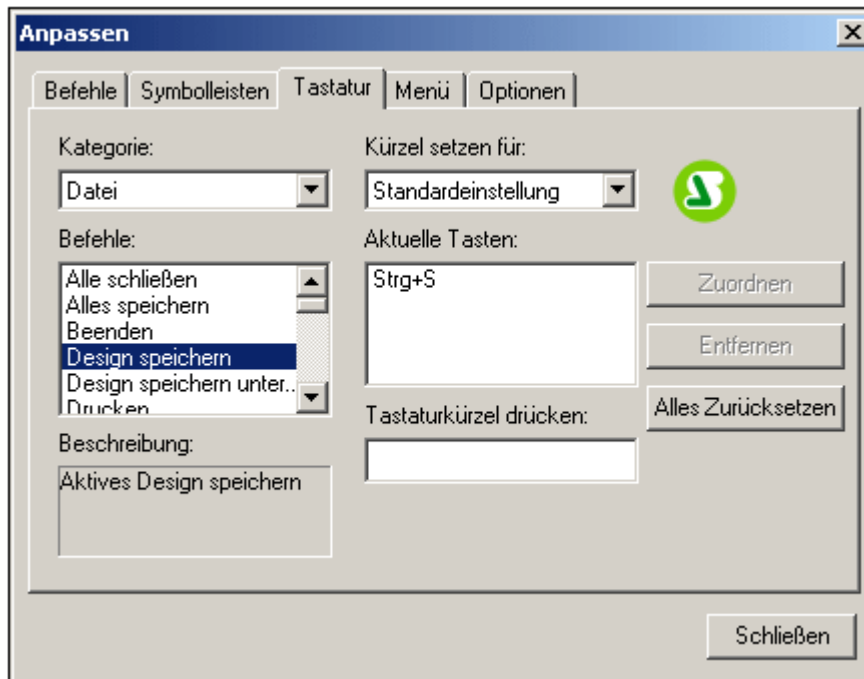
### Menüleiste

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.)

### 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

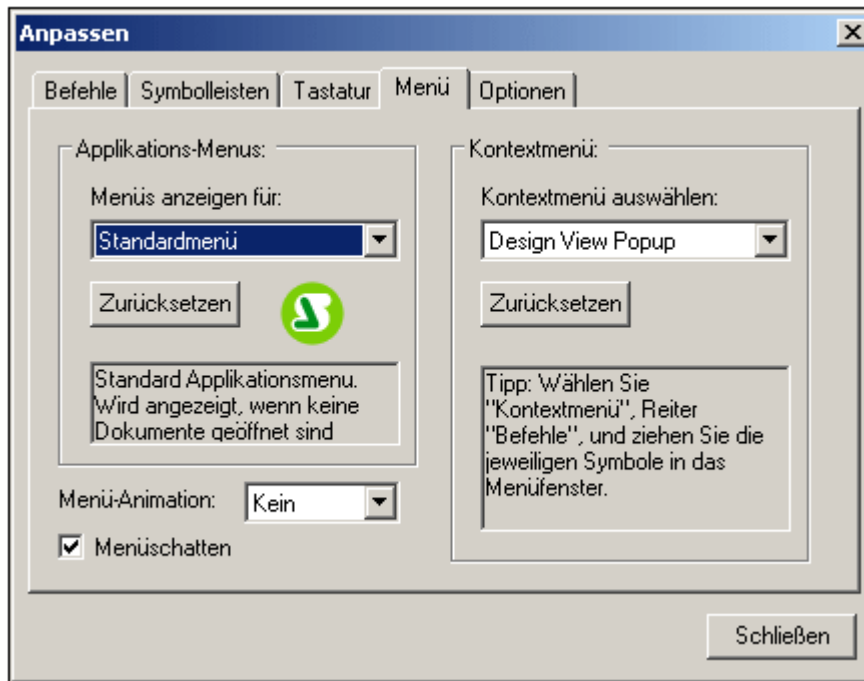
1. Klicken Sie auf die Schaltfläche **Alles zurücksetzen**, um den Originalzustand wieder herzustellen. Daraufhin erscheint ein Dialogfeld, in dem Sie gefragt werden, ob Sie wirklich alle Tastenzuweisungen zurücksetzen möchten.
2. Klicken Sie auf "Ja", um alle Tastenzuweisungen zurückzusetzen.

### Kürzel setzen für

Derzeit steht keine Funktion zur Verfügung.

### Menü

Über das Register **Menü** können Sie die Hauptmenüleiste sowie die Kontextmenüs anpassen.



### So passen Sie ein Menü an

1. Wählen Sie die gewünschte Menüleiste aus (aktuell das Standardmenü).
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. Klicken Sie mit der rechten Maustaste auf den Befehl oder das Symbol für den Befehl und
2. Wählen Sie im Popup-Menü die Option **Löschen** oder,
  1. Wählen Sie den Befehl **Extras | Anpassen** um das Dialogfeld "Anpassen" zu öffnen und
  2. Ziehen Sie den Befehl aus dem Menü und lassen Sie die Maustaste los, sobald unterhalb des Mauszeigers das Häkchensymbol erscheint.

### So setzen Sie eine der Menüleisten zurück

1. Wählen Sie in der Auswahlliste das Standardmenü 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
  1. Wählen Sie den Befehl **Extras | Anpassen**, um das Dialogfeld "Anpassen" zu öffnen und
  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.

#### **Optionen**

Über das Register **Optionen** können Sie zusätzliche Funktionen der Symbolleiste anpassen.

Die Tooltips 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 Tooltips angezeigt werden sollen oder nicht.

### **Symbolleisten und Fenster wiederherstellen**

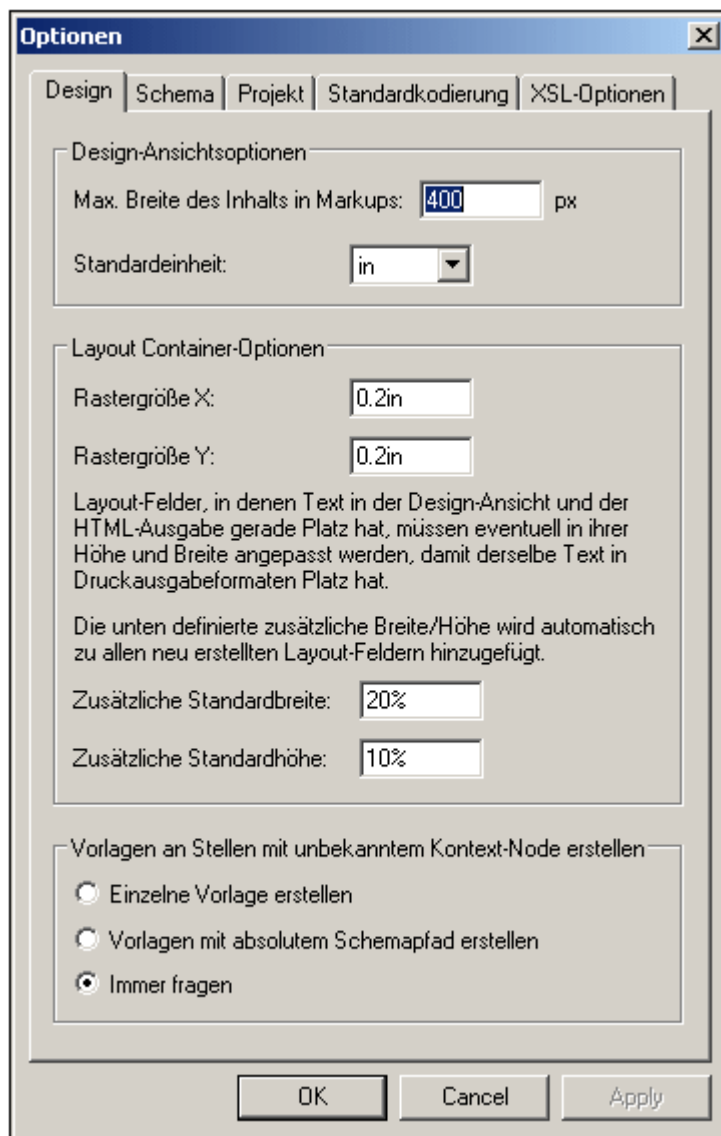
Mit diesem Befehl wird der ursprüngliche Standardzustand von Symbolleisten, Fenster, Eingabehilfen und anderen GUI-Komponenten wiederhergestellt. Damit die Änderungen wirksam werden, müssen Sie StyleVision neu starten.

## Optionen

Mit dem Befehl **Optionen** wird ein Dialogfeld geöffnet, in welchem Sie die Kodierung für die HTML-Ausgabedatei festlegen können.

### Optionen der Design-Ansicht

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*.

### Schemastruktur-Optionen

In der Schemastruktur können Elemente und Attribute alphabetisch in aufsteigender Reihenfolge aufgelistet werden. Aktivieren Sie dazu auf dem Register "Optionen" die entsprechenden Optionen. Standardmäßig werden Attribute alphabetisch und Elemente in der Reihenfolge, wie sie in der Schemastruktur erscheinen, aufgelistet, sofern dies möglich ist.

### Projektoptionen

Wenn Sie in der Projekt-Seitenleiste auf eine XML- oder XSD-Datei doppelklicken, wird je nach den Optionen, die im Dialogfeld "Optionen" auf dem Register "Projekt" festgelegt wurden, eine von drei Aktionen ausgeführt: (i) Bearbeiten der Datei in XMLSpy; (ii) Erstellen eines neuen Designs auf Basis der ausgewählten Datei; (iii) Frage an den Benutzer, welche Aktion ausgeführt werden soll.

### Standardkodierung

Auf dem Register "Standardkodierung" (*Abbildung unten*) können Sie die Standardkodierung für die verschiedenen Ausgabeformate separat definieren. Mit der Kodierung werden die Codepoints für verschiedene Zeichensätze definiert. In der Dropdown-Liste für jede Auswahlliste wird eine Liste der Kodierungsoptionen angezeigt. Wählen Sie die Kodierung für die einzelnen Ausgabeformate und klicken Sie auf **OK**. In jedem neuen SPS, das von nun an erstellt wird, werden die auf diesem Register verwendeten Ausgabekodierungen 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" />`

In der XSLT-für-RTF-Datei sind die Ausgabe-Kodierungs-Informationen im `encoding` Attribut des Stylesheet `xsl: output` Elements registriert:

- `<xsl:output version="1.0" encoding="ISO-8859-1" indent="no" method="text" omit-xml-declaration="yes" media-type="text/rtf" />`
- 

**Anmerkung:** Diese Einstellung ist die Standard-Kodierung für die HTML-Ausgabe, das für neue SPSs verwendet wird. 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](#).

## 19.14 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.

## 19.15 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](#), 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:

- [Inhaltsverzeichnis, Index, Suchen](#)
- [Registrierung, Bestellformular](#)
- [Weitere Befehle](#)

## Inhaltsverzeichnis, Index, Suchen

Mit dem Befehl **Inhaltsverzeichnis** wird die Bildschirmhilfe für StyleVision geöffnet, wobei das Inhaltsverzeichnis auf der linken Seite des Hilfefensters angezeigt wird. Das Inhaltsverzeichnis bietet eine gute Übersicht über das gesamte Hilfedokument. Wenn Sie auf einen Eintrag im Inhaltsverzeichnis klicken, gelangen Sie zu diesem Thema.

Mit dem Befehl **Index** wird die Bildschirmhilfe für StyleVision geöffnet, wobei auf der linken Seite des Hilfefensters der Stichwortindex angezeigt wird. Im Index werden Stichwörter aufgelistet und Sie können durch Doppelklick auf das Stichwort zum jeweiligen Thema navigieren. Wenn ein Stichwort mit mehr als einem Thema verknüpft ist, können Sie das Thema aus einer Liste von Themen auswählen.

Mit dem Befehl **Suchen** wird die Bildschirmhilfe für StyleVision geöffnet, wobei auf der linken Seite des Hilfefensters das Suchdialogfeld angezeigt wird. Um nach einem Begriff zu suchen, geben Sie ihn in das Eingabefeld ein und drücken Sie die Eingabetaste. Das Hilfesystem führt eine Volltextsuche am gesamten Hilfedokument durch und gibt eine Trefferliste zurück. Doppelklicken Sie auf einen Eintrag, um das Thema anzuzeigen.

## Aktivierung, Bestellformular, Registrieren, Updates

### Software-Aktivierung

Nachdem Sie Ihre Altova-Software heruntergeladen haben, können Sie sie entweder mit Hilfe eines kostenlosen Evaluierungs-Keycode oder eines käuflich erworbenen permanenten Lizenzkeycode aktivieren.

- **Kostenloser Evaluierungs-Keycode.** Wenn Sie die Software zum ersten Mal starten, wird das Dialogfeld "Software-Aktivierung" angezeigt. Es enthält eine Schaltfläche, über die Sie einen kostenlosen Evaluierungs-Keycode anfordern können. Geben Sie in das Dialogfeld, das daraufhin angezeigt wird, Ihren Namen, den Namen Ihrer Firma und Ihre E-Mail-Adresse ein und klicken Sie auf "Jetzt anfordern!". Der Evaluierungs-Keycode wird an die von Ihnen angegebene E-Mail-Adresse gesendet und sollte in wenigen Minuten eintreffen. Geben Sie den Schlüssel nun im Dialogfeld "Software-Aktivierung" in das Feld Key-Code ein und klicken Sie auf OK, um Ihr Altova-Produkt zu aktivieren. Die Software wird für einen Zeitraum von 30 Tagen aktiviert.
- **Permanenter Lizenz Keycode.** Das Dialogfeld "Software-Aktivierung" enthält eine Schaltfläche, über die Sie einen permanenten Lizenz-Keycode erwerben können. Wenn Sie auf diese Schaltfläche klicken, gelangen Sie zum Altova Online Shop, wo Sie einen permanenten Lizenz-Keycode für Ihr Produkt erwerben können. Bei Ihrem permanenten Lizenz-Keycode handelt es sich entweder um eine Einzelbenutzerlizenz oder um eine Mehrfachlizenz. Beide Arten werden per E-Mail zugesandt. Eine *Einzelbenutzerlizenz* enthält Ihre Lizenzdaten, Ihren Namen, die Firma, E-Mail-Adresse und Keycode. Eine *Mehrfachlizenz* enthält Ihre Lizenzdaten, ihren Firmennamen und den Keycode. Beachten Sie, dass gemäß Ihrer Lizenzvereinbarung auf den Computern Ihres Unternehmens nur so viele Versionen der Altova-Software installiert werden dürfen, wie lizenziert wurden (Mehrplatzlizenz).

**Anmerkung:** Stellen Sie bitte sicher, dass die von Ihnen in das Registrierungsdialogfeld eingegebenen Daten genau mit denen in Ihrer Lizenz-E-Mail übereinstimmen. Geben Sie außerdem bei Mehrplatzlizenzen den Namen des jeweiligen Benutzers in das Feld "Name" ein.

Das Dialogfeld "Software-Aktivierung" kann jederzeit über den Befehl **Hilfe | Software-Aktivierung** aufgerufen werden.

### Bestellformular

Sobald Sie eine lizenzierte Version von StyleVision bestellen möchten, klicken Sie im Dialogfeld "Software-Aktivierung" (*siehe oben*) auf die Schaltfläche **Permanenter Key-Code erwerben...** im zum sicheren Online-Shop von Altova weitergeleitet zu werden.

### Registrierung

Wenn Sie Ihre Altova-Software nach der Aktivierung das erste Mal starten, erscheint ein Dialogfeld, in dem Sie gefragt werden, ob Sie Ihr Produkt registrieren möchten. Es gibt in diesem Dialogfeld drei Schaltflächen:

- **OK:** Leitet Sie zum Registrierungsformular weiter.
- **Später erinnern:** Ruft ein Dialogfeld auf, in dem Sie auswählen können, wann Sie das nächste Mal erinnert werden möchten.
- **Abbrechen:** Schließt das Dialogfeld und unterdrückt die Anzeige in Zukunft. Wenn Sie das Produkt zu einem späteren Zeitpunkt registrieren möchten, verwenden Sie den Befehl **Hilfe | Registrieren**.

### Auf Updates überprüfen

Überprüft, ob am Altova Server eine neuere Version Ihres Produkts vorhanden ist und zeigt

eine entsprechende Meldung an.

## Weitere Befehle

Der Befehl **Support Center** ist ein Link zum Altova Support Center im Internet. Im Support Center finden Sie Antworten auf häufig gestellte Fragen, Diskussionsforen, in denen Sie Software-Probleme besprechen können und ein Formular, um unsere Mitarbeiter vom technischen Support zu kontaktieren.

Der Befehl **Fragen und Antworten im Web** ist ein Link zur Altova-Fragen- und Antworten-Datenbank im Internet. Diese Datenbank wird ständig aktualisiert, wenn die Mitarbeiter von Altova von unseren Kunden mit neuen Fragen konfrontiert werden.

Der Befehl **Komponenten downloaden** 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.

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](#).

Der Befehl **StyleVision Training** ist ein Link zur Online Training-Seite auf der [Altova Website](#). Hier können Sie aus Online-Kursen, die von ausgebildeten Altova Experten gehalten werden, auswählen.

Mit dem Befehl **Über StyleVision** wird das Willkommensfenster und die Versionsnummer Ihres Produkts angezeigt. Wenn Sie die 64-Bit-Version von StyleVision verwenden, wird dies durch das Suffix (x64) nach dem Applikationsnamen angezeigt. Die 32-Bit-Version hat kein Suffix.





## **Kapitel 20**

---

**Referenz für Programmierer**

## 20 Referenz für Programmierer

### **StyleVision als Automation Server**

StyleVision ist ein Automation Server, d.h. eine Applikation, die anderen Applikationen (so genannten Automation Clients) programmierbare Objekte zur Verfügung stellt. Dadurch hat ein Automation Client direkten Zugriff auf die vom Automation Server bereit gestellten Objekte und Funktionalitäten, sodass sich der Automation Client die Funktionalitäten von StyleVision zunutze machen kann. So kann ein Automation Client z.B. eine XSLT-Datei anhand eines StyleVision SPS generieren. Entwickler können daher Ihre Applikationen verbessern, indem Sie die fertigen Funktionalitäten von StyleVision verwenden.

Die programmierbaren Objekte von StyleVision stehen den Automation Clients über die **StyleVision API**, eine COM API, zur Verfügung. Eine vollständige Beschreibung aller verfügbaren Objekte finden Sie in dieser Dokumentation (siehe Abschnitt [Applikations-API](#)).

## 20.1 Skripterstellung

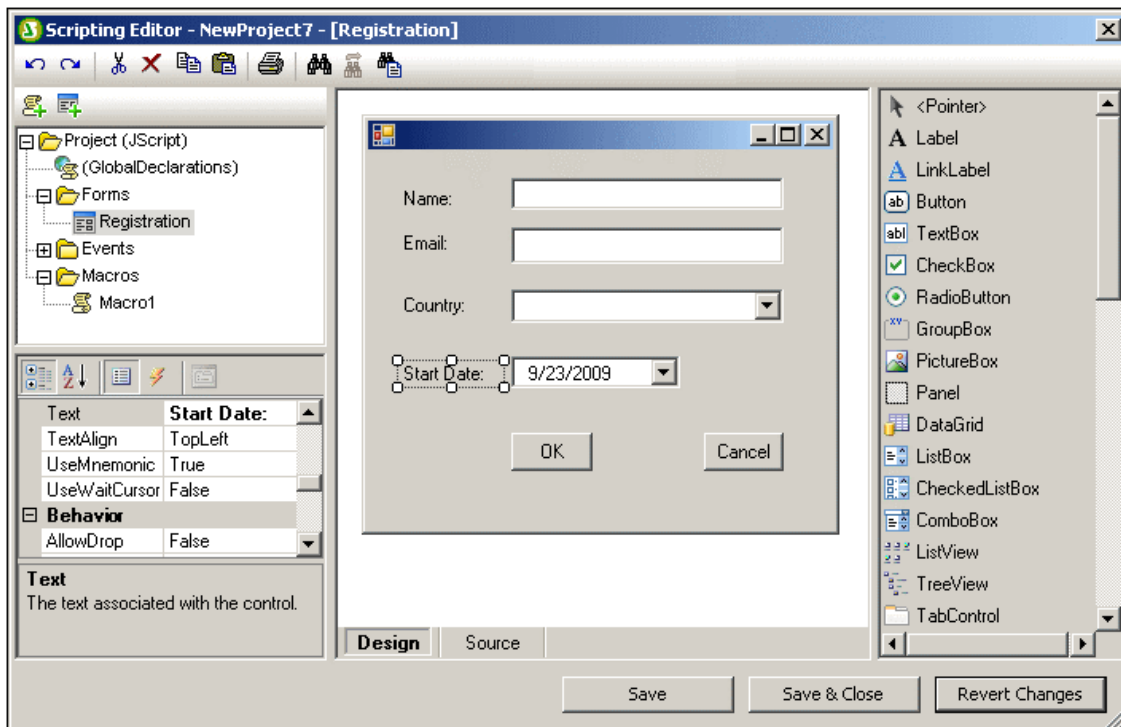
Der Skript-Editor von StyleVision verwendet die Formular-Editor-Komponenten von Microsoft .NET Framework und gestattet somit Zugriff auf Microsoft .NET Framework.

Sie können daher in der Authentic-Ansicht der Altova-Produkte StyleVision, Authentic Desktop, Authentic Browser und XMLSpy Ihre eigenen Makros und Formulare erstellen und verwenden.

**Anmerkung:** Voraussetzung für die Verwendung des Skript-Editors ist die vorherige Installation von Microsoft **.NET Framework 2.0 oder höher**.

### Der Skript-Editor

Der Skript-Editor (*Abbildung unten*) wird in einem separaten Fenster geöffnet und über den Menübefehl **Extras | Authentic Scripts bearbeiten** der StyleVision Benutzeroberfläche aufgerufen. Er kann auch durch Klicken auf das Register "Design" am unteren Rand der Benutzeroberfläche und Auswahl des Eintrags "Authentic Skript" aufgerufen werden. In der Skripting-Umgebung können die Programmiersprachen **JScript** und **VBScript** verwendet werden. Durch Rechtsklick auf den Projekteintrag im Projektfenster und Auswahl des Befehls **Skriptsprache** und anschließend der gewünschten Skriptsprache können Sie die Skriptsprache ändern.



### Was können Sie mit dem Skript-Editor tun?

Sie können im Skript-Editor Formulare, Event-Handler und Makros erstellen, um ein Skripting-Projekt anzulegen. Das Skripting-Projekt kann temporär im Arbeitsspeicher gespeichert werden. Klicken Sie dazu entweder in der Symbolleiste des Skript-Editors auf die Schaltfläche **Speichern** oder **Speichern & schließen**. Das Skripting-Projekt wird nur dann mit dem SPS gespeichert, wenn das SPS anschließend gespeichert wird. Skripts, die im Skripting-Projekt erstellt wurden, können anschließend im SPS-Design verwendet werden, um in der Ausgabe der Authentic-Ansicht Effekte zu produzieren.

**Dokumentation zum Skript-Editor**

Die Dokumentation zur Beschreibung der Skripting-Umgebung (dieser Abschnitt) ist in die folgenden Unterabschnitte gegliedert:

- [eine Übersicht](#), in der der Skript-Editor und Skripting-Projekt überblicksmäßig beschrieben werden.
- eine Liste der Schritte, die zur Erstellung eines Scripting-Projekts erforderlich sind.
- [eine Erläuterung zu globalen Deklarationen](#) sowie ein Beispiel dazu.
- [eine Anleitung zur Erstellung von Formularen](#).
- [eine Erörterung von StyleVision-spezifischen Event Handlern](#).
- [eine Anleitung zur Verwendung von Makros](#) im Skript-Edito und in StyleVision.

## Übersicht

Der Skript-Editor bietet eine Benutzeroberfläche über die Sie (i) Formulare grafisch erstellen können, wobei Sie Komponenten im Formular Formularskripts zuweisen; (ii) Event Handler und (iii) Makros erstellen können.

Diese Formulare, Event Handler und Makros sind in einem einzigen Skripting-Projekt gespeichert, das über den Befehl **Authentic | Authentic-Skripts bearbeiten** aufgerufen werden kann. Das Skripting-Projekt wird im [Skript-Editor](#) von StyleVision geöffnet und kann dort bearbeitet werden. Sobald das Skripting-Projekt gespeichert ist, stehen die Formulare, Event Handler und Makros darin für die Verwendung im SPS-Design der Ausgabe der Authentic-Ansicht zur Verfügung.

Variablen und Funktionen können in einem globalen Deklarations-Skript definiert werden, das immer vor der Ausführung von Makros oder Event Handler Skripts ausgeführt wird.

Dieser Abschnitt enthält eine Übersicht über den Skript-Editor und Skripting-Projekte. Er ist in die folgenden Abschnitte gegliedert:

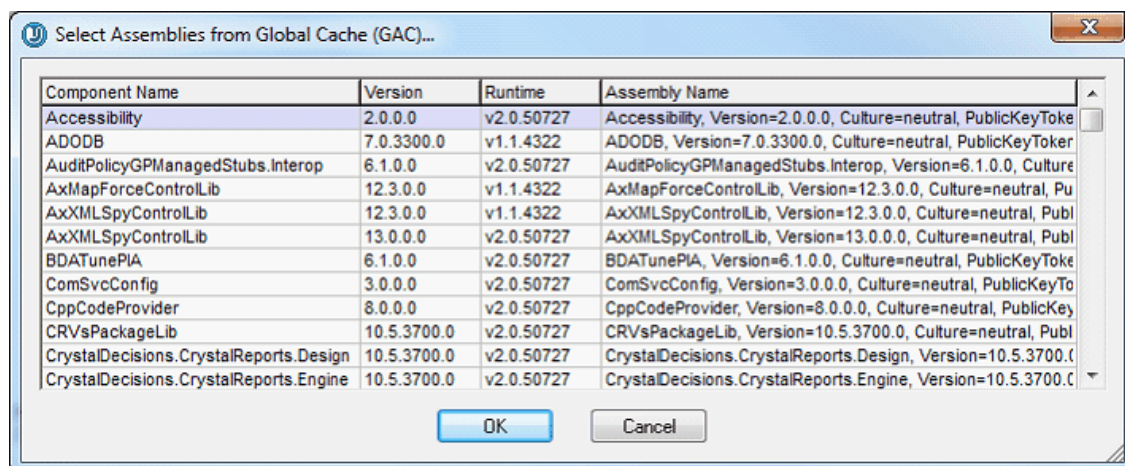
- [Die Benutzeroberfläche des Skript-Editors](#): Hier werden die einzelnen Bestandteile der Skript-Editor-Benutzeroberfläche und ihre Verwendung im Detail beschrieben.
- [Komponenten eines Skripting-Projekts](#): Hier werden die verschiedenen Komponenten beschrieben, aus denen ein Skripting-Projekt besteht.

Nähere Informationen zur Erstellung der verschiedenen Komponenten ([Globale Deklarationen](#), [Formulare](#), [Event Handler](#) und [Makros](#)) finden Sie in den entsprechenden Abschnitten.

### .NET Assemblies

Jedes Skripting-Projekt kann - zusätzlich zu den Standardreferenzen - Referenzen auf .NET Assemblies enthalten. .NET Assemblies können für das gesamte Skripting-Projekt oder für einzelne Makros hinzugefügt werden (mit Hilfe des neuen Befehls `CLR.LoadAssembly` im Quellcode; siehe [Vordefinierte Befehle](#)). Assemblies können z.B. aus dem Global Assembly Cache hinzugefügt werden.

Um eine Assembly hinzuzufügen, klicken Sie mit der rechten Maustaste auf das Projekt oder Makro und wählen Sie im Kontextmenü den Befehl **.NET Assembly hinzufügen | Assembly aus Global Cache (GAC)**.



Dies funktioniert auf dieselbe Art wie bei Visual Studio und bietet nicht nur Zugriff auf das

komplette Microsoft .NET Framework, sondern auch auf jede benutzerdefinierte Assembly.

## Die Benutzeroberfläche des Skript-Editors

Unten sehen die Benutzeroberfläche des Skript-Editors. Er besteht aus den folgenden Bestandteilen:

- einer [Symbolleiste](#)
- einem [Skripting-Projekt-Strukturfenster](#) (linke obere Seite)
- einem [Eigenschafts- und Events-Fenster](#) (links unten)
- einem [Hauptfenster](#) mit den Registern "Design" und "Quelle"
- einer [Formularobjektpalette](#) (rechte Seite)

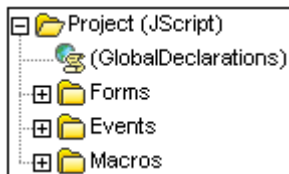
### Skript-Editor-Symbolleiste

Die Symbolleiste des Skript-Editors enthält die folgenden Schaltflächen:

- Schaltflächen für Standarddateibefehle und Standardbearbeitungsbefehle wie **Rückgängig**, **Wiederholen**, **Ausschneiden**, **Kopieren**, **Einfügen**, **Drucken**, **Suchen** und **Ersetzen**. Diese Befehle werden für verschiedene Bearbeitungsprozesse und zum Ausdrucken von Skripting-Projekten verwendet. Beachten Sie, dass die Befehle **Suchen** und **Ersetzen** auf dem Register "Source" des Skript-Editors auf Code angewendet werden.

### Skripting-Projektstruktur

In der Skripting-Projektstruktur (*Abbildung unten*) sehen Sie die verschiedenen Komponenten des Skripting-Projekts, die in vier Hauptzweige gegliedert sind: (i) Globale Deklarationen (GlobalDeclarations), (ii) Formulare (Forms), (iii) Events und (iv) Makros (Macros).



Über das Skripting-Projekt haben Sie Zugriff auf die einzelnen Komponenten des Skripting-Projekts. Um z.B. ein bestimmtes Formular anzuzeigen und zu bearbeiten, erweitern Sie den Ordner "Forms" in der Struktur (*siehe Abbildung oben*), klicken Sie mit der rechten Maustaste auf das gewünschte Formular und wählen Sie im Kontextmenü den Befehl **Open**.

Eine schnellere Methode, um ein Formular, Event oder das Skript für globale Deklarationen zu öffnen, ist, auf das entsprechende Symbol bzw. den Text doppelzuklicken. Um ein Formular oder Makro aus dem Skripting-Projekt zu löschen, klicken Sie mit der rechten Maustaste auf die Komponente und wählen Sie im Kontextmenü den Befehl **Delete**.

Das Fenster "Skripting-Projektstruktur" enthält eine Symbolleiste mit Schaltflächen (*Abbildung unten*).

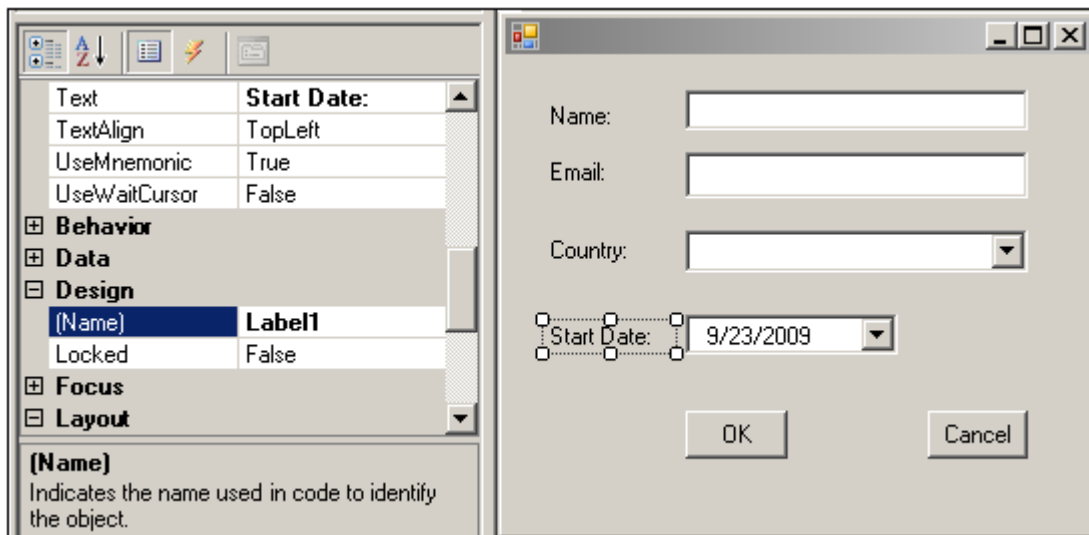


Die Schaltflächen dienen von links nach rechts beschrieben zum (i) [Erstellen eines neuen Makros](#) und (ii) [Erstellen eines neuen Formulars](#). Diese Befehle stehen auch über das Kontextmenü zur Verfügung, das angezeigt wird, wenn Sie mit der rechten Maustaste auf eine Komponente in der Skripting-Projektstruktur klicken.

### Eigenschaften und Events

Im Eigenschafts- und Eventfenster (*Abbildung unten*) werden die folgenden Informationen angezeigt:

- Formulareigenschaften, wenn das Formular ausgewählt ist
- Objekteigenschaften, wenn ein Objekt in einem Formular ausgewählt ist. (In der Abbildung unten sehen Sie auf der linken Seite die Eigenschaften des Objekts, das im Formular auf der rechten Seite ausgewählt ist)
- Formular-Events, wenn ein Formular ausgewählt ist
- Objekt-Events, wenn ein Objekt in einem Formular ausgewählt ist



Um zwischen den Eigenschaften und Events der ausgewählten Komponente hin- und herzuwechseln, klicken Sie auf die Schaltfläche **Properties** (die dritte von links in der Symbolleiste "Properties und Events", *siehe Abbildung oben*) bzw. auf die Schaltfläche **Events** (vierte von links).

Die erste und die zweite Schaltfläche (von links) der Symbolleiste sind die Schaltflächen **Categorized** und **Alphabetical**. Sie dienen zum Anzeigen der Eigenschaften oder Events, entweder nach Kategorie oder in aufsteigender alphabetischer Reihenfolge geordnet.

Wird eine Eigenschaft oder ein Event ausgewählt, wird am unteren Rand des Eigenschafts- und Event-Fensters eine kurze Beschreibung angezeigt.

### Hauptfenster

Im Hauptfenster wird immer nur eine Komponente angezeigt. Es hat ein oder zwei Register, je nachdem, was gerade darin angezeigt wird. Wenn ein globales Deklarationsskript, ein Event oder ein Makro angezeigt wird, so sehen Sie ein einziges Register, das Register "Source", auf dem der Quellcode der ausgewählten Komponente angezeigt wird.

Das Register "Source" unterstützt:

- Syntaxfärbung
- Klappleiste
- Setzen/Löschen von Lesezeichen mit **Strg+F2**
- Autokomplettierungseingabehilfe mit Parameterinfo



- Goto Brace, Goto Brace Extend
- Zoom In / Zoom Out
- vollständige Methoden/Eigenschaftssignatur. Wird neben der Autokomplettierungseingabehilfe angezeigt
- Markierung von geschwungenen Klammern bei der Codeeingabe  

```
if (x == y.GetName(a, b, c()))
```
- Popup-Anzeige bei Mauspositionierung; bei Positionierung der Maus über eine bekannte Methode oder Eigenschaft wird ihre Signatur (und Dokumentation, falls vorhanden) angezeigt

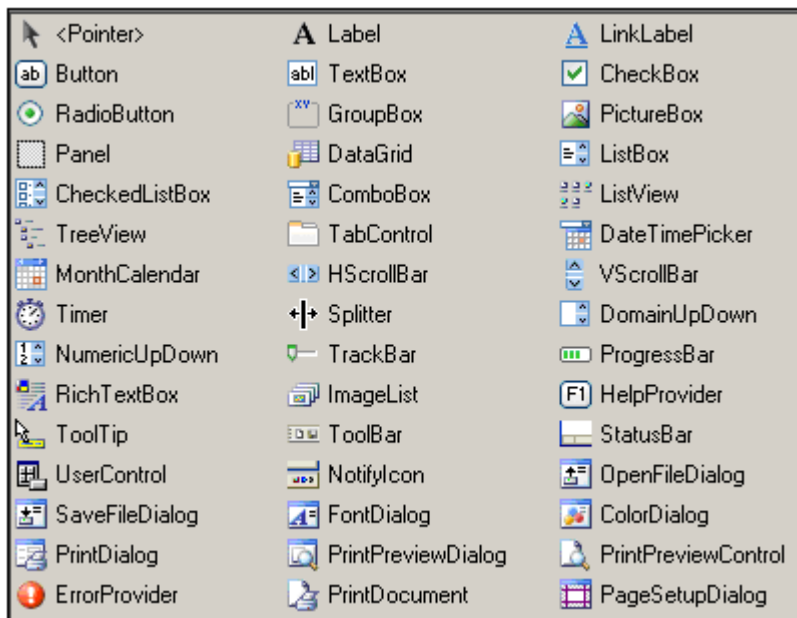
Wenn ein **Formular** angezeigt wird, sehen Sie im Hauptfenster zwei Register: ein Register "Design", in dem das Layout des Formulars angezeigt und aktiviert wird und ein Register "Source", in dem der Quellcode für das Formular angezeigt wird. Sowohl der Inhalt des Registers "Design" als auch der des Registers "Source" kann bearbeitet werden.

**Anmerkung:** Da JScript und VB Script Sprachen typenlose Sprachen sind, werden die Eingabehilfen und die Autokomplettierung nur im Fall von "vollständig qualifizierten Konstrukten" und "vordefinierten" Namen unterstützt.

Wenn Namen mit `objDocument`, `objProject`, `objXMLDat`, oder `objAuthenticRange` beginnen, werden Mitglieder der entsprechenden Schnittstelle angezeigt. Auch Autokomplettierungseingabehilfen und Parameterinformationen werden bei der Bearbeitung angezeigt, können aber auch jederzeit durch Drücken von Strg+Leerzeichen angezeigt werden.

### Formularobjektpalette

Die Formularobjektpalette enthält alle Objekte, die beim Erstellen von Formularen zur Verfügung stehen, und sieht in etwa so aus, wie in der Abbildung unten. Registrierte ActiveX Controls können durch Rechtsklick in das Fenster und Auswahl des Befehls **Add ActiveX Control** zur Formularobjektpalette hinzugefügt werden.



Um ein Objekt aus der Formularobjektpalette einzufügen, klicken Sie in der Palette auf das

gewünschte Objekt und anschließend an die Stelle im Formular, an der das Objekt eingefügt werden soll. Daraufhin wird das Objekt an dieser Stelle eingefügt. In vielen Fällen müssen Sie über das Eigenschafts- und Events-Fenster mehr Eigenschaften des Objekts definieren. Sie können das Objekt an eine andere Stelle verschieben und seine Größe ändern. Außerdem können über das Kontextmenü des ausgewählten Formularobjekts eine Reihe von Bearbeitungsbefehlen wie Zentrieren und Stapeln von Objekten aufgerufen werden.

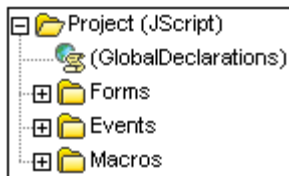
Einige Formularobjekte wie z.B. `Timer` werden nicht zum Formular hinzugefügt, sondern als Bereichskomponenten in einem Bereich am unteren Rand des Hauptfensters erstellt. Sie können das Objekt im Bereich auswählen und die Eigenschaften und Event Handler für das Objekt über das Eigenschafts- und Event-Fenster definieren. Ein Beispiel dafür, wie eine Bereichskomponente behandelt wird, finden Sie unter [Verwendung von Formularen und Befehlen](#).

## Komponenten eines Skripting-Projekts

Ein Altova Skripting-Projekt besteht aus den folgenden vier Hauptbestandteilen:

- *Globalen Deklarationen*, einer Komponente, die die Definitionen von Variablen und Funktionen enthält, die allen Formularen, Makros und Event Handler Skripts im Skripting-Projekt zur Verfügung stehen und von ihnen verwendet werden können.
- *Formularen*, einer Komponente, die alle im Skripting-Projekt definierten Formulare enthält.
- *Events*, einer Komponente, die Event Handler Skripts für alle applikationsbasierten - im Gegensatz zu formularbasierten - Events enthält.
- *Makros*, eine Komponente, die alle im Skripting-Projekt definierten Makros enthält.

Diese Komponenten werden in der Skripting-Projektstruktur des Skript-Editors angezeigt und können hier aufgerufen werden (*Abbildung unten*).



Im Folgenden sehen Sie eine kurze Beschreibung der einzelnen Komponenten.

### Globale Deklarationen

Die Komponente "Global Declarations" (Globale Deklarationen) ist ein Skript, das Variablen und Funktionen enthält, die von Formularen, Event Handlern und Makros verwendet werden können. Die Funktionen nutzen zum Aufrufen der StyleVision Funktionalität die StyleVision API. Wenn Sie im Modul "Global Declarations" eine Variable oder Funktion erstellen, können Sie diese von allen Formularen, Event Handlern und Makros im Skripting-Projekt aus aufrufen.

Um eine Variable oder Funktion hinzuzufügen, öffnen Sie die Komponente "Global Declarations" (durch Rechtsklick in die Skripting-Projektstruktur und Auswahl von **Öffnen**) und bearbeiten Sie das Global Declarations-Skript im Hauptfenster. Fügen Sie in diesem Skript die gewünschte Variable oder Funktion hinzu.

### Formulare

Sie können im Skript-Editor grafisch mit Hilfe einer Palette von Formularobjekten wie Texteingabefeldern und Schaltflächen ein Formular erstellen. So können Sie z.B. ein Formular erstellen, um die Eingabe eines Elementnamens zu akzeptieren und dann alle Instanzen dieses Elements aus dem aktiven XML-Dokument entfernen.

Bei einem solchen Formular kann ein Funktionsskript mit einem Textfeld verknüpft werden, damit dieses eine Eingabevariable akzeptiert, und ein Event Handler kann mit einer Schaltfläche in der Authentic-Ansicht verknüpft werden, um die Ausführung der Löschfunktion zu starten, die in der StyleVision API zur Verfügung steht. Ein Formular wird entweder durch Aufruf dieses Formulars innerhalb einer Funktion (im globalen Deklarationsskript) oder direkt über ein Makro aufgerufen. Nähere Informationen zum Erstellen und Bearbeiten von Formularen finden Sie im Abschnitt [Formulare](#).

### Behandlung von Events

Event Handler Skripts können mit den verschiedensten verfügbaren Events in der Authentic-Ansicht verknüpft werden. Das mit einem Event verknüpfte Skript wird sofort beim Auslösen des Event ausgeführt.

Die meisten Events haben Parameter, die ausführliche Informationen zum Event enthalten. Der Rückgabewert aus dem Skript weist die Applikation normalerweise an, wie sie mit der Verarbeitung fortfahren soll (so gestattet die Applikation z.B. die Bearbeitung nicht).

Ein Event Handler wird ausgeführt, wenn das entsprechende Event im Formular . Nähere Informationen zum Erstellen von Event Handlern finden Sie unter [Event Handler](#).

### **Makros**

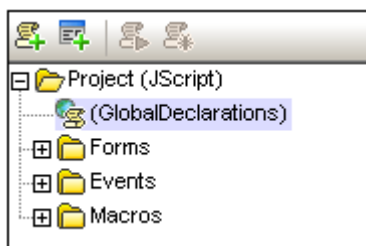
Makros dienen zum Implementieren komplexer oder sich wiederholender Aufgaben. Makros verwenden weder Parameter noch Rückgabewerte.

In einem Makro können alle in den globalen Deklarationen definierten Variablen und Funktion aufgerufen werden und es könnten Formulare für die Benutzereingabe angezeigt werden.

Ein einfaches Beispiel zur Erstellung eines Makros finden Sie unter [Schreiben eines Makros](#). Eine Beschreibung zu den Aufrufmethoden eines Makros finden Sie unter Ausführen von Makros. Ein Makro wird auf der StyleVision Benutzeroberfläche durch Binden an ein Authentic-Event und Auslösen dieses Events ausgeführt.

## Globale Deklarationen

Die Komponente "Global Declarations" ist standardmäßig in jedem Skripting-Projekt vorhanden (siehe *Abbildung unten*) und muss daher nicht erstellt werden. Um Variablen und Funktionen zum Global Declarations-Skript eines Skripting-Projekts hinzuzufügen, müssen Sie das Global Declarations-Skript öffnen und das Codefragment zu diesem Skript hinzufügen. Nähere Informationen hierzu finden Sie unter [Komponenten eines Skripting-Projekts](#).



Klicken Sie zum Öffnen des Global Declarations-Skripts mit der rechten Maustaste in der Skripting-Projektstruktur auf den Eintrag *Global Declarations* (*Abbildung oben*) und wählen Sie den Befehl **Öffnen**. Daraufhin wird das Global Declarations-Skript im Hauptfenster geöffnet.

**Anmerkung:** Bei jeder Ausführung eines Makros bzw. bei jedem Aufruf eines Event Handlers werden die globalen Deklarationen neu initialisiert.

## Formulare

Die Erstellung und Bearbeitung von Formularen im Skript-Editor erfolgt folgendermaßen:

1. [Anlegen eines neuen Formulars](#). Das neue Formular wird erstellt und erhält einen Namen. Es sind Eigenschaften dafür definiert.
2. [Erstellung des Formulars](#). Das Formular wird entworfen, indem Formularobjekte zum Formular hinzugefügt und den verschiedenen Formularobjekten Werte zugewiesen werden.
3. [Skripterstellung für Formular-Events](#). Den Formular-Events werden Skripts zugewiesen.

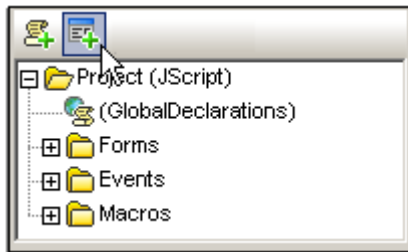
## Erstellung eines neuen Formulars

Die Erstellung eines neuen Formulars im Skript-Editor erfolgt in den folgenden Schritten:

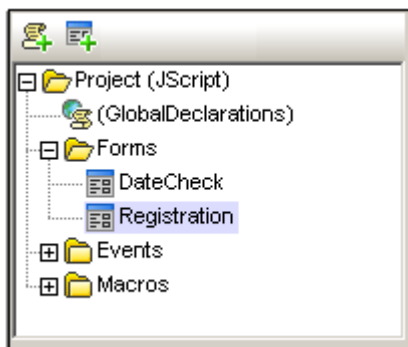
1. [Erstellung eines neuen Formulars und Benennung des Formulars](#)
2. [Definition der Formulareigenschaften](#)

### Erstellung eines neuen Formulars und Benennung des Formulars

Um ein neues Formular zu einem Skripting-Projekt hinzuzufügen, klicken Sie in der Symbolleiste des Projektübersichtsfensters auf die Schaltfläche **Formular hinzufügen** (in der Abbildung unten markiert). Geben Sie den Namen des Formulars ein.

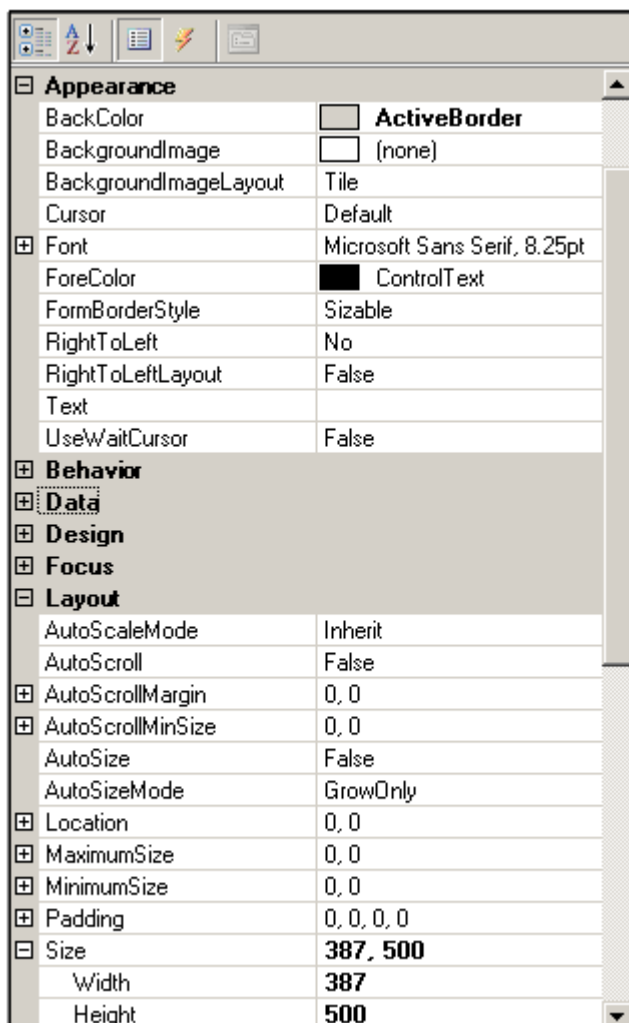


Ein neues Formular wird zum Projekt hinzugefügt und im Hauptfenster angezeigt. Im Skripting-Projektstrukturfenster wird dafür unterhalb der Überschrift "Form" ein Eintrag erstellt. Drücken Sie die Taste F2, um das Formular umzubenennen oder klicken Sie mit der rechten Maustaste auf den Formularnamen und wählen Sie im Kontextmenü den Befehl "Umbenennen" aus. In der Abbildung unten haben wir dem neuen Formular den Namen gegeben.



### Formulareigenschaften

Die Eigenschaften des Formulars wie z.B. seine Größe, Hintergrundfarbe und die Schriftarteneigenschaften können im Eigenschaftsfenster definiert werden. In der Abbildung unten sehen Sie, dass die Eigenschaftswerte für "size" und "background-color" in den Kategorien *Layout* und *Appearance* fett markiert erscheinen.



### Testen eines Formulars

Sie können ein Formular im Skript-Editor testen, indem Sie mit der rechten Maustaste in das Projektübersichtsfenster klicken und den Befehl **Formular testen** auswählen. Das Formular selbst wird mit allen Formularobjekten geladen. Beachten Sie allerdings, dass Code, der mit verschiedenen Formularsteuerelementen verknüpft ist, **nicht** ausgeführt wird. Um den Code zu testen, müssen Sie außerhalb des Skript-Editors ein Makro ausführen, das das Formular enthält.



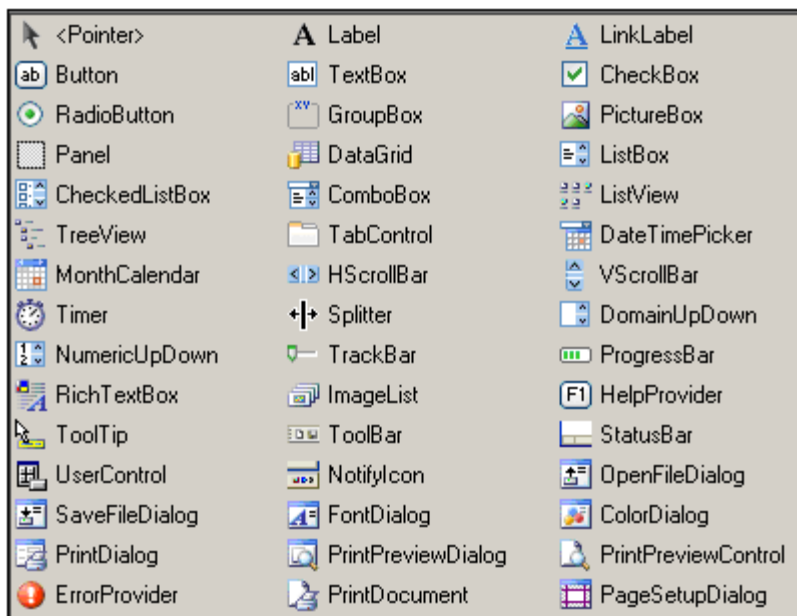
## Formulardesign und Formularobjekte

Gehen Sie beim Entwerfen eines Formulars folgendermaßen vor:

- Platzieren Sie ein Objekt aus der [Formularobjektpalette](#) in das Formulardesign.
- Weisen Sie dem Objekt Werte für die [Eigenschaften einzelner Formularobjekte zu](#).
- [Weisen Sie Skripts für formularbasierte Events zu](#).

### Die Formularobjektpalette

Die Formularobjektpalette enthält alle Objekte, die beim Erstellen von Formularen zur Verfügung stehen, und sieht in etwa so aus, wie in der Abbildung unten. Registrierte ActiveX Controls können durch Rechtsklick in das Fenster und Auswahl des Befehls **Add ActiveX Control** zur Formularobjektpalette hinzugefügt werden.



Um ein Objekt aus der Formularobjektpalette einzufügen, klicken Sie in der Palette auf das gewünschte Objekt und anschließend an die Stelle im Formular, an der das Objekt eingefügt werden soll. Daraufhin wird das Objekt an dieser Stelle eingefügt. In vielen Fällen müssen Sie über das Eigenschafts- und Events-Fenster mehr Eigenschaften des Objekts definieren. Sie können das Objekt an eine andere Stelle verschieben und seine Größe ändern. Außerdem können über das Kontextmenü des ausgewählten Formularobjekts eine Reihe von Bearbeitungsbefehlen wie Zentrieren und Stapeln von Objekten aufgerufen werden.

Einige Formularobjekte wie z.B. `Timer` werden nicht zum Formular hinzugefügt, sondern als Bereichskomponenten in einem Bereich am unteren Rand des Hauptfensters erstellt. Sie können das Objekt im Bereich auswählen und die Eigenschaften und Event Handler für das Objekt über das Eigenschafts- und Event-Fenster definieren. Ein Beispiel dafür, wie eine Bereichskomponente behandelt wird, finden Sie unter [Verwendung von Formularen und Befehlen](#).

Im Folgenden sind einige der am häufigsten verwendeten Objekte beschrieben:



**Beschriftung:** Fügt Textfelder wie Überschriften oder Feldbeschreibungen hinzu.

-  **Schaltfläche:** Fügt eine Schaltfläche hinzu. Auch Bitmaps können diesen Schaltflächen als Hintergrundbild zugewiesen werden.
-  **Kontrollkästchen:** Fügt ein Kontrollkästchen hinzu, über das Auswahlmöglichkeiten vom Typ Ja/Nein erfolgen können.
-  **Auswahlliste:** Fügt eine Auswahlliste hinzu, über die der Benutzer eine Option aus einem Dropdown-Menü auswählen kann.
-  **Listenfeld:** Fügt ein Listenfeld hinzu, das eine Liste von Einträgen zur Auswahl anzeigt.
-  **Textfeld:** Dient zur Eingabe einer einzigen Textzeile.
-  **Mehrzeiliges Bearbeitungsfeld:** Dient zur Eingabe mehrzeiliger Texte.

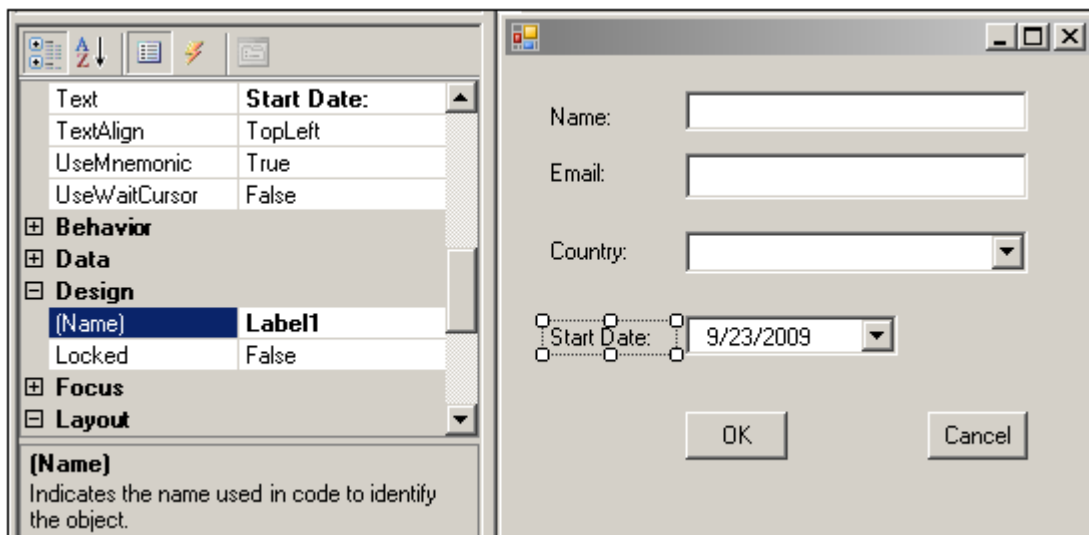
### Erstellen von Objekten und Definieren ihrer Eigenschaften

Um im Formular ein Objekt zu erstellen, wählen Sie das gewünschte Objekt in der Formularobjektpalette zuerst aus und klicken Sie anschließend an die Stelle im Formular, an der es eingefügt werden soll. Nachdem das Objekt eingefügt wurde, können Sie seine Größe anpassen und es an eine andere Stelle im Formular ziehen.

Wenn ein Objekt im Design ausgewählt ist, können Sie seine Eigenschaften im Eigenschafts- und Event-Fenster definieren. Klicken Sie in der Symbolleiste des Eigenschafts- und Event-Fensters auf das Eigenschaftssymbol, um eine Liste der Eigenschaften des Objekts anzuzeigen.

So wurde z.B. in der Abbildung unten das Beschriftungsobjekt mit dem Text *Start Date* im Design ausgewählt. Im Eigenschafts- und Event-Fenster wird der Name des Objekts (welcher der Name ist, der verwendet wird, um das Objekt im Code zu identifizieren - in der Abbildung unten `Label1`) in der Eigenschaftskategorie *Design* angegeben; in diesem Fall ist der Name des Objekts `Label1`.

Der Beschriftungstext (also der Text, der im Formular angezeigt wird) muss als Wert der Eigenschaft *Text* in der Eigenschaftskategorie *Appearance* eingegeben werden.



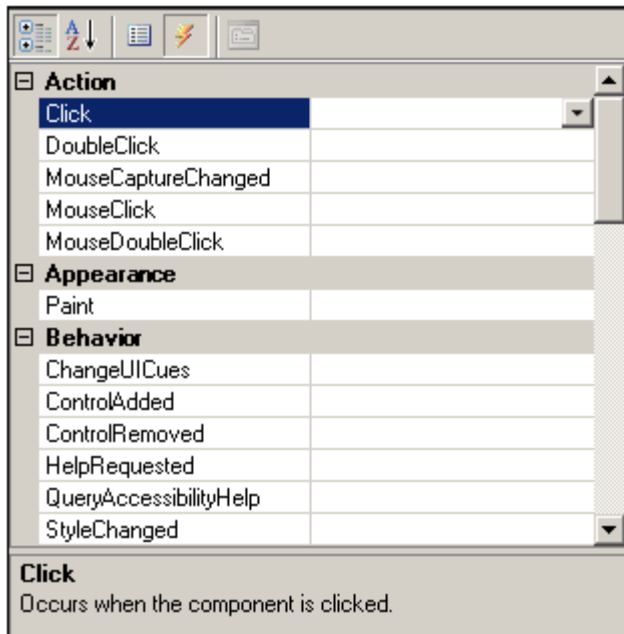
Um andere Objekteigenschaften zuzuweisen, geben Sie dafür Werte in das Eigenschafts- und Event-Fenster ein.

**Testen eines Formulars**

Sie können ein Formular im Skript-Editor testen, indem Sie mit der rechten Maustaste in das Projektübersichtsfenster klicken und den Befehl **Formular testen** auswählen. Das Formular selbst wird mit allen Formularobjekten geladen. Beachten Sie allerdings, dass Code, der mit verschiedenen Formularsteuerelementen verknüpft ist, **nicht** ausgeführt wird. Um den Code zu testen, müssen Sie außerhalb des Skript-Editors ein Makro ausführen, das das Formular enthält.

## Formular-Events

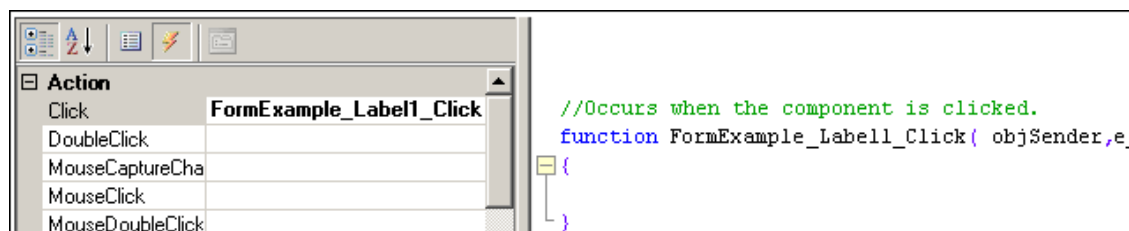
Wenn ein Objekt im Design ausgewählt ist, werden beim Klicken auf die Schaltfläche "Events" in der Symbolleiste des Eigenschafts- und Events-Fensters (*vierte Schaltfläche von links*) alle für dieses Objekt verfügbaren Events angezeigt (*siehe Abbildung unten*). Diese können entweder nach Kategorie (*Abbildung unten*) oder alphabetisch angezeigt werden.



Sie können für jedes Event den Namen eines bestehenden Event Handlers oder einer Funktion eingeben. Alternativ dazu können Sie:

- durch Doppelklick auf ein Event (i) auf dem Register *Source* des Hauptfensters ein leeres Funktionsskript erstellen und (ii) eine Verknüpfung der neu erstellten Funktion mit dem ausgewählten Event herstellen.
- durch Doppelklick auf eine Schaltfläche im Register "Design" den Handler Stub direkt im Codefenster generieren.

Der Screenshot unten wurde aufgenommen, nachdem auf das *Click* Event doppelgeklickt wurde. Beachten Sie, dass im Hauptfenster eine leere Event Handler-Funktion namens *FormExample\_Label1\_Click* erstellt wurde und dass diese Funktion im Eigenschafts- und Event-Fenster mit dem *Click* Event verknüpft wurde.



Geben Sie den gewünschten Skripting-Code ein und speichern Sie das Projekt.

## Schreiben des erforderlichen Skripts

Nachdem das visuelle Design des Formulars abgeschlossen ist, werden Formularobjekte normalerweise mit passenden Skripts verknüpft. Im Beispiel unten wird ein Skript gezeigt, das Farben hinzufügt, wenn man auf eine Schaltfläche klickt. Das Script wird als Event Handler für

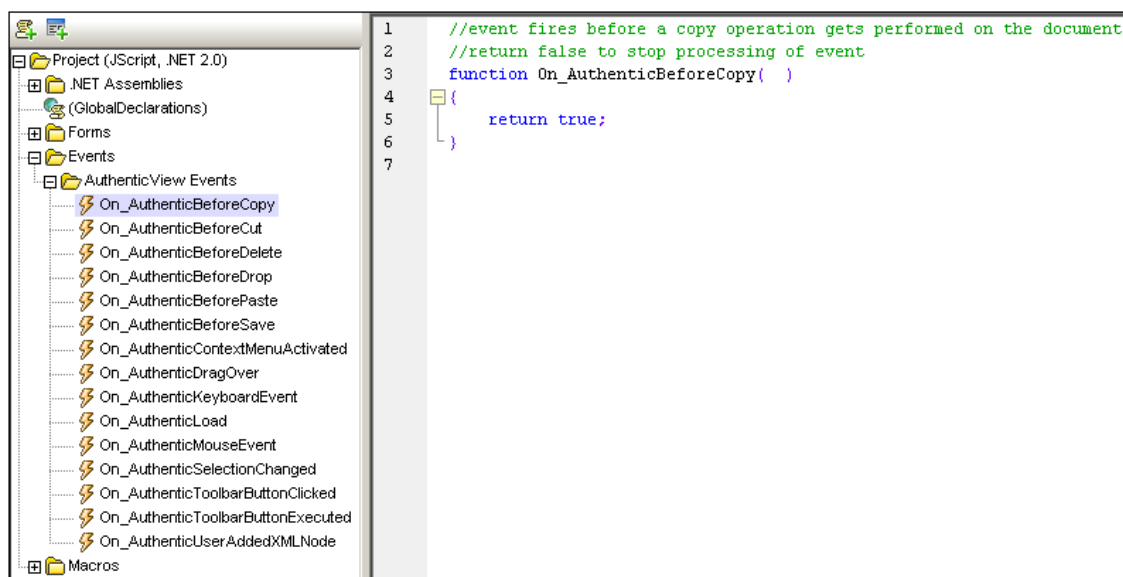
das `Click` Event der Schaltfläche `Button1` eingefügt (das Event steht im Eigenschafts- und Event-Fenster zur Verfügung, wenn die Schaltfläche im Design ausgewählt wird):

```
function FormExample_Button1_Click(objSender, e_EventArgs)
{
 // Sets the ForeColor (red) of the button.
 objSender.ForeColor = CLR.Static("System.Drawing.Color").Red;
 // Sets the BackColor (blue) of the button.
 objSender.BackColor = CLR.Static("System.Drawing.Color").Blue;
 // Sets the form BackColor (green).
 objSender.FindForm().BackColor = CLR.Static("System.Drawing.Color"
).Green;
}
```

## Events

Der Ordner "Events" des Skripting-Projekts (*siehe Abbildung unten*) enthält einen Ordner für Events der Authentic-Ansicht. Für jedes Event im Ordner können Event Handler Skripts geschrieben werden.

Um Zugriff auf das Event Handler Skript eines dieser Events zu erhalten, klicken Sie mit der rechten Maustaste auf das Event und wählen Sie im Kontextmenü den Befehl **Öffnen**. Das Skript wird im Hauptfenster angezeigt (*siehe Abbildung unten*) und kann dort bearbeitet werden. Nachdem Sie mit der Bearbeitung fertig sind, speichern Sie die Änderungen durch Klicken auf die Schaltfläche **Speichern** in der Symbolleiste des Skript-Editors.



Beachten Sie die folgenden Punkte:

- Bei Event Handlern müssen die Funktionsüberschriften den exakt richtigen Event-Namen haben, da der Event Handler sonst nicht aufgerufen wird.
- Sie können in Makros und Event Handlern lokale Variablen und Hilfsfunktionen definieren.

## Makros

Makros können sich wiederholende oder komplizierte Aufgaben automatisieren. In der Skripting-Umgebung können Sie ein Skript erstellen, das Applikationsfunktionen sowie eigene benutzerdefinierte Funktionen aufruft. Dadurch steht Ihnen eine gute Methode zum Automatisieren von Aufgaben in StyleVision zur Verfügung. Dieser Abschnitt über Makros ist in die folgenden Unterabschnitte gegliedert:

- [Erstellen und Bearbeiten eines Makros](#): Hier wird beschrieben, wie man ein neues Makro erstellt und ein bestehendes bearbeitet.
- [Ausführen eines Makros](#): Hier wird erklärt, wie ein Makro von Skript-Editor ausgeführt werden kann.
- [Debuggen](#): Hier wird beschrieben, wie Makros debugged werden können.

### Wichtige Punkte zu Makros

Im Folgenden finden Sie eine Zusammenfassung der wichtigsten Punkte zu Makros.

- Sie können beliebig viele Makros zum aktiven Skripting-Projekt hinzufügen. Diese Makros werden in der Altova Skripting-Projektdatei (`.asprj`-Datei) gespeichert.
- In einem Makro verwendete Funktionen können als globale Deklaration gespeichert werden. Alle globale Deklarationen werden auch in der Altova Skripting-Projektdatei (`.asprj`-Datei) gespeichert.
- Das Makro kann durch Ausführung vom Skript-Editor aus getestet und debuggt werden.
- StyleVision kann eine globale Skripting-Projektdatei haben und ein zweites Skripting-Projekt, das dem aktuell geladenen Projekt zugewiesen ist und jederzeit aktiv ist; die Makros stehen beiden Projekten zur Verfügung. Nähere Informationen dazu finden Sie unter [Ausführen eines Makros](#).

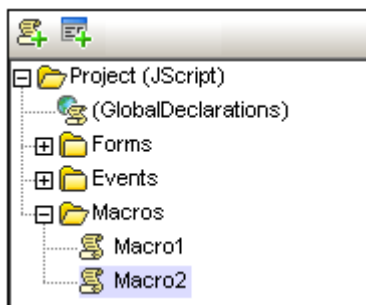
## Erstellen und Bearbeiten eines Makros

Mit Hilfe der folgenden Operationen können Sie ein neues Makro erstellen und ein vorhandenes bearbeiten.

### Erstellen eines neuen Makros

Klicken Sie mit der rechten Maustaste in der Skripting-Projektstruktur auf den Ordner "Macro" und wählen Sie im Kontextmenü den Befehl **Makro hinzufügen**. (Der Befehl **Makro hinzufügen** kann auch im Kontextmenü jedes Elements in der Skripting-Projektstruktur ausgewählt werden.) Klicken Sie alternativ dazu in der Symbolleiste der Skripting-Projektstruktur auf die Schaltfläche **Neues Makro**.

Das neu erstellte (und leere) Makro wird im Hauptfenster angezeigt und der Name des Makros wird in der Titelleiste des Skript-Editors angezeigt (*Abbildung unten*).



### Benennen oder Umbenennen eines Makros

Um einem Makro einen Namen zu geben oder es umzubenennen, klicken Sie im Skripting-Projekt auf den Makronamen und drücken Sie die Taste F2 oder klicken Sie mit der rechten Maustaste auf den Makronamen und wählen Sie im Kontextmenü den Befehl "Umbenennen" aus.

### Öffnen eines Makros

Um ein Makro zu öffnen, klicken Sie mit der rechten Maustaste im Ordner "Macros" der Skripting-Projektstruktur auf das Makro (*siehe Abbildung oben*) und wählen Sie den Befehl **Öffnen**. Das Makro wird im Hauptfenster angezeigt und sein Name wird in der Titelleiste des Skript-Editors angezeigt (*Abbildung unten*). Alternativ dazu können Sie ein Makro in der Skripting-Projektstruktur doppelklicken, um es im Hauptfenster zu öffnen.

**Anmerkung:** Wenn Sie in der Skripting-Projektstruktur auf ein Makrosymbol doppelklicken, wird das Makro im Hauptfenster geöffnet.

### Bearbeiten des Makros

Um ein Makro zu bearbeiten, geben Sie Code ins Hauptfenster ein bzw. bearbeiten Sie diesen. Der folgende Code erstellt, zum Beispiel, das Formular `Form1` im Speicher und zeigt es dann an. `Form1` muss bereits (mit Hilfe der [Formularerstellung](#) des Scripting Editors) erstellt worden sein bevor dieses Makro ausgeführt wird.

```
objForm = CreateForm('Form1');
objForm.ShowDialog();
```

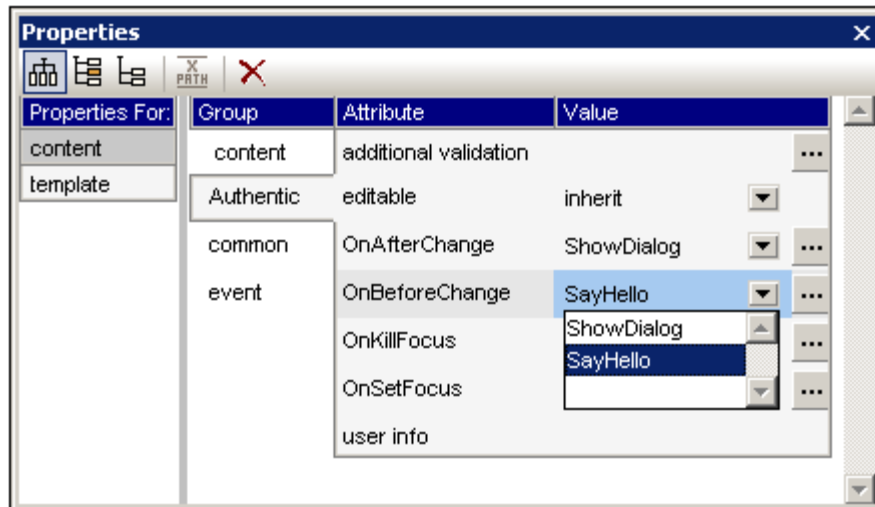
**Anmerkung:** Makros unterstützen Parameter oder Rückgabewerte nicht.



## Ausführen eines Makros

Um ein Makro auszuführen, müssen Sie es zuerst an ein Authentic-Event binden. Das Makro wird ausgeführt, wenn das Authentic-Event ausgelöst wird. Beispielsweise könnten Sie ein Makro ausführen bevor Sie Text in Authentic bearbeiten. Gehen Sie dazu wie folgt vor:

1. Wählen Sie in der Design-Ansicht bearbeitbaren Inhalt aus, für den das Makro erstellt werden soll.
2. Wählen Sie in der Seitenleiste "Eigenschaften" (*Abbildung unten*) die Eigenschaftsgruppe "Authentic" der betreffenden Komponente aus.



3. Wählen Sie das Authentic-Ereignis, das das Makro auslösen wird und öffnen Sie die Dropdown-Liste der Auswahlliste *Wert* für dieses Event. Alle Makros, die im Scripting-Projekt des SPS definiert sind, werden aufgelistet. Wählen Sie das Makro aus, das Sie dieser Funktion zuweisen möchten.
4. In der Authentic-Ansicht wird das Makro ausgeführt, wenn ein beliebiges Authentic-Event, dem ein Makro zugewiesen ist, ausgelöst wird.

Im Abschnitt [Authentic Scripting >> Makros](#) finden Sie eine detaillierte Beschreibung zur Verwendung von Makros in Design-Dokumenten von StyleVision.

### Debuggen eines Makros

Sie können ein Makro mit Hilfe eines installierten Debuggers zu debuggen. Klicken Sie dazu auf das Dropdown-Symbol des Registers der Authentic-Ansicht und schalten Sie die Option *Skripts im Debug-Modus ausführen* ein.



Ab nun wird, wann immer ein Makro durch die Authentic-Ansicht ausgelöst wird, dieses Makro in einem Debugger laufen, den Sie auswählen können. Wenn Sie Skripts außerhalb von StyleVision ausführen, zum Beispiel in Authentic Browser, und das Skript im Debug-Modus ausführen möchten, aktivieren Sie das Kontrollkästchen Authentic-Skripts außerhalb von StyleVision ausführen, das sich im dialogfeld "Eigenschaften" auf dem Register "Authentic" befindet (Aufruf mit dem Befehl **Datei | Eigenschaften**).

Daraufhin wird das Dialogfeld "Just-In-Time" angezeigt (*Abbildung unten*), in dem die auf dem Rechner verfügbaren Debugger aufgelistet werden. Wählen Sie den gewünschten Debugger aus und klicken Sie auf **Yes**.

Der ausgewählte Debugger wird gestartet.

## Anmerkungen zur Programmierung

Beachten Sie bitte die folgenden Anmerkungen zur Programmierung:

- Alle Namespaces und Typen der folgenden .NET Assemblies können standardmäßig im Microsoft .NET Framework aufgerufen werden:

```
System
System.Data
System.Design
System.Drawing
System.Windows.Forms
System.XML
```

Zusätzliche Assemblies können über das [Kontextmenü des Projekts](#) oder dynamisch im Quellcode (zur Laufzeit) mit Hilfe von [CLR.LoadAssembly](#) zum Skripting-Projekt hinzugefügt werden.

- Für Out-Parameter von Methoden der XMLSpy API werden in JScript spezielle Variablen benötigt. Unten sehen Sie einige Beispiele.

```
var dictionary = CLR.Create("System.Collections.Generic.Dictionary<
System.String,
System.String>");
dictionary.Add("1", "A");
dictionary.Add("2", "B");

// use JScript method to access out-parameters
var strOut = new Array(1);
if (dictionary.TryGetValue("1", strOut)) // TryGetValue will set
the out parameter
 alert(strOut[0]); // use out parameter
```

- .NET-Methoden, für die Integer-Argumente benötigt werden, sollten nicht direkt mit JScript Zahlenobjekten aufgerufen werden, bei denen es sich um Gleitkommawerte handelt.

Anstelle von:

```
var objCustomColor = CLR.Static("System.Drawing.Color"). FromArgb(
128, 128, 128);
```

verwenden Sie:

```
var objCustomColor = CLR.Static("System.Drawing.Color"). FromArgb(
Math.floor(128), Math.floor(128), Math.floor(128));
```

- Zum Iterieren durch .NET Collections kann sowohl die JScript Enumerator-Technologie als auch die .NET Iterator-Technologie verwendet werden:

Beispiel:

```
// iterate using the JScript iterator
var itr = new Enumerator(coll);
for (; !itr.atEnd(); itr.moveNext())
 alert(itr.item());

// iterate using the .NET iterator
var itrNET = coll.GetEnumerator();
while(itrNET.MoveNext())
 alert(itrNET.Current);
```

- .NET-Vorlagen können wie unten gezeigt instantiiert werden:

```
var coll = CLR.Create("System.Collections.Generic.List<System.String>");
```

oder

```
CLR.Import("System");
CLR.Import("System.Collections.Generic");
var dictionary = CLR.Create("Dictionary<String, Dictionary<String, String >>");
```

- .NET Enum-Werte werden folgendermaßen aufgerufen:

```
var enumValStretch = CLR.Static("System.Windows.Forms.ImageLayout").Stretch;
```

- Anstelle von numerischen Werten können jetzt Enumerations-Literale, wie in den Altova-Typbibliotheken definiert, verwendet werden.

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

## Vordefinierte Befehle

In diesem Abschnitt finden Sie eine Liste von:

- [vordefinierten Befehlen](#)  
[alert](#)  
[conform](#)  
[doevents](#)  
[createForm](#)  
[lastform](#)  
[prompt](#)  
[showform](#)  
[watchdog](#)
- und Befehlen zur [.NET-Interoperabilität](#).  
[CLR.Create](#)  
[CLR.Import](#)  
[CLR.LoadAssembly](#)  
[CLR.ShowImports](#)  
[CLR.ShowLoadedAssemblies](#)  
[CLR.Static](#)

## Vordefinierte Befehle

Die folgenden vordefinierten Befehle stehen zur Verfügung.

**ShowForm(strFormName : String)**

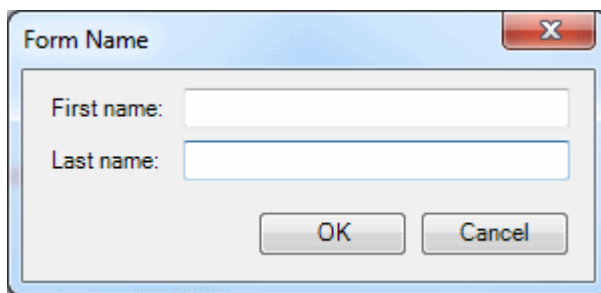
Instantiiert ein neues Formularobjekt anhand des angegebenen Formularnamens und zeigt es sofort als Dialogfeld an.

**Rückgabewert:** Eine Zahl, die für das generierte Dialogfeldergebnis DialogResult (System.Windows.Forms.DialogResult) steht.

Beispiel:

```
var dialogResult = ShowForm("FormName");
```

Zeigt das Formular "FormName" als Dialogfeld an:



Das Ergebnis (DialogResult) kann z.B. folgendermaßen überprüft werden:

```
if (dialogResult == CLR.Static("System.Windows.Forms.DialogResult").
 OK)
 alert("ok");
else
 alert("cancel");
```

**CreateForm(strFormName : String)**

Instantiiert ein neues Formularobjekt anhand des vorgegebenen Formularnamens.

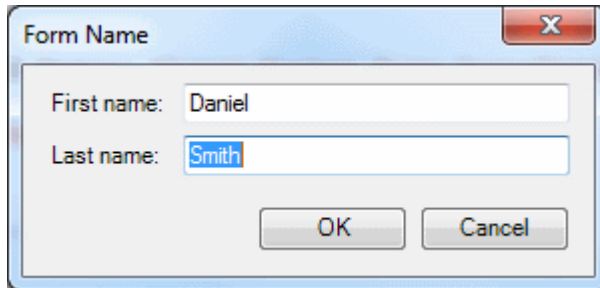
**Rückgabewert:** Das Formularobjekt (System.Windows.Forms.Form) des angegebenen

Namens oder `null`, wenn kein Formular dieses Namens vorhanden ist.

Beispiel:

```
var myForm = CreateForm("FormName");
if (myForm != null)
{
 myForm.textboxFirstName.Text = "Daniel";
 myForm.textboxLastName.Text = "Smith";
 var dialogResult = myForm.ShowDialog();
}
```

Zeigt das Formular "FormName" als Dialogfeld an - TextBoxes werden initialisiert:



Das Ergebnis (DialogResult) kann z.B. folgendermaßen überprüft werden:

:

```
if (dialogResult == CLR.Static("System.Windows.Forms.DialogResult").
 OK)
 alert("ok");
else
 alert("cancel");
```

#### **lastform**

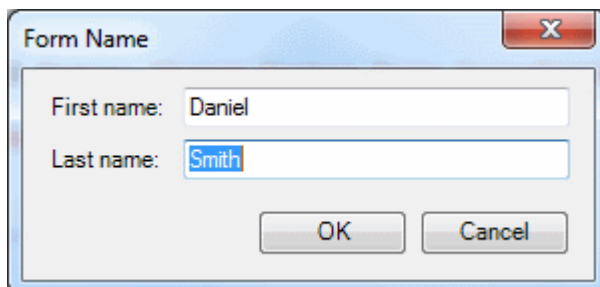
Mit Hilfe dieses globalen Felds kann das zuletzt erstellte Formularobjekt jederzeit wieder aufgerufen werden.

**Rückgabewert:** Gibt eine Referenz auf das letzte Formularobjekt (`System.Windows.Forms.Form`) zurück, das erfolgreich mittels `CreateForm()` oder `ShowForm()` instantiiert wurde.

Beispiel:

```
CreateForm("FormName");
if (lastform != null)
{
 lastform.textboxFirstName.Text = "Daniel";
 lastform.textboxLastName.Text = "Smith";
 var dialogResult = lastform.ShowDialog();
}
```

Zeigt das Formular "FormName" als Dialogfeld an - TextBoxes werden initialisiert (ähnlich wie `CreateForm` im Beispiel oben):



**doevents()**

Verarbeitet alle Windows-Meldungen, die sich derzeit in der Nachrichtenwarteschlange befinden.

*Rückgabewert:* Keiner

Beispiel:

```
for (i=0; i < nLongLastingProcess; ++i)
{
 // do long lasting process

 doevents(); // process windows messages; give UI a chance to
 update
}
```

**watchdog( bEnable : boolean)**

Bei CPU-intensiven Skripts, deren Verarbeitung lange dauert, fragt der Watchdog den Benutzer, ob das Skript abgebrochen werden soll. Mit Hilfe der `watchdog()` Methode können Sie dieses Verhalten aktivieren bzw. deaktivieren. Standardmäßig ist der Watchdog aktiviert.

*Rückgabewert:* Keiner

Beispiel:

```
watchdog(false); // disable watchdog - we know the next statement is
CPU intensive but it will terminate for sure
doCPUIntensiveScript();
watchdog(true); // re-enable watchdog
```

**Verwendungstipp:**

Durch Aufruf von `watchdog(true)` können Sie den Watchdog auch zurücksetzen. Dies empfiehlt sich manchmal vor der Ausführung langer speicherintensiver Aufgaben, um sicherzustellen, dass diesen der maximale vom Skript her zulässige Verarbeitungsspeicher zur Verfügung steht.

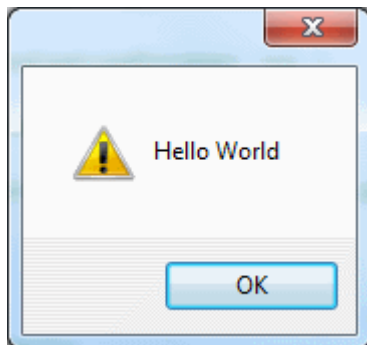
**alert(strMessage : String) oder MsgBox(strMessage : String)**

Zeigt eine bestimmte Meldung über ein Meldungsfeld an. Der Benutzer muss zum Fortfahren auf OK klicken.

*Rückgabewert:* Keiner

Beispiel:

```
alert("Hello World");
```

**confirm(strMessage : String)**

Öffnet ein Dialogfeld, in dem die angegebene Bestätigungsmeldung angezeigt wird.

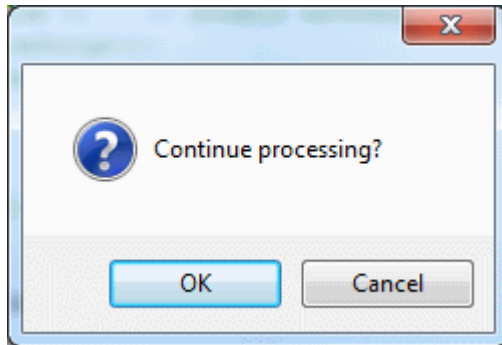
Eine Bestätigungsbox dient oft dazu, etwas zu überprüfen oder zu akzeptieren. Der Benutzer muss entweder auf "OK" oder auf "Abbrechen" klicken, um fortfahren zu können.

*Rückgabewert:* Ein Boolescher Wert, der für die Antwort des Benutzers steht. Wenn der Benutzer auf "OK" klickt, gibt das Dialogfeld "true" zurück, wenn der Benutzer auf "Abbrechen"

klickt, gibt das Dialogfeld "false" zurück.

Beispiel:

```
if (confirm("Continue processing?") == false)
 return;
```



**prompt(strMessage : String, strDefault : String)**

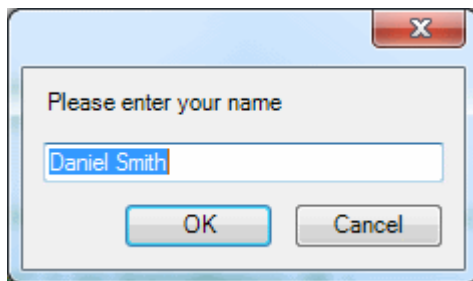
Öffnet ein Dialogfeld, in dem die angegebene Eingabeaufforderung sowie ein Textfeld-Steuererelement mit einer Standardantwort angezeigt wird.

Ein solches Aufforderungskästchen dient oft zur Eingabe eines String-Werts.

**Rückgabewert:** Ein String, der den Textfeldwert enthält oder Null, wenn der Benutzer auf "Abbrechen" geklickt hat.

Beispiel:

```
var name = prompt("Please enter your name", "Daniel Smith");
if (name != null)
 alert("Hello " + name + "!");
```



### Befehle zur .NET-Interoperabilität

Zur verbesserten Interoperabilität mit dem .NET Framework stehen unter CLR zusätzliche Funktionen zur Verfügung.

**CLR.Import(strNamespaceCLR : String)**

Dies ist das Skripting-Äquivalent zur C# / VB.Net Verwendung des Schlüsselworts "imports". Damit können Sie den angegebenen Namespace in aufeinander folgenden Aufrufen wie z.B.

CLR.Create() und CLR.Static() auslassen.

**Rückgabewert:** Keiner

Beispiel:

Anstatt immer den vollständigen qualifizierten Namen verwenden zu müssen:

```
if (ShowForm("FormName") == CLR.Static(
 "System.Windows.Forms.DialogResult").OK)
{
 var sName = lastform.textboxFirstName.Text + " " + lastform.
 textboxLastName.Text;
```



```
CLR.Static("System.Windows.Forms.MessageBox").Show("Hello " +
sName);
}
```

können Namespaces importiert und die Kurzform verwendet werden:

```
CLR.Import("System.Windows.Forms");
if (ShowForm("FormName") == CLR.Static("DialogResult").OK)
{
 var sName = lastform.textboxFirstName.Text + " " + lastform.
textboxLastName.Text;
 CLR.Static("MessageBox").Show("Hello " + sName);
}
```

#### Bitte beachten Sie:

Durch den Import eines Namespace wird die entsprechende Assembly nicht zum Skripting-Projekt hinzugefügt bzw. geladen!

Assemblies können mittels ....zum Skripting-Projekt hinzugefügt oder mittels [CLR.LoadAssembly](#) dynamisch (zur Laufzeit) im Quellcode eingefügt werden.

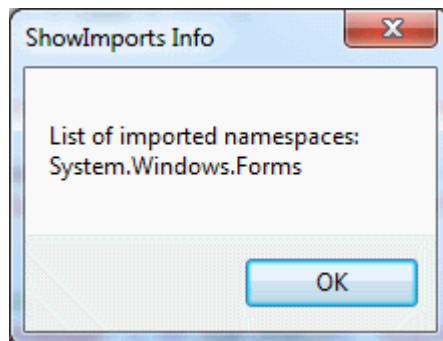
#### CLR.ShowImports()

Öffnet ein Meldungsdialogfeld, in dem die aktuell importierten Namespaces angezeigt werden. Der Benutzer muss zum Fortfahren auf OK klicken.

*Rückgabewert:* Keiner

Beispiel:

```
CLR.ShowImports();
```



#### CLR.LoadAssembly(strAssemblyNameCLR : String)

Lädt die .NET Assembly mit dem angegebenen Assembly-Namen oder Dateipfad.

*Rückgabewert:* ein Boolescher Wert. True, wenn die Assembly geladen werden konnte, false, wenn nicht.

Beispiel:

```
// set clipboard text (if possible)
// System.Windows.Clipboard is part of the PresentationCore assembly,
// so load this assembly first:
if (CLR.LoadAssembly("PresentationCore, Version=3.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35", true))
{
 var clipboard = CLR.Static("System.Windows.Clipboard");
 if (clipboard != null)
 clipboard.SetText("HelloClipboard");
}
```

#### CLR.ShowLoadedAssemblies()

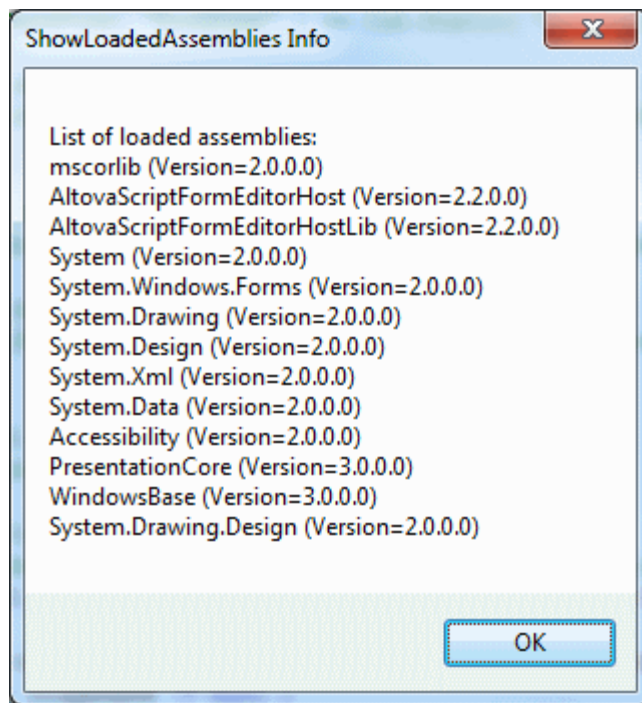
Öffnet ein Meldungsdialogfeld, in dem die aktuell geladenen Assemblies angezeigt werden. Der

Benutzer muss zum Fortfahren auf OK klicken.

*Rückgabewert:* Keiner

Beispiel:

```
CLR.ShowLoadedAssemblies();
```



**CLR.Create(strTypeNameCLR : String, constructor arguments ...)**

Erstelle eine neue .NET-Objektinstanz für den angegebenen Typnamen. Wenn mehr als ein Argument übergeben wird, werden die nachfolgenden Argumente als Argumente für den Konstruktor des .NET-Objekts interpretiert.

*Rückgabewert:* Eine Referenz auf das erstellte .NET-Objekt

Beispiele:

```
var objArray = CLR.Create("System.Collections.ArrayList");

var newItem = CLR.Create("System.Windows.Forms.ListViewItem",
 "NewItemText");

var coll = CLR.Create("System.Collections.Generic.List<System.String>"
);

CLR.Import("System");
CLR.Import("System.Collections.Generic");
var dictionary = CLR.Create("Dictionary<String, Dictionary<String,
 String > >");
```

**CLR.Static(strTypeNameCLR : String)**

Verschafft Zugriff auf .NET-Typen, die keine Instanzen haben und nur statische Mitglieder enthalten.

*Rückgabewert:* Eine Referenz auf das statische .NET-Objekt

Beispiele:

```
var enumValStretch = CLR.Static("System.Windows.Forms.ImageLayout"
).Stretch
```

```
var clipboard = CLR.Static("System.Windows.Clipboard");
clipboard.SetText("HelloClipboard");

if (ShowForm("FormName") == CLR.Static(
 "System.Windows.Forms.DialogResult").OK)
 alert("ok");
else
 alert("cancel");
```

## Verwendung von Formularen und Befehle

Formulare werden folgendermaßen verwendet:

Bei Formularobjekten kann die Formularkomponentenstruktur über Feldzugriff aufgerufen werden

Beispiel: Angenommen Sie haben das folgende Formular:

```
MyForm
 ButtonPanel
 OkButton
 CancelButton
 TextEditor
 AxMediaPlayer1

TrayComponents:
 MyTimer
```

Das Formular kann dann vom Skript aus instantiiert als:

```
var objForm = CreateForm("MyForm");
```

Um eine seiner Komponenten aufzurufen, kann der Feldaufruf verwendet werden:

```
objForm.ButtonPanel.OkButton.Enabled = false;
```

oder

```
objForm.TextEditor.Text = "Hello World";
```

Um Komponenten aus der Taskleiste aufzurufen, verwenden Sie die folgende Methode im Formularobjekt:

```
var objTrayComponent = <A form object>.GetTrayComponent(strComponentName
: String);
```

Um in unserem Beispiel eine Referenz zur Timer-Komponente abzurufen, um deren Verwendung zu aktivieren, verwenden Sie den folgenden Code:

```
var objTimer = objForm.GetTrayComponent("MyTimer");
objTimer.Enabled = true;
```

Bei ActiveX Controls kann das zugrunde liegende COM-Objekt über die OCX-Eigenschaft abgerufen werden:

```
var ocx = lastform.AxMediaPlayer1.OCX; // get underlying COM object
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

## 20.2 Applikations-API

Mit Hilfe der COM-basierten API von StyleVision können andere Applikationen die Funktionalitäten von StyleVision verwenden.

StyleVision und die Applikations-API entsprechen den von Microsoft vorgegebenen allgemeinen Spezifikationen für Automation Server. Die Methoden und Eigenschaften der Applikations-API können mit Skripting-Sprachen wie JavaScript und VBScript und von gängigen Entwicklungsumgebungen wie z.B. solchen, die C, C++, VisualBasic und Delphi verwenden, aufgerufen werden.

### Ausführungsumgebungen für die Applikations-API

Die Applikations-API kann über die folgenden Ausführungsumgebungen aufgerufen werden:

- externe Programme (Beschreibung siehe [unten](#) und in diesem Abschnitt im Kapitel [Übersicht](#))
- vom integrierten Skript-Editor von StyleVision aus. Eine Beschreibung zur Skripting-Umgebung finden Sie im Abschnitt [Skript-Editor](#).
- über ein ActiveX Control, das verfügbar ist, wenn das [Integration Package](#) installiert wurde. Nähere Informationen dazu finden Sie im Abschnitt [ActiveX Integration](#).

### Externe Programme

Im Abschnitt [Übersicht](#) dieses Abschnitts wird beschrieben, wie die Funktionalitäten von StyleVision von externen Programmen aus aufgerufen und automatisiert werden können.

Um die Applikations-API außerhalb von StyleVision verwenden zu können, muss zuerst eine Instanz von StyleVision gestartet werden. Wie dies erfolgt, hängt von der verwendeten Programmiersprache ab. Nähere Informationen zu den einzelnen Sprachen finden Sie unter [Programmierersprachen](#).

Im Wesentlichen wird StyleVision über seine COM Registrierung gestartet. Anschließend wird das mit der StyleVision-Instanz verknüpfte `Application` Objekt zurückgegeben. Je nach COM-Einstellungen kann ein mit einer bereits laufenden StyleVision-Instanz verknüpft Objekt zurückgegeben werden. Es kann jede Programmiersprache verwendet werden, die die Erstellung und den Aufruf von COM-Objekten unterstützt. Die am häufigsten dazu verwendeten Sprachen sind unten aufgelistet.

- [JScript](#) und VBScript Skript-Dateien haben eine einfache Syntax und dienen dazu COM-Objekte aufzurufen. Sie können direkt über eine DOS-Befehlszeile oder durch Doppelklick im Windows Explorer ausgeführt werden. Am besten eignen sich diese Sprachen für einfache Automationsaufgaben.
- [C#](#) ist eine umfangreiche Programmiersprache mit einer breiten Palette an Funktionalitäten. Der Zugriff auf COM-Objekte kann in C# automatisch über Wrapper erfolgen.
- C++ ermöglicht die direkte Steuerung des COM-Zugriffs. Allerdings muss dafür im Vergleich zu anderen Sprachen relativ viel Code geschrieben werden.
- [Java](#): Im Lieferumfang von Altova-Produkten sind native Java-Klassen inkludiert, die als Wrapper für die Applikations-API verwendet werden und dadurch eine Java-Umgebung ermöglichen.
- Weitere nützliche Alternativen sind Visual Basic für Applikationen, Perl und Python.

### Anmerkungen zur Programmierung

Berücksichtigen Sie bitte in Ihrem Client-Code die folgenden Einschränkungen:

- Denken Sie daran, dass bei einem Crash Ihres Client-Code eventuell noch Instanzen von StyleVision im System laufen.
- Behalten Sie Referenzen auf Objekte - vor allem jene von der `XMLData`-Schnittstelle -

nicht länger als nötig im Arbeitsspeicher. Wenn der Benutzer zwischen zwei Aufrufen Ihres Client Eingaben macht, ist nicht gewährleistet, dass diese Referenzen noch gültig sind.

- Vergessen Sie nicht, Dialogfelder zu deaktivieren, wenn die Benutzeroberfläche nicht sichtbar ist.
- Nähere Informationen, wie Sie lästige Fehlermeldungen vermeinden, finden Sie unter Fehlerbehandlung in JScript (und in C# und [Java](#)).
- Geben Sie Referenzen explizit frei, wenn Sie C oder C++ verwenden..

## Übersicht

Diese Übersicht über die Applikations-API ist in die folgenden Abschnitte gegliedert:

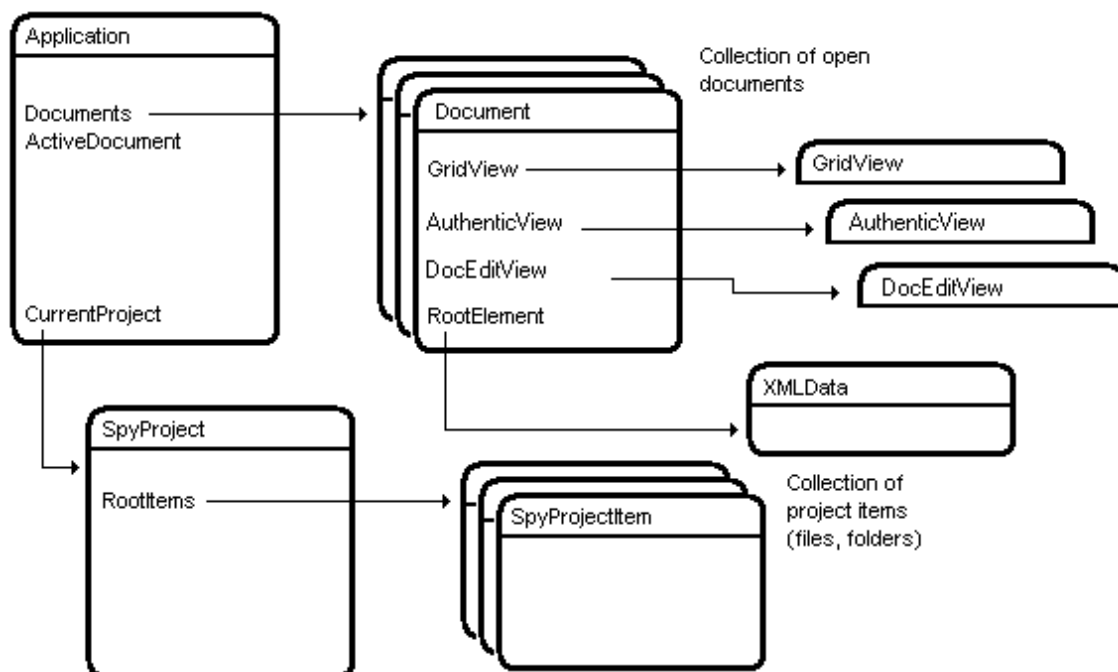
- [Das Objektmodell](#): Hier werden die Beziehungen zwischen den Objekten der Applikations-API beschrieben.
- [Programmiersprachen](#): Hier wird beschrieben, wie Sie die Funktionalitäten der Applikations-API über die gebräuchlichsten Programmiersprachen (JScript, VBScript, C# und Java) aufrufen können. Die Codefragmente aus den mitinstallierten Beispieldateien dienen zum Veranschaulichen einiger grundlegender Mechanismen.
- [DOM und XMLData](#): Hier wird die Beziehung zwischen der `XMLData`-Schnittstelle der Applikations-API und der DOM erläutert.
- [Nicht mehr verwendete Zeilenoperationen in der Authentic-Ansicht](#): Hier finden Sie Informationen zu nicht mehr verwendeten Objekten für Tabellenzeilenoperationen in der Authentic-Ansicht.
- Nicht mehr verwendete Bearbeitungsoptionen in der Authentic-Ansicht: Hier finden Sie Informationen zu nicht mehr verwendeten Objekten für Bearbeitungsoptionen in der Authentic-Ansicht.

## Objektmodell

Der Ausgangspunkt für alle Applikationen, die die Applikations-API verwenden, ist das [Application](#) Objekt. Dieses Objekt enthält allgemeine Methoden wie Import/Export-Funktionen und Referenzen auf offene Dokumente und das offene Projekt, falls vorhanden.

Um eine Instanz des `Application` Objekts zu erstellen, rufen Sie von VisualBasic aus `CreateObject("XMLSpy.Application")` oder von der Entwicklungsumgebung Ihre Wahl eine ähnliche Funktion auf, um ein COM-Objekt zu erstellen. Es müssen keine weiteren Objekte erstellt werden, um die komplette Applikations-API zu verwenden (dies ist auch gar nicht möglich). Alle anderen Schnittstellen werden über andere Objekte aufgerufen, wobei das `Application`-Objekt als Ausgangspunkt dient.

In der Abbildung unten sehen Sie die Verknüpfungen zwischen den Hauptobjekten der Applikations-API:



Das `Application`-Objekt besteht aus den folgenden Teilen:

1. Dokumentsammlung und Referenz auf das aktive Dokument.
2. Referenz auf das aktuelle Projekt und Methoden zum Erstellen und Öffnen von Projekten.
3. Methoden zur Unterstützung des Exports und Imports aus/in Datenbanken, Textdateien und Word-Dokumente.
4. URL-Verwaltung.
5. Methoden für Makro-Menübefehle.

Nach Erstellung eines `Application`-Objekts können Sie beginnen, die Funktionalitäten von zu nutzen. In den meisten Fällen werden Sie entweder ein Projekt öffnen und die Dokumente von dort aufrufen oder ein Dokument direkt über die [Documents](#) Schnittstelle öffnen.

## Programmiersprachen

Der Zugriff auf die COM-Schnittstelle erfolgt in den verschiedenen Programmiersprachen auf unterschiedliche Weise. Eine Beschreibung dazu finden Sie anhand einiger Beispiele für die gebräuchlichsten Programmiersprachen (*Links siehe unten*). Anhand der Codefragmente in diesem Abschnitt wird gezeigt, wie grundlegende Funktionalitäten aufgerufen werden. Diese Grundfunktionen befinden sich in Dateien im Ordner API Examples und können sofort verwendet werden. Den Pfad zum Ordner "API Examples" finden Sie unten:

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C: /Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C: /Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

### JScript

Im JScript-Code werden die folgenden grundlegenden Funktionen verwendet:

- [Starten der Applikation oder Anhängen einer laufenden Instanz](#)
- [Einfacher Dokumentzugriff](#)
- [Iteration](#)
- Fehlerbehandlung
- Events

### VBScript

VBScript unterscheidet sich nur syntaktisch von JScript und funktioniert ansonsten auf genau die gleiche Art. Der unten aufgelistete Code enthält ein Beispiel für die Verwendung von VBScript. Nähere Informationen finden Sie in den [JScript-Beispielen](#).

- Events: Zeigt, wie Events mit VBScript verwendet werden.

### C#

Sie können mit Hilfe von C# auf die Funktionalitäten der Application API zugreifen. Der unten aufgelistete Code enthält ein Beispiel dafür, wie Sie die API für bestimmte grundlegende Funktionalitäten aufrufen können.

- [Start StyleVision](#): Startet StyleVision, das als Automation Server registriert ist, oder aktiviert das Programm, wenn StyleVision bereits ausgeführt wird.
- [Open OrgChart.pxf](#): Navigiert zu einem der mit StyleVision installierten Beispieldokumente und öffnet es. Wenn das Dokument bereits offen ist, wird es zum aktiven Dokument.
- [OnDocumentOpened Event On/Off](#): Zeigt, wie die Applikation auf StyleVision Events hört. Ist die Funktion aktiv, erscheint ein Meldungsfeld, nachdem ein Dokument geöffnet wurde.
- [Open ExpReport.xml](#): Öffnet ein weiteres Beispieldokument.
- [Shutdown StyleVision](#): Beendet StyleVision.

### Java

Sie können mit Hilfe von Java auf die Funktionalitäten der Application API zugreifen. [In diesem](#)



[Abschnitt](#) wird erklärt, wie einige grundlegende XMLSpy-Funktionen über Java-Code aufgerufen werden können. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- [Mapping-Regeln für den Java Wrapper](#)
  - [Java-Beispielprojekt](#)
  - [Starten und Beenden der Applikation](#)
  - [Einfacher Dokumentaufruf](#)
  - [Iterationen](#)
  - [Event Handler](#)

## JScript

Dieser Abschnitt enthält eine Liste von JScript-Code, anhand dessen die folgenden grundlegenden Funktionalitäten erläutert werden:

- [Starten der Applikation oder Anhängen einer laufenden Instanz](#)
- [Einfacher Dokumentzugriff](#)
- [Iteration](#)

## Beispieldateien

Sie finden den Code in diesem Abschnitt in den Beispieldateien, die Sie je nach Bedarf testen und modifizieren können. Die JScript-Beispieldateien befinden sich im Ordner `JScript` des Ordners `API Examples`:

|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Die Beispieldateien können auf zwei Arten ausgeführt werden:

- *Über die Befehlszeile:* Öffnen Sie ein Eingabeaufforderungsfenster und geben Sie den Namen eines der Beispiel-Skripts ein (z.B. `Start.js`). Der Windows Scripting Host, der neuere Versionen von Windows (XP, Vista, 7) enthält, führt das Skript aus.
- *Über den Windows Explorer:* Navigieren Sie im Windows Explorer zur JScript-Datei und doppelklicken Sie darauf. Der Windows Scripting Host, der neuere Versionen von Windows (XP, Vista, 7) enthält, führt das Skript aus. Die Befehlskonsole wird nach Ausführung des Skript automatisch geschlossen.

## Applikation starten

Mit dem unten aufgeführten JScript-Code wird die Applikation gestartet und beendet. Wenn bereits eine Instanz der Applikation ausgeführt wird, so wird diese Instanz aufgerufen.

## Skript-Code

Der nachstehende JScript-Code enthält Erklärungen im Code.

```
// Initialize application's COM object. This will start a new instance of the
application and
// return its main COM object. Depending on COM settings, a the main COM object of an
already
// running application might be returned.
try
{
 objStyleVision = WScript.GetObject("", "StyleVision.Application");
```

```
}
catch(err)
{
 Exit("Can't access or create StyleVision.Application");
}

// if newly started, the application will start without its UI visible. Set it to
visible.
objStyleVision.Visible = true;

WScript.Echo("Hello");

objStyleVision.Visible = false; // will shutdown application if it has no more COM
connections
//objStyleVision.Visible = true; // will keep application running with UI visible
```

**Ausführen des Skripts**

Der oben angeführte JScript-Code steht in der Datei `Start.js` im Ordner `JScript` im Ordner `API Examples` zur Verfügung:

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C: /Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C: /Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Um das Skript auszuführen, starten Sie es über ein Eingabeaufforderungsfenster oder über den Windows Explorer.

Einfacher Dokumentaufruf

Im unten aufgelisteten JScript-Code wird gezeigt, wie man Dokumente öffnet, ein Dokument zum aktiven Dokument macht, durch die offenen Dokumente iteriert und Dokumente schließt.

**Ausführen des Skripts**

Der unten aufgelistete JScript-Code steht in der Datei `DocumentAccess.js` im Ordner `JScript` des Ordners `API Examples` zur Verfügung:

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C: /Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C: /Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Um das Skript auszuführen, starten Sie es über ein Eingabeaufforderungsfenster oder über den Windows Explorer.

**Skript-Code**

Der nachstehende JScript-Code enthält Erklärungen im Code.

```
// Initialize application's COM object. This will start a new instance of the
application and
```

```

// return its main COM object. Depending on COM settings, a the main COM object of an
already
// running application might be returned.
try
{
 objStyleVision = WScript.GetObject("", "StyleVision.Application");
}
catch(err)
{
 Exit("Can't access or create StyleVision.Application");
}

// if newly started, the application will start without its UI visible. Set it to
visible.
objStyleVision.Visible = true;

// ***** code snippet for "Simple Document Access"

// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\StyleVision2012\\StyleVisionExamples\\";

// Tell Authentic to open two documents. No dialogs
objDoc1 = objStyleVision.Documents.OpenDocument(strExampleFolder + "OrgChart.pxf");
objStyleVision.Documents.OpenDocument(strExampleFolder + "BiggestCitiesPerContinent.sps"
);
// ***** code snippet for "Simple Document Access"

// ***** code snippet for "Iteration"

// go through all open documents using a JScript Enumerator
for (var iterDocs = new Enumerator(objStyleVision.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
 objName = iterDocs.item().Name;
 WScript.Echo("Document name: " + objName);
}

// go through all open documents using index-based access to the document collection
for (i = objStyleVision.Documents.Count; i > 0; i--)
 objStyleVision.Documents.Item(i).Close();

// ***** code snippet for "Iteration"

//objStyleVision.Visible = false; // will shutdown application if it has no
more COM connections
objStyleVision.Visible = true; // will keep application running with UI visible

```

### Iteration

Im unten aufgelisteten JScript-Code wird gezeigt, wie man durch offene Dokumente iteriert.

### Ausführen des Skripts

Sie können dieses Skript testen, indem Sie den unten aufgelisteten Code in eine Datei kopieren, ihr die Dateierweiterung `.js` geben und die Datei entweder über die Befehlszeile oder den Windows Explorer ausführen. Sie können die Datei auch in den Ordner `JScript` des Ordners `API Examples` kopieren:

|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Um das Skript auszuführen, starten Sie es über ein Eingabeaufforderungsfenster oder über den Windows Explorer.

### Skript-Code

Der nachstehende JScript-Code enthält Erklärungen im Code.

```
// Initialize application's COM object. This will start a new instance of the
// application and
// return its main COM object. Depending on COM settings, a the main COM object of an
// already
// running application might be returned.
try
{
 objSpy = WScript.GetObject("", "XMLSpy.Application");
}
catch(err)
{
 Exit("Can't access or create XMLSpy.Application");
}

// if newly started, the application will start without its UI visible. Set it to
// visible.
objSpy.Visible = true;

// ***** Accessing documents, setting the active document
// *****

// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
// release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\XMLSpy2012\\Examples\\";

// Tell XMLSpy to open two documents. No dialogs
objDoc1 = objSpy.Documents.OpenFile(strExampleFolder + "OrgChart.pxf", false);
objSpy.Documents.OpenFile(strExampleFolder + "ExpReport.xml", false);

// The document currently active can be easily located.
objDoc2 = objSpy.ActiveDocument;

// Let us make sure that the document is shown in grid view.
objDoc2.SwitchViewMode(0); // SPYViewModes.spyViewGrid = 0

// Now switch back to the document opened first
objDoc1.SetActiveDocument();

// ***** Two ways of iterating; close documents
// *****

// Go through all open documents using a JScript Enumerator
bRequiresSaving = false;
for (var iterDocs = new Enumerator(objSpy.Documents); !iterDocs.atEnd(); iterDocs.
moveNext())
 if (iterDocs.item().IsModified)
```

```

 bRequiresSaving = true;

// Go through all open documents using index-based access to the document collection
// and close each document. Comment out to leave documents open.

for (i = objSpy.Documents.Count; i > 0; i--)
 objSpy.Documents.Item(i).Close(false); // Closes the iterated documents

// ***** Shut down the application or not
// *****

//objSpy.Visible = false; // will shutdown application if it has no more COM
//connections
objSpy.Visible = true; // will keep application running with UI visible

```

### Iteration

Im unten aufgelisteten JScript-Code wird gezeigt, wie man durch offene Dokumente iteriert.

### Ausführen des Skripts

Sie können dieses Skript testen, indem Sie den unten aufgelisteten Code in eine Datei kopieren, ihr die Dateierweiterung `.js` geben und die Datei entweder über die Befehlszeile oder den Windows Explorer ausführen. Sie können die Datei auch in den Ordner `JScript` des Ordners `API Examples` kopieren:

|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Um das Skript auszuführen, starten Sie es über ein Eingabeaufforderungsfenster oder über den Windows Explorer.

### Skript-Code

Der nachstehende JScript-Code enthält Erklärungen im Code.

```

// Initialize application's COM object. This will start a new instance of the
// application and
// return its main COM object. Depending on COM settings, a the main COM object of an
// already
// running application might be returned.
try
{
 objStyleVision = WScript.GetObject("", "StyleVision.Application");
}
catch(err)
{
 Exit("Can't access or create StyleVision.Application");
}

// if newly started, the application will start without its UI visible. Set it to
// visible.
objStyleVision.Visible = true;

// ***** code snippet for "Simple Document Access"
// *****

```

```
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
// release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\StyleVision2012\\StyleVisionExamples\\";

// Tell Authentic to open two documents. No dialogs
objDoc1 = objStyleVision.Documents.OpenDocument(strExampleFolder + "OrgChart.pxf");
objStyleVision.Documents.OpenDocument(strExampleFolder + "BiggestCitiesPerContinent.sps"
);
// ***** code snippet for "Simple Document Access"

// ***** code snippet for "Iteration"

// go through all open documents using a JScript Enumerator
for (var iterDocs = new Enumerator(objStyleVision.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
 objName = iterDocs.item().Name;
 WScript.Echo("Document name: " + objName);
}

// go through all open documents using index-based access to the document collection
for (i = objStyleVision.Documents.Count; i > 0; i--)
 objStyleVision.Documents.Item(i).Close();

// ***** code snippet for "Iteration"

//objStyleVision.Visible = false; // will shutdown application if it has no
more COM connections
objStyleVision.Visible = true; // will keep application running with UI visible
```

## C#

Sie können mit Hilfe von C# auf die Funktionalitäten der Application API zugreifen. Zur Erstellung des C#-Code können Sie Visual Studio 2008 oder Visual Studio 2010 verwenden und den Code in einem Visual Studio-Projekt speichern. Erstellen Sie das Projekt folgendermaßen:

1. Fügen Sie in Microsoft Visual Studio mit dem Befehl **Datei | Neu | Projekt** ein neues Projekt hinzu.
2. Fügen Sie durch Auswahl des Befehls **Projekt | Verweis hinzufügen** hinzufügen eine Referenz zur StyleVision Typbibliothek hinzu. Daraufhin wird das Dialogfeld "Verweis hinzufügen" angezeigt, in dem eine Liste der installierten COM-Komponenten angezeigt wird. Wählen Sie die StyleVision Typbibliothekkomponente aus der Liste aus, um sie hinzuzufügen.
3. Geben Sie den gewünschten Code ein.
4. Kompilieren Sie den Code und führen Sie ihn aus.

## C#-Beispielprojekt

Das StyleVision-Paket enthält ein C#-Beispielprojekt, das Sie im C#-Ordner des Ordners API Examples finden:

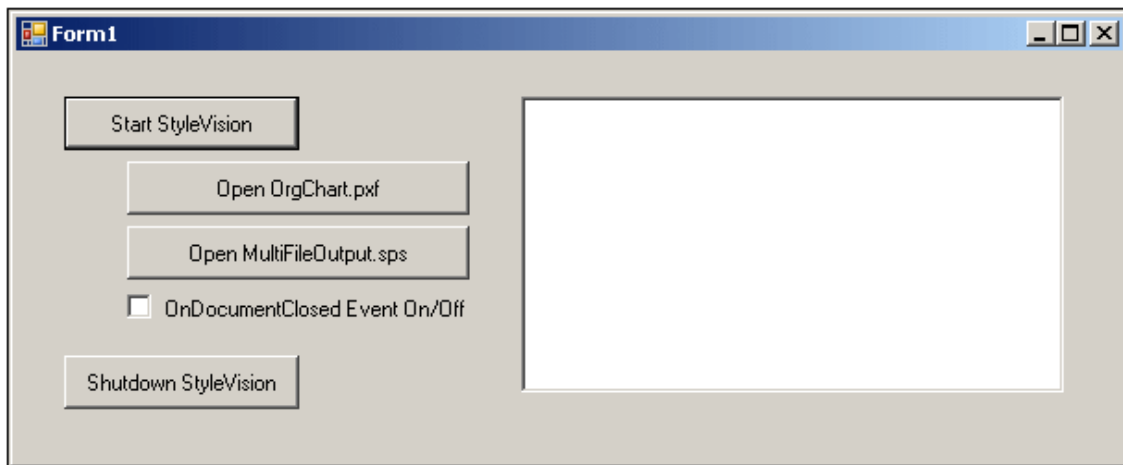
|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Sie können das Projekt innerhalb von Visual Studio 2008 oder Visual Studio 2010 kompilieren und ausführen.

Anhand der unten stehenden Codefragmente wird gezeigt, wie Sie Grundfunktionen der Applikation verwenden können. Der Code ist dem Code im C#-Beispielprojekt im Ordner API Examples Ihres Applikationsordners ähnlich, kann sich aber geringfügig davon unterscheiden.

### Funktionen im Codebeispiel unten

In diesem Beispiel sehen Sie eine einfache Benutzeroberfläche (*Abbildung unten*) mit Schaltflächen, über die grundlegende StyleVision-Operationen aufgerufen werden:



- [Start StyleVision](#): Startet StyleVision, das als Automation Server registriert ist, oder aktiviert das Programm, wenn StyleVision bereits ausgeführt wird.
- [Open OrgChart.pxf](#): Navigiert zu einem der mit StyleVision installierten Beispieldokumente und öffnet es. Wenn das Dokument bereits offen ist, wird es zum aktiven Dokument.
- [Open MultiFileOutput.sps](#): Öffnet ein weiteres Beispieldokument.
- [Shutdown StyleVision](#): Beendet StyleVision.

Sie können den Code (des Codefragments unten oder des C#-Beispiels aus dem Ordner API Examples) beliebig modifizieren und ausführen.

### Kompilieren und Ausführen des Beispiels

Doppelklicken Sie im Ordner API Examples auf die Datei `AutomateStyleVision_VS2008.sln` (Visual Studio 2008) bzw. auf die Datei `AutomateStyleVision_VS2010.sln` (Visual Studio 2010). Alternativ dazu können Sie die Datei auch von Visual Studio aus öffnen (mit dem Befehl **Datei | Öffnen | Projekt/Projektmappe**). Um das Beispiel zu kompilieren und auszuführen wählen Sie **Debuggen | Debuggen starten** bzw. **Debuggen | Starten ohne Debugging**.

### Code des Beispiels

Nachstehend sehen Sie den C#-Code der Grundfunktionen des im `AutomateStyleVision` Beispiel erstellten Formulars (`Form1.cs`). Beachten Sie, dass ich der unten aufgelistete Code geringfügig von dem Code im Formular unter API Examples unterscheiden kann. Der Code ist zum besseren Verständnis mit Kommentaren versehen. Abhängig davon, welche Application

API-Funktionen sie aufrufen, sind Teile dieses Codes auch separat in Unterabschnitten dieses Abschnitts beschrieben.

Der Code besteht im Wesentlichen aus einer Reihe von Handlern für die Schaltflächen der oben gezeigten Benutzeroberfläche.

```
namespace WindowsFormsApplication2
{
 public partial class Form1 : Form
 {
 public Form1()
 {
 InitializeComponent();

 // An instance of StyleVision is accessed via its automation interface.
 StyleVisionLib.Application StyleVision;

 // Location of examples installed with StyleVision
 String strExamplesFolder;

 private void Form1_Load(object sender, EventArgs e)
 {
 // Locate examples installed with StyleVision.
 // REMARK: You might need to adapt this if you have a different major
 version of the product.
 strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") +
 "\\My Documents\\Altova\\StyleVision2012\\StyleVisionExamples\\";
 }

 // Handler for the "Start StyleVision" button
 private void StartStyleVision_Click(object sender, EventArgs e)
 {
 if (StyleVision == null)
 {
 Cursor.Current = Cursors.WaitCursor;

 // If there is no StyleVision instance, create one and make it visible.
 StyleVision = new StyleVisionLib.Application();
 StyleVision.Visible = true;

 Cursor.Current = Cursors.Default;
 }
 else
 {
 // If a StyleVision instance is already running, make sure it's visible.
 if (!StyleVision.Visible)
 StyleVision.Visible = true;
 }
 }

 // Handler for the "Open OrgChart.pxf" button
 private void openOrgChart_Click(object sender, EventArgs e)
 {
 // Make sure there's a running StyleVision instance, and that it's visible
 StartStyleVision_Click(null, null);

 // Open a sample files installed with the product.
 StyleVision.Documents.OpenDocument(strExamplesFolder + "OrgChart.pxf");
 updateListBox();
 }

 // Handler for the "Open MultiFileOutput.sps" button
 private void openMultiFileOutput_Click(object sender, EventArgs e)
 {
 // Make sure there's a running StyleVision instance, and that it's visible
 StartStyleVision_Click(null, null);

 // Open one of the sample files installed with the product.
 StyleVision.Documents.OpenDocument(strExamplesFolder + "MultiFileOutput.sps"
);
 }
 }
 }
}
```



```

 }

 // Handler for the "Shutdown StyleVision" button
 // Shut down the application instance by explicitly releasing the COM object.
 private void shutdownStyleVision_Click(object sender, EventArgs e)
 {
 if (StyleVision != null)
 {
 // Allow shut-down of StyleVision by releasing UI
 StyleVision.Visible = false;

 // Explicitly release the COM object
 try
 {
 int i = System.Runtime.InteropServices.Marshal
.ReleaseComObject(StyleVision);
 while (System.Runtime.InteropServices.Marshal
.ReleaseComObject(StyleVision) > 0) ;
 }
 finally
 {
 // Disallow subsequent access to this object.
 StyleVision = null;
 }
 }
 }

 delegate void addListBoxItem_delegate(string sText);
 // Called from the UI thread
 private void addListBoxItem(string sText)
 {
 listBoxMessages.Items.Add(sText);
 }
 // Wrapper method to call UI control methods from a worker thread
 void syncWithUiThread(Control ctrl, addListBoxItem_delegate methodToInvoke,
String sText)
 {
 // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
 // Control.BeginInvoke: Executes on the UI thread, and calling thread
doesn't wait for completion.
 if (ctrl.InvokeRequired)
 ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
 }

 // Event handler for OnDocumentClosed event
 private void handleOnDocumentClosed(StyleVisionLib.Document i_ipDocument)
 {
 String sText = "";

 if (i_ipDocument.Name.Length > 0)
 sText = "Document " + i_ipDocument.Name + " was closed!";

 // Synchronize the calling thread with the UI thread because
 // COM events are triggered from a working thread
 addListBoxItem_delegate methodToInvoke = new addListBoxItem_delegate
(addListBoxItem);
 // Call syncWithUiThread with the following arguments:
 // 1 - listBoxMessages - list box control to display messages from COM
events
 // 2 - methodToInvoke - a C# delegate which points to the method which will
be called from the UI thread
 // 3 - sText - the text to be displayed in the list box
 syncWithUiThread(listBoxMessages, methodToInvoke, sText);
 }

 private void updateListBox()
 {
 // Iterate through all open documents
 listBoxMessages.Items.Clear();

 for (int i = 1; i <= StyleVision.Documents.Count; i++)

```

```

 {
 StyleVisionLib.Document doc = StyleVision.Documents[i];

 if (doc != null)
 {
 if (checkBoxEventOnOff.Checked)
 doc.OnDocumentClosed += new StyleVisionLib.
_IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);
 else
 doc.OnDocumentClosed -= new StyleVisionLib.
_IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);

 listBoxMessages.Items.Add(doc.Name);
 StyleVisionLib.SchemaSources sources = doc.SchemaSources;

 for (int j = 1; j <= sources.Count; j++)
 {
 StyleVisionLib.SchemaSource source = sources[j];

 if (source != null)
 {
 listBoxMessages.Items.Add("\tSchema file name : " +
source.SchemaFileName + "\n");
 listBoxMessages.Items.Add("\tWorking XML file name : " +
source.WorkingXMLFileName + "\n");
 listBoxMessages.Items.Add("\tIs main schema source : " +
source.IsMainSchemaSource + "\tType name : " + source.TypeName + "\n");
 }
 }
 }
 }
 }
}

```

### Starten und Beenden der Applikation

In den nachstehenden Codefragmenten wurden den Schaltflächen im Beispiel [AutomateStyleVision](#) die Methoden `StartStyleVision_Click` und `ShutdownStyleVision_Click` zugewiesen. Sie starten bzw. beenden die Applikation. Dieses Beispiel befindet sich im Ordner C# des Ordners API Examples (*siehe Datei Form1.cs*):

|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Sie können das Projekt von Visual Studio 2008 oder Visual Studio 2010 aus kompilieren und ausführen.

### Starten von StyleVision

Im folgenden Codefragment aus dem Beispiel [AutomateStyleVision](#) wird gezeigt, wie man die Applikation startet.

```

// Handler for the "Start StyleVision" button
private void StartStyleVision_Click(object sender, EventArgs e)
{
 if (StyleVision == null)
 {
 Cursor.Current = Cursors.WaitCursor;
 }
}

```

```

 // If there is no StyleVision instance, create one and make it visible.
 StyleVision = new StyleVisionLib.Application();
 StyleVision.Visible = true;

 Cursor.Current = Cursors.Default;
 }
 else
 {
 // If an instance of StyleVision is already running, make sure it's
visible if (!StyleVision.Visible)
 StyleVision.Visible = true;
 }
}

```

### Beenden von StyleVision

Im folgenden Codefragment aus dem Beispiel [AutomateStyleVision](#) wird gezeigt, wie man die Applikation beendet.

```

// Handler for the "Shutdown StyleVision" button
// Shut down the application instance by explicitly releasing the COM object.
private void shutdownStyleVision_Click(object sender, EventArgs e)
{
 if (StyleVision != null)
 {
 // Allow shut-down of StyleVision by releasing UI
 StyleVision.Visible = false;

 // Explicitly release the COM object
 try
 {
 int i = System.Runtime.InteropServices.Marshal
.ReleaseComObject(StyleVision);
 while (System.Runtime.InteropServices.Marshal
.ReleaseComObject(StyleVision) > 0) ;
 }
 finally
 {
 // Disallow subsequent access to this object.
 StyleVision = null;
 }
 }
}

```

### Öffnen von Dokumenten

In den unten stehenden Codefragmenten (aus dem Beispiel [AutomateStyleVision](#)) wird gezeigt, wie zwei Dateien über zwei separate Methoden, die zwei Schaltflächen der Benutzeroberfläche zugewiesen wurden, geöffnet werden. Beide Methoden bedienen sich desselben Zugriffsmechanismus auf die Application API: `Documents.OpenFile(string, boolean)`.

Das Beispiel [AutomateStyleVision](#) (siehe *Datei form1.cs*) befindet sich im Ordner C# des Ordners API Examples:

|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Sie können das Projekt von Visual Studio 2008 oder Visual Studio 2010 aus kompilieren und ausführen.

### Codefragment

```
// Handler for the "Open OrgChart.pxf" button
private void openOrgChart_Click(object sender, EventArgs e)
{
 // Make sure there's a running StyleVision instance, and that it's visible
 StartStyleVision_Click(null, null);

 // Open a sample files installed with the product.
 StyleVision.Documents.OpenDocument(strExamplesFolder + "OrgChart.pxf");
 updateListBox();
}

// Handler for the "Open MultiFileOutput.sps" button
private void openMultiFileOutput_Click(object sender, EventArgs e)
{
 if (StyleVision == null)
 StartStyleVision_Click(null, null);

 // Open one of the sample files installed with the product.
 StyleVision.Documents.OpenDocument(strExamplesFolder + "MultiFileOutput.sps"
);
 updateListBox();
}
```

Die zuletzt geöffnete Datei ist die aktive Datei.

### Iterieren durch offene Dokumente

Im unten gezeigten Codefragment (aus dem Beispiel [AutomateXMLSpy](#); siehe Datei *form1.cs*) wird gezeigt, wie man durch offene Dokumente iteriert. Anschließend wird in der Iterationsschleife eine Bedingung überprüft und es wird zwischen der Text- und der Authentic-Ansicht gewechselt.

```
// Handler for the "Toggle view mode" button
private void toggleView_Click(object sender, EventArgs e)
{
 // Make sure there's a running XMLSpy instance, and that it's visible
 StartXMLSpy_Click(null, null);

 // Iterate through open documents and toggle current view between text and
 authentic view.
 foreach (XMLSpyLib.Document doc in XMLSpy.Documents)
 {
 if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewText)
 doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
 else
 doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewText);
 }
}
```

Das Beispiel [AutomateXMLSpy](#) befindet sich im Ordner C# des Ordners API Examples:

|                             |                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C:/Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C:/Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Sie können das Projekt von Visual Studio 2008 oder Visual Studio 2010 aus kompilieren und

ausführen.

## Events

Im unten gezeigten Codefragment (aus dem Beispiel [AutomateXMLSpy](#)) wird der Code für zwei Event Handler aufgelistet. Das [AutomateXMLSpy Beispiel](#) (siehe Datei *form1.cs*) befindet sich im Ordner C# des Ordners API Examples:

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C: /Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C: /Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Sie können das Projekt von Visual Studio 2008 oder Visual Studio 2010 aus kompilieren und ausführen.

```
// Event handler for OnDocumentOpened event
private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
{
 MessageBox.Show("Document " + i_ipDocument.Name + " was opened!");
}

// Remember if the event handler is currently registered.
private bool bEventHandlerIsRegistered = false;

// Handler for button 'OnDocuemntOpened Event On/Off
private void toggleOnDocumentOpenedEvent_Click(object sender, EventArgs e)
{
 if (XMLSpy != null)
 {
 if (bEventHandlerIsRegistered)
 XMLSpy.OnDocumentOpened -= new XMLSpyLib.
_IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
 else
 XMLSpy.OnDocumentOpened += new XMLSpyLib.
_IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);

 bEventHandlerIsRegistered = !bEventHandlerIsRegistered;
 }
}
```

## Events

Im unten gezeigten Codefragment (aus dem Beispiel [AutomateStyleVision](#)) wird der Code für zwei Event Handler aufgelistet. Das Beispiel [AutomateStyleVision](#) (siehe Datei *form1.cs*) befindet sich im Ordner C# des Ordners API Examples:

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C: /Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C: /Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Sie können das Projekt von Visual Studio 2008 oder Visual Studio 2010 aus kompilieren und ausführen.

```
// Event handler for OnDocumentClosed event
```

```

private void handleOnDocumentClosed(StyleVisionLib.Document i_ipDocument)
{
 String sText = "";

 if (i_ipDocument.Name.Length > 0)
 sText = "Document " + i_ipDocument.Name + " was closed!";

 // Synchronize the calling thread with the UI thread because
 // COM events are triggered from a working thread
 addListBoxItem_delegate methodToInvoke = new addListBoxItem_delegate
(addListBoxItem);
 // Call syncWithUiThread with the following arguments:
 // 1 - listBoxMessages - list box control to display messages from COM
events
 // 2 - methodToInvoke - a C# delegate which points to the method which will
be called from the UI thread
 // 3 - sText - the text to be displayed in the list box
 syncWithUiThread(listBoxMessages, methodToInvoke, sText);
}

private void updateListBox()
{
 // Iterate through all open documents
 listBoxMessages.Items.Clear();

 for (int i = 1; i <= StyleVision.Documents.Count; i++)
 {
 StyleVisionLib.Document doc = StyleVision.Documents[i];

 if (doc != null)
 {
 if (checkBoxEventOnOff.Checked)
 doc.OnDocumentClosed += new StyleVisionLib.
_IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);
 else
 doc.OnDocumentClosed -= new StyleVisionLib.
_IDocumentEvents_OnDocumentClosedEventHandler(handleOnDocumentClosed);

 listBoxMessages.Items.Add(doc.Name);
 StyleVisionLib.SchemaSources sources = doc.SchemaSources;

 for (int j = 1; j <= sources.Count; j++)
 {
 StyleVisionLib.SchemaSource source = sources[j];

 if (source != null)
 {
 listBoxMessages.Items.Add("\tSchema file name : " +
source.SchemaFileName + "\n");
 listBoxMessages.Items.Add("\tWorking XML file name : " +
source.WorkingXMLFileName + "\n");
 listBoxMessages.Items.Add("\tIs main schema source : " +
source.IsMainSchemaSource + "\tType name : " + source.TypeName + "\n");
 }
 }
 }
 }
}

```

## Java

Sie können mit Hilfe von Java-Code auf die Applikations-API zugreifen. Damit Sie direkt vom Java-Code aus Zugriff auf den XMLSpy Automation Server haben, müssen sich die unten aufgelisteten Bibliotheken im `classpath` befinden. Sie sind im Applikationsordner im Ordner `JavaAPI` installiert.

- `AltovaAutomation.dll`: ein JNI Wrapper für Altova Automation Server
- `AltovaAutomation.jar`: Java-Klassen zum Aufrufen von Altova Automation Servern

- `XMLSpyAPI.jar`: Java-Klassen, die als Wrapper für die XMLSpy Automation-Schnittstelle dienen
- `XMLSpyAPI_JavaDoc.zip`: eine Javadoc Datei, die die Hilfedokumentation zur Java API enthält

**Anmerkung:** Um die Java API verwenden zu können, müssen sich die DLL und Jar-Dateien im Java Classpath befinden.

### Java-Beispielprojekt

Im Lieferumfang Ihres Produkts ist ein Java-Beispielprojekt enthalten. Sie können das Java-Projekt nach Belieben testen und verwenden. Nähere Informationen zum Java-Beispielprojekt finden Sie im Abschnitt [Java-Beispielprojekt](#).

### Regeln für das Mappen der Application API-Namen auf Java

Für das Mapping zwischen der Application API und dem Java Wrapper gelten die folgenden Regeln:

- **Klassen und Klassennamen**  
Für jedes Interface des XMLSpy Automation Interface gibt es eine Java-Klasse mit dem Namen des Interface.
- **Methodennamen**  
Die Methodennamen im Java Interface sind dieselben wie die in den COM Interfaces, beginnen aber aufgrund der Java-Namenskonventionen mit einem Kleinbuchstaben. Zum Aufrufen von COM-Eigenschaften können dem Eigenschaftsnamen vorangestellte Java-Methoden mit `get` und `set` verwendet werden. Wenn eine Eigenschaft keinen Schreibzugriff ermöglicht, steht keine Setter-Methode zur Verfügung. Beispiel: Für die Eigenschaft `Name` des `Document` Interface stehen die Java-Methoden `getName` und `setName` zur Verfügung.
- **Enumerationen**  
Für jede im Automation Interface definierte Enumeration ist eine Java-Enumeration desselben Namens und mit denselben Werten definiert.
- **Events und Event Handler**  
Für jedes Interface im Automation Interface, das Events unterstützt, steht ein Java-Interface desselben Namens plus `'Event'` zur Verfügung. Um das Überladen von Einzel-Events zu vereinfachen, gibt es eine Java-Klasse mit Standardimplementierungen für alle Events. Der Name dieser Java-Klasse ist der Name des Event Interface plus `'DefaultHandler'`. Beispiel:  
Application: Java-Klasse zum Aufrufen der Applikation  
ApplicationEvents: Events Interface für die Applikation  
ApplicationEventsDefaultHandler: Standard-Handler für ApplicationEvents

### Ausnahmen für Mapping-Regeln

Zu den oben aufgelisteten Regeln gibt es die folgenden Ausnahmen:

| Interface                                  | Java-Name                    |
|--------------------------------------------|------------------------------|
| Document, Methode <code>SetEncoding</code> | <code>setFileEncoding</code> |
| AuthenticView, Methode <code>Goto</code>   | <code>gotoElement</code>     |
| AuthenticRange, Methode <code>Goto</code>  | <code>gotoElement</code>     |
| AuthenticRange, Methode <code>Clone</code> | <code>cloneRange</code>      |

### Dieser Abschnitt

In diesem Abschnitt wird erklärt, wie einige grundlegende XMLSpy-Funktionen über Java-Code aufgerufen werden können. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- [Java-Beispielprojekt](#)
- Starten und Beenden der Applikation
- [Einfacher Dokumentaufruf](#)
- [Iterationen](#)
- [Verwendung von Out-Parametern](#)
- [Event Handler](#)

### Java-Beispielprojekt

Das StyleVision-Installationspaket enthält ein Java-Beispielprojekt, das Sie im Java-Ordner des Ordners API Examples finden:

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Windows XP                  | C: /Dokumente und<br>Einstellungen/<Benutzername>/Eigene Dateien/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/ |
| Windows Vista, Windows<br>7 | C: /Users/<Benutzername>/Documents/<br>Altova/StyleVision20XX/StyleVisionExamples/API<br>/                               |

Dieser Ordner enthält Java-Beispiele für die StyleVision API. Sie können das Beispielprojekt mit Hilfe der Batch-Datei `BuildAndRun.bat`, direkt über die Befehlszeile testen oder Sie können es in Eclipse kompilieren und ausführen. Anleitungen dafür finden Sie weiter unten.

### Dateiliste

Der Ordner für die Java-Beispiele enthält alle zum Ausführen des Beispielprojekts erforderlichen Dateien. Diese Dateien sind unten aufgelistet:

|                                      |                                                                                                                                                                              |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>AltovaAutomation.dll</code>    | Java-COM Bridge: DLL-Teil                                                                                                                                                    |
| <code>AltovaAutomation.jar</code>    | Java-COM Bridge: Java-Bibliotheksteil                                                                                                                                        |
| <code>StyleVisionAPI.jar</code>      | Java-Klassen der Authentic Desktop API                                                                                                                                       |
| <code>RunStyleVision.java</code>     | Java-Beispielquellcode                                                                                                                                                       |
| <code>BuildAndRun.bat</code>         | Batch-Datei zum Kompilieren und Ausführen des Beispielcodes über die Befehlszeile. Es wird ein Ordner benötigt, in dem sich die Java Virtual Machine als Parameter befindet. |
| <code>.classpath</code>              | Hilfdatei Eclipse-Projekt                                                                                                                                                    |
| <code>.project</code>                | Eclipse-Projektdatei                                                                                                                                                         |
| <code>StyleVision_JavaDoc.zip</code> | Javadoc Datei, die die Hilfedokumentation für die Java API enthält                                                                                                           |

### Funktionen in diesem Beispiel



In diesem Beispielprojekt wird StyleVision gestartet und einige Operationen wie das Öffnen und Schließen von Dokumenten werden ausgeführt. StyleVision bleibt danach geöffnet. Sie müssen die Applikation manuell schließen.

### Starten und Beenden der Applikation

Im nachstehenden Code wird gezeigt, wie man die Applikation startet und beendet.

#### Starten der Applikation

Bevor Sie die Applikation starten müssen die entsprechenden Klassen importiert werden (*siehe unten*).

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access XMLSpy Java-COM bridge
05 import com.altova.automation.StyleVision.*;
06 import com.altova.automation.StyleVision.Enums.ENUMApplicationStatus;
07
08 /**
09 * A simple example that starts the COM server and performs a View operations on it.
10 * Feel free to extend.
11 */
12 public class RunStyleVision
13 {
14 public static void main(String[] args)
15 {
16 // An instance of the application.
17 Application stylevision = null;
18
19 // Instead of COM error-handling, use Java exception mechanism
20 try
21 {
22 // Start StyleVision as COM server.
23 stylevision = new Application();
24
25 ENUMApplicationStatus status = ENUMApplicationStatus.eApplicationRunning;
26 do{
27 // Check the application status
28 status = stylevision.getStatus();
29 System.out.println("status : " + status + "\n");
30 } while (status != ENUMApplicationStatus.eApplicationRunning);
31
32 // COM servers start up invisible, so we make the server visible
33 stylevision.setVisible(true);
34
35 ...
36 }
37 }
38 }
39
```

#### Beenden der Applikation

Die Applikation kann wie unten gezeigt beendet werden.

```
1 {
2 // Make sure that StyleVision can shut down properly.
3 if (stylevision != null)
4 stylevision.dispose();
5
6 // Since the COM server was made visible and still is visible, it will keep
7 running
8 // and needs to be closed manually.
9 System.out.println("Now close StyleVision!");
10 }
```

```
9 }
```

### Einfacher Dokumentzugriff

Im unten stehenden Code wird gezeigt, wie man ein Dokument öffnet.

```
1 // Locate samples installed with the product.
2 String strExamplesFolder = System.getenv("USERPROFILE") + "\\My
Documents\\Altova\\StyleVision2012\\StyleVisionExamples\\";
3
4 // We open two files form the product samples.
5 stylevision.getDocuments().openDocument(strExamplesFolder + "OrgChart.pxf");
6 stylevision.getDocuments().openDocument(strExamplesFolder +
"BiggestCitiesPerContinent.sps");
```

### Iterationen

Im nachstehenden Code wird gezeigt, wie man durch offene Dokumente iteriert.

```
01 // Iterate through all open documents
02 for (Document doc: stylevision.getDocuments())
03 {
04 System.out.println("Document name : " + doc.getName() + "\n");
05 SchemaSources sources = doc.getSchemaSources();
06
07 for (int i = 1; i <= sources.getCount(); i++)
08 {
09 SchemaSource source = sources.getItem(i);
10 System.out.println("\tSchema file name : " + source.getSchemaFileName()
+ "\n");
11 System.out.println("\tWorking XML file name : " +
source.getWorkingXMLFileName() + "\n");
12 System.out.println("\tIs main schema source : " +
source.getIsMainSchemaSource() + "\tType name : " + source.getTypeName() + "\n");
13 }
14 }
```

### Verwendung von Out-Parametern

Im unten stehenden Code wird durch offene Dokumente iteriert, wobei jedes davon validiert wird. Für jede Validierung wird mit Hilfe der Ausgabeparameter der Methode "Validation" eine Meldung generiert.

### Event Handler

Im nachstehenden Code wird gezeigt, wie die Applikation auf XMLSpy Events hört und diese verwendet.

```
01 // The following lines attach to the document events using a default implementation
02 // for the events and override one of its methods.
03 // If you want to override all document events it is better to derive your listener
class
04 // from DocumentEvents and implement all methods of this interface.
05 Document doc = stylevision.getActiveDocument();
06 doc.addListener(new DocumentEventsDefaultHandler()
07 {
08 @Override
09 public void onDocumentClosed(Document i_ipDoc) throws AutomationException
10 {
11 System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
```

```
12 }
13 });
14 doc.close();
15 doc = null;
```

## DOM und XML Data

Über die `XMLData` Schnittstelle haben Sie Zugriff auf die gesamte XML-Struktur, die dem aktuellen Dokument zugrunde liegt. Sie benötigen dafür weniger Methoden als DOM und der Zugriff ist einfacher. Die `XMLData` Schnittstelle ist eine möglichst einfache Methode, um vorhandene oder neu erstellte XML-Daten zu lesen und zu bearbeiten. Eventuell sollten Sie jedoch eine DOM-Struktur verwenden, weil Sie auf diese von einer externen Quelle aus zugreifen können oder weil Sie die MSXML DOM-Implementierung vorziehen.

Die unten gezeigten Funktionen `ProcessDOMNode()` und `ProcessXMLDataNode()` konvertieren alle Segmente einer XML-Struktur zwischen `XMLData` und DOM.

Um die `ProcessDOMNode()` Funktion zu verwenden:

- übergeben Sie das Root-Element des zu konvertierenden DOM-Segments in `objNode` und
- übergeben Sie das Plugin-Objekt mit der `CreateChild()` Methode in `objCreator`

Um die `ProcessXMLDataNode()` Funktion zu verwenden:

- übergeben Sie das Root-Element des `XMLData`-Segments in `objXMLData` und
- übergeben Sie das mit MSXML in `xmlDoc` erstellte `DOMDocument`

```

//
// DOM To XMLData conversion
Function ProcessDOMNode(objNode, objCreator)
{
 var objRoot;
 objRoot = CreateXMLDataFromDOMNode(objNode, objCreator);

 If(objRoot) {
 If((objNode.nodeValue != Null) && (objNode.nodeValue.length > 0))
 objRoot.TextValue = objNode.nodeValue;
 // add attributes
 If(objNode.attributes) {
 var Attribute;
 var oNodeList = objNode.attributes;

 For(var i = 0; i < oNodeList.length; i++) {
 Attribute = oNodeList.item(i);

 var newNode;
 newNode = ProcessDOMNode(Attribute, objCreator);

 objRoot.AppendChild(newNode);
 }
 }
 If(objNode.hasChildNodes) {
 try {
 // add children
 var Item;
 oNodeList = objNode.childNodes;

 For(var i = 0; i < oNodeList.length; i++) {
 Item = oNodeList.item(i);

 var newNode;
 newNode = ProcessDOMNode(Item, objCreator);

 objRoot.AppendChild(newNode);
 }
 }
 catch(err) {
 }
 }
 }
 Return objRoot;
}

```

```

Function CreateXMLDataFromDOMNode(objNode, objCreator)
{
 var bSetName = True;
 var bSetValue = True;

 var nKind = 4;

 switch(objNode.nodeType) {
 Case 2: nKind = 5; break;
 Case 3: nKind = 6; bSetName = False; break;
 Case 4: nKind = 7; bSetName = False; break;
 Case 8: nKind = 8; bSetName = False; break;
 Case 7: nKind = 9; break;
 }
 var objNew = Null;
 objNew = objCreator.CreateChild(nKind);

 If(bSetName)
 objNew.Name = objNode.nodeName;

 If(bSetValue && (objNode.nodeValue != Null))
 objNew.TextValue = objNode.nodeValue;

 Return objNew;
}
////////////////////////////////////
// XMLData To DOM conversion

Function ProcessXMLDataNode(objXMLData, xmlDoc)
{
 var objRoot;
 objRoot = CreateDOMNodeFromXMLData(objXMLData, xmlDoc);

 If(objRoot) {
 If(IsTextNodeEnabled(objRoot) && (objXMLData.TextValue.length > 0))
 objRoot.appendChild(xmlDoc.createTextNode(objXMLData.TextValue));

 If(objXMLData.HasChildren) {
 try {
 var objChild;
 objChild = objXMLData.GetFirstChild(-1);

 While(True) {
 If(objChild) {
 var newNode;
 newNode = ProcessXMLDataNode(objChild, xmlDoc);

 If(newNode.nodeType == 2) {
 // child node is an attribute
 objRoot.attributes.setNamedItem(newNode);
 }
 Else
 objRoot.appendChild(newNode);
 }
 objChild = objXMLData.GetNextChild();
 }
 } catch(err) {
 }
 }
 }
 Return objRoot;
}

Function CreateDOMNodeFromXMLData(objXMLData, xmlDoc)
{
 switch(objXMLData.Kind) {
 Case 4: Return xmlDoc.createElement(objXMLData.Name);
 Case 5: Return xmlDoc.createAttribute(objXMLData.Name);
 Case 6: Return xmlDoc.createTextNode(objXMLData.TextValue);
 Case 7: Return xmlDoc.createCDATASection(objXMLData.TextValue);
 Case 8: Return xmlDoc.createComment(objXMLData.TextValue);
 Case 9: Return
 xmlDoc.createProcessingInstruction(objXMLData.Name, objXMLData.TextValue);
 }
}

```

```
Return xmlDoc.createElement(objXMLData.Name);
}
Function IsTextNodeEnabled(objNode)
{
 switch(objNode.nodeType) {
 Case 1:
 Case 2:
 Case 5:
 Case 6:
 Case 11: Return True;
 }
 Return False;
}
```

### Nicht mehr verwendete Zeilenoperationen in der Authentic-Ansicht

Wenn im Schema, auf dem das XML-Dokument basiert, festgelegt ist, dass ein Element wiederholbar ist, kann eine derartige Struktur in der Authentic-Ansicht als Tabelle dargestellt werden. Bei Darstellung in einer Tabelle können Zeilen und ihr Inhalt einzeln bearbeitet werden, wodurch Sie jedes der wiederholbaren Elemente einzeln bearbeiten können. Solche Zeilenoperationen würden von einem externen Skript durchgeführt werden.

Wenn ein externes Skript Zeilenoperationen durchführen soll, sind zwei Schritte erforderlich:

- Im ersten Schritt wird überprüft, ob sich der Cursor gerade in einer Zeile befindet, in der eine Eigenschaft verwendet wird. Eine solche Überprüfung könnte z.B. sein `IsRowInsertEnabled`. Diese Methode gibt das Ergebnis TRUE oder FALSE zurück.
- Wenn der Rückgabewert TRUE ist, kann eine Zeilenmethode wie z.B. `RowAppend` aufgerufen werden. (`RowAppend` hat keine Parameter und gibt keinen Wert zurück.)

Im Folgenden sehen Sie eine Liste von Eigenschaften und Methoden, die für Tabellenoperationen zur Verfügung stehen. Jede Eigenschaft gibt einen booleschen Wert zurück und die Methoden haben keine Parameter.

| Eigenschaft                        | Methode                                                                                  | Tabellenoperationen                         |
|------------------------------------|------------------------------------------------------------------------------------------|---------------------------------------------|
| <code>IsRowInsertEnabled</code>    | <b>RowInsert</b> , wird nun ersetzt durch <a href="#">AuthenticRange.InsertRow</a>       | Operation "Zeile einfügen"                  |
| <code>IsRowAppendEnabled</code>    | <b>RowAppend</b> , wird nun ersetzt durch <a href="#">AuthenticRange.AppendRow</a>       | Operation "Zeile anhängen"                  |
| <code>IsRowDeleteEnabled</code>    | <b>RowDelete</b> , wird nun ersetzt durch <a href="#">AuthenticRange.DeleteRow</a>       | Operation "Zeile löschen"                   |
| <code>IsRowMoveUpEnabled</code>    | <b>RowMoveUp</b> , wird nun ersetzt durch <a href="#">AuthenticRange.MoveRowUp</a>       | XML-Daten eine Zeile nach oben verschieben  |
| <code>IsRowMoveDownEnabled</code>  | <b>RowMoveDown</b> , wird nun ersetzt durch <a href="#">AuthenticRange.MoveRowDown</a>   | XML-Daten eine Zeile nach unten verschieben |
| <code>IsRowDuplicateEnabled</code> | <b>RowDuplicate</b> , wird nun ersetzt durch <a href="#">AuthenticRange.DuplicateRow</a> | Aktuell ausgewählte Zeile duplizieren       |

## **Schnittstellen**

Dieses Kapitel enthält die Referenz der StyleVision Typbibliothek.



**Application****See also****Methoden**

GetDatabaseImportElementList  
GetDatabaseSettings  
GetDatabaseTables  
ImportFromDatabase  
CreateXMLSchemaFromDBStructure

GetTextImportElementList  
GetTextImportExportSettings  
ImportFromText

ImportFromWord

ImportFromSchema

GetExportSettings

NewProject  
OpenProject

AddMacroMenuItem  
ClearMacroMenu

ShowForm

ShowApplication

URLDelete  
URLMakeDirectory

FindFiles

[Quit](#)

**Eigenschaften**

[Application](#)

[Parent](#)

[ActiveDocument](#)

[Documents](#)

CurrentProject

Dialogs

WarningNumber  
WarningText

[Status](#)

[MajorVersion](#)

[MinorVersion](#)

[Edition](#)

[IsAPISupported](#)

[ServicePackVersion](#)

**Beschreibung**

Application ist die Root für alle anderen Objekte. Es ist das einzige Objekt, das Sie mit `CreateObject` (VisualBasic) oder anderen ähnlichen COM-Funktionen erstellen können.

**Beispiel**

```
Dim objSpy As Application
Set objSpy = CreateObject("XMLSpy.Application")
```

**Events****OnShutDown**

**Event:** `OnShutDown()`

**Beschreibung**

Wird direkt vor dem Beenden von StyleVision gesendet.

**ActiveDocument****Siehe auch**

**Eigenschaft:** `ActiveDocument` als [Document](#)

**Beschreibung**

Referenz auf das aktive Dokument. Wenn kein Dokument geöffnet ist, ist `ActiveDocument` null (nothing).

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Application****Siehe auch**

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

**Beschreibung**

Ruft das StyleVision application-Objekt auf.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Documents****Siehe auch**

**Eigenschaft:** `Documents` als [Documents](#)

**Beschreibung**

Sammlung aller offenen Dokumente. Siehe auch Einfacher Dokumentzugriff.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Edition

**Siehe auch**

**Eigenschaft:** [Edition](#) als String

**Beschreibung**

Gibt die Edition der Applikation, also z.B. Enterprise, Professional oder Basic zurück.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsAPISupported

**Siehe auch**

**Eigenschaft:** [IsAPISupported](#) als Boolean

**Beschreibung**

Gibt zurück, ob die API in dieser Version unterstützt wird oder nicht.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MajorVersion

**See also**

**Eigenschaft:** [MajorVersion](#) als Integer

**Beschreibung**

Gibt die Hauptreleasennummer der Applikation zurück.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MinorVersion

**See also**

**Eigenschaft:** [MinorVersion](#) as Integer

**Beschreibung**

Gibt die Nebenrelease-Versionsnummer der Applikation zurück.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

NewDocument

**Methode:** [NewDocument](#) als [Document](#)

**Rückgabewert**

Keiner

**Beschreibung**

Erstellt anhand der vorherigen Vorlage ein neues leeres Dokument.

**Fehler**

- 1000 Das application-Objekt ist ungültig.
- 1005 Fehler beim Erstellen eines neuen Dokuments
- 1006 Das Dokument kann nicht erstellt werden

OpenDocument

**Methode:** `OpenDocument(strFileName als String)` als [Document](#)

**Rückgabewert**

Keiner

**Beschreibung**

Öffnet eine vorhandene SPS-Datei

**Fehler**

- 1000 Das application-Objekt ist ungültig.
- 1002 Ungültige Dateierweiterung.
- 1003 Fehler beim Öffnen des Dokuments
- 1004 Das Dokument kann nicht geöffnet werden

Parent

**Siehe auch**

**Eigenschaft:** `Parent` als [Application](#) (schreibgeschützt)

**Beschreibung**

Ruft das StyleVision application-Objekt auf.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Quit

**Siehe auch**

**Method:** `Quit()`

**Rückgabewert**

Keiner

**Beschreibung**

Diese Methode beendet StyleVision. Alle geänderten Dokumente werden ohne Speicherung der Änderungen geschlossen. Dies gilt auch für ein offenes Projekt.

Wenn StyleVision automatisch von einem Client-Programm als Automation Server gestartet wird, wird die Applikation bei Schließen des Client-Programms nicht automatisch geschlossen, falls noch ein Projekt oder Dokument geöffnet ist. Verwenden Sie die Quit-Methode, um sicherzustellen, dass das Programm automatisch beendet wird.

**Fehler**

1111 Das application-Objekt ist nicht mehr gültig.

ServicePackVersion

**See also**

**Eigenschaft:** [ServicePackVersion](#) als Long

**Beschreibung**

Gibt die Service Pack-Versionsnummer der Applikation zurück, also z.B. 1 für 2010 R2 SP1

**Fehler**

1111 Das application-Objekt ist nicht mehr gültig.

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Status

**Siehe auch**

**Eigenschaft:** [Status](#) als [ENUMApplicationStatus](#)

**Beschreibung**

Gibt den aktuellen Status der laufenden Applikation zurück.

**Fehler**

1111 Das application-Objekt ist nicht mehr gültig.

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Visible

**Siehe auch**

**Eigenschaft:** [Visible](#) als `VARIANT_BOOL`

**Beschreibung**

Setzt oder ruft das Sichtbarkeitsattribut von StyleVision ab.

**Fehler**

1110 Das application-Objekt ist nicht mehr gültig.

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## AppOutputLine

Stellt eine Meldungszeile dar. Diese Zeile kann eine Sammlung von Child-Zeilen enthalten und dadurch eine Baumstruktur von Meldungszeilen bilden.

### Eigenschaften und Methoden

Eigenschaften zum Navigieren im Objektmodell:

[Application](#)

[Parent](#)

Zeilenaufruf:

[GetLineSeverity](#)

[GetLineSymbol](#)

[GetLineText](#)

[GetLineTextEx](#)

[GetLineTextWithChildren](#)

[GetLineTextWithChildrenEx](#)

Eine einzige AppOutputLine-Zeile besteht aus einer oder mehreren untergeordneten Zeilen.

Aufruf der untergeordneten Zeilen:

[GetLineCount](#)

Eine untergeordnete Zeile besteht aus einer oder mehreren Zellen.

Zellaufruf:

[GetCellCountInLine](#)

[GetCellIcon](#)

[GetCellSymbol](#)

[GetCellText](#)

[GetCellTextDecoration](#)

[GetIsCellText](#)

Unterhalb einer AppOutputLine-Zeile können null, eine oder mehrere Child-Zeilen stehen, die selbst den Typ AppOutputLine haben und dadurch eine Baumstruktur bilden.

Aufruf der Child-Zeilen:

[ChildLines](#)

Application

**Eigenschaft:** [Application](#) als [AppOutputLine](#) (schreibgeschützt)

### Beschreibung

Ruft das Objekt der obersten Ebene von application auf.

### Fehler

4100 Das Objekt ist nicht mehr gültig.

4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ChildLines

**Eigenschaft:** [ChildLines](#) als [AppOutputLines](#) (schreibgeschützt)

### Beschreibung

Gibt eine Sammlung der direkten Child-Zeilen der aktuellen Zeile zurück.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetCellCountInLine

**Methode:** `GetCellCountInLine` (*nLine* als Long ) als Long

**Beschreibung**

Ruft die Anzahl der Zellen in der Subzeile ab, die in der aktuellen AppOutputLine-Zeile durch *nLine* angegeben wird.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetCellIcon

**Methode:** `GetCellIcon` (*nLine* als Long , *nCell* als Long ) als Long

**Beschreibung**

Ruft das Symbol (Icon) der Zelle auf, die in der durch *nLine* gekennzeichneten Subzeile der aktuellen AppOutputLine-Zeile durch *nCell* gekennzeichnet ist.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetCellSymbol

**Methode:** `GetCellSymbol` (*nLine* als Long , *nCell* als Long ) als AppOutputLineSymbol

**Beschreibung**

Ruft das Symbol der Zelle auf, die in der durch *nLine* gekennzeichneten Subzeile der aktuellen AppOutputLine-Zeile durch *nCell* gekennzeichnet ist.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetCellText

**Methode:** `GetCellText` (*nLine* als Long , *nCell* als Long ) als String

**Beschreibung**

Ruft den Text der Zelle auf, die in der durch *nLine* gekennzeichneten Subzeile der aktuellen AppOutputLine-Zeile durch *nCell* gekennzeichnet ist.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetCellTextDecoration

**Methode:** `GetCellTextDecoration` (*nLine* als Long, *rCell* als Long) als Long

**Beschreibung**

Ruft die Darstellung der Textzelle auf, die in der durch *nLine* gekennzeichneten Subzeile der aktuellen AppOutputLine-Zeile durch *nCell* gekennzeichnet ist. Es kann sich dabei um einen der [ENUMAppOutputLine\\_TextDecoration](#) Werte handeln.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetIsCellText

**Methode:** `GetIsCellText` (*nLine* als Long, *rCell* als Long) als Boolean

**Beschreibung**

Gibt "true" zurück, wenn es sich bei der durch *nCell* gekennzeichneten Zelle der durch *nLine* gekennzeichneten Subzeile um eine Textzelle handelt.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetLineCount

**Methode:** `GetLineCount` () als Long

**Beschreibung**

Ruft die Anzahl der Subzeilen auf, aus denen die aktuelle Zeile besteht.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetLineSeverity

**Methode:** `GetLineSeverity` () als Long

**Beschreibung**

Ruft den Schweregrad der Zeile ab. Dabei handelt es sich um einen der [ENUMAppOutputLine\\_Severity](#) Werte.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetLineSymbol

**Methode:** `GetLineSymbol` () als [AppOutputLineSymbol](#)

**Beschreibung**

Ruft das Symbol ab, das der gesamten Zeile zugewiesen wurde.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.



4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### GetLineText

**Methode:** `GetLineText` () als `String`

#### Beschreibung

Ruft den Inhalt der Zeile als Text ab.

#### Fehler

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### GetLineTextEx

**Methode:** `GetLineTextEx` (`psTextmatSeparator` als `String` , `psLineSeparator` als `String` ) als `String`

#### Beschreibung

Ruft den Inhalt der Zeile mit Hilfe der angegebenen Abschnitts- und Zeilentrennzeichen als Text ab.

#### Fehler

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### GetLineTextWithChildren

**Methode:** `GetLineTextWithChildren` () als `String`

#### Beschreibung

Ruft den Inhalt der Zeile einschließlich aller Child- und Nachfahrenzeilen als Text ab.

#### Fehler

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### GetLineTextWithChildrenEx

**Methode:** `GetLineTextWithChildrenEx` (`psmatSep` als `String` , `psLineSep` als `String` , `psTabSep` als `String` , `psItemSep` als `String` ) als `String`

#### Beschreibung

Ruft den Inhalt der Zeile einschließlich aller Child- und Nachfahrenzeilen unter Verwendung der angegebenen Abschnitts-, Zeilen-, Tabulator- und Elementtrennzeichen als Text ab.

#### Fehler

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### Parent

**Eigenschaft:** `Parent` als `AppOutputLines` (schreibgeschützt)

#### Beschreibung

Das Parent-Objekt gemäß dem Objektmodell.

**Fehler**

- 4100 Das Objekt ist nicht mehr gültig.
- 4101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## AppOutputLines

Stellt eine Sammlung von `AppOutputLine` Meldungszeilen dar.

### Eigenschaften und Methoden

Eigenschaften zum Navigieren im Objektmodell:

[Application](#)

[Parent](#)

Iterieren durch die Sammlung:

[Count](#)

[Item](#)

Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

### Beschreibung

Ruft das Objekt der obersten Ebene von application auf.

### Fehler

- 4000 Das Objekt ist nicht mehr gültig.
- 4001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Count

**Eigenschaft:** `Count` als `Integer` (schreibgeschützt)

### Beschreibung

Ruft die Anzahl der Zeilen in der Sammlung ab.

### Fehler

- 4000 Das Objekt ist nicht mehr gültig.
- 4001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Item

**Eigenschaft:** `Item` (`nIndex` als `Integer`) als [AppOutputLine](#) (schreibgeschützt)

### Beschreibung

Ruft die Zeile am Index `nIndex` aus der Sammlung auf. Die Indizes beginnen mit 1.

### Fehler

- 4000 Das Objekt ist nicht mehr gültig.
- 4001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

**Eigenschaft:** `Parent` als [AppOutputLine](#) (schreibgeschützt)

### Beschreibung

Das Parent-Objekt gemäß dem Objektmodell.

**Fehler**

- 4000 Das Objekt ist nicht mehr gültig.
- 4001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## AppOutputLineSymbol

Ein `AppOutputLineSymbol` stellt einen Link einer `AppOutputLine` Meldungszeile dar, auf den Sie im StyleVision-Fenster "Meldungen" klicken können.

Es wird auf eine Zelle einer `AppOutputLine` oder auf die gesamte Zeile selbst angewendet.

### Eigenschaften und Methoden

Eigenschaften zum Navigieren im Objektmodell:

[Application](#)

[Parent](#)

Aufruf der `AppOutputLineSymbol` Methoden:

[GetSymbolHREF](#)

[GetSymbolID](#)

[IsSymbolHREF](#)

Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

### Beschreibung

Ruft das Objekt der obersten Ebene von application auf.

### Fehler

- 4200 Das Objekt ist nicht mehr gültig.
- 4201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetSymbolHREF

**Methode:** `GetSymbolHREF ()` als `String`

### Beschreibung

Wenn das Symbol den Typ URL hat, wird die URL als String zurückgegeben.

### Fehler

- 4200 Das Objekt ist nicht mehr gültig.
- 4201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

.

GetSymbolID

**Methode:** `GetSymbolID ()` als `Long`

### Beschreibung

Ruft die ID des Symbols ab.

### Fehler

- 4200 Das Objekt ist nicht mehr gültig.
- 4201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsSymbolHREF

**Methode:** `IsSymbolHREF ()` als `Boolean`

**Beschreibung**

Gibt an, ob das Symbol von der Art URL ist.

**Fehler**

- 4200 Das Objekt ist nicht mehr gültig.
- 4201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

**Eigenschaft:** `Parent` als `Applet` (schreibgeschützt)

**Beschreibung**

Das Parent-Objekt gemäß dem Objektmodell.

**Fehler**

- 4200 Das Objekt ist nicht mehr gültig.
- 4201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## AuthenticContextMenu

Über die Kontextmenü-Schnittstelle können die in Authentic angezeigten Kontextmenüs angepasst werden. Die Schnittstelle hat die in diesem Abschnitt aufgelisteten Methoden.

CountItems

**Methode:** `CountItems()` nItems als long

### Rückgabewert

Gibt die Anzahl der Menüoptionen zurück.

### Fehler

2501 Ungültiges Objekt.

DeleteItem

**Methode:** `DeleteItem(IndexPosition als long)`

### Rückgabewert

Löscht den Menüeintrag mit der im ersten Parameter übergebenen Indexposition.

### Fehler

2501 Ungültiges Objekt

2502 Ungültiger Index

GetItemText

**Methode:** `GetItemText(IndexPosition als long)` Menübefehlsname als String

### Rückgabewert

Ruft den Namen des Menübefehls mit der im ersten Parameter übergebenen Indexposition ab.

### Fehler

2501 Ungültiges Objekt

2502 Ungültiger Index

InsertItem

**Methode:** `InsertItem(IndexPosition als long, MenuItemName als string, MacroName als string)`

### Rückgabewert

Fügt an der durch den ersten Parameter definierten Stelle im Menü einen benutzerdefinierten Menübefehl mit dem im zweiten Parameter definierten Namen ein. Der Menübefehl startet ein Makro, daher muss ein gültiger Makroname angegeben werden.

### Errors

2501 Ungültiges Objekt

2502 Ungültiger Index

2503 Kein Makro dieses Namens vorhanden

2504 Interner Fehler

SetItemText

**Methode:** `SetItemText(IndexPosition als long, MenuItemName als string)`

**Rückgabewert**

Definiert den Namen des Menübefehls, der sich an der im ersten Parameter übergebenen Indexposition befindet.

**Fehler**

- 2501 Ungültiges Objekt
- 2502 Ungültiger Index



## AuthenticEventContext

Über die `EventContext` Schnittstellen haben Sie Zugriff auf viele Eigenschaften des Kontexts, in dem ein Makro ausgeführt wird.

### EvaluateXPath

**Methode:** `EvaluateXPath(strExpression als string)` als `strValue` als `string`

### Rückgabewert

Diese Methode wertet den XPath-Ausdruck im Kontext des Node aus, in dem das Event ausgelöst wurde, und gibt einen String zurück.

### Beschreibung

`EvaluateXPath()` führt einen XPath-Ausdruck im angegebenen Event-Kontext aus. Das Ergebnis wird als String zurückgegeben. Im Fall einer Sequenz ist der String durch Leerzeichen getrennt.

### Fehler

- 2201 Ungültiges Objekt.
- 2202 Kein Kontext.
- 2209 Ungültiger Parameter.
- 2210 Interner Fehler.
- 2211 XPath-Fehler.

### GetEventContextType

**Methode:** `GetEventContextType()` `Type` als `AuthenticEventContextType` Enumeration

### Rückgabewert

Gibt den Kontext-Node-Typ zurück.

### Beschreibung

Mit Hilfe von `GetEventContextType` kann der Benutzer feststellen, ob sich das Makro in einem XML-Node oder in einem atomaren XPath Element-Kontext befindet. Die Enumeration `AuthenticEventContextType` ist folgendermaßen definiert:

```
authenticEventContextXML,
authenticEventContextAtomicItem,
authenticEventContextOther
```

Wenn der Kontext ein normaler XML-Node ist, erhalten Sie mit der Funktion `GetXMLNode()` Zugriff darauf (falls nicht, wird `NULL` zurückgegeben).

### Fehler

- 2201 Ungültiges Objekt.
- 2202 Kein Kontext.
- 2209 Ungültiger Parameter.

### GetNormalizedTextValue

**Methode:** `GetNormalizedTextValue()` `strValue` als `String`

### Rückgabewert

Gibt den Wert des aktuellen Node als String zurück.

### Fehler

- 2201 Ungültiges Objekt.

2202 Kein Kontext.  
 2203 Ungültiger Kontext  
 2209 Ungültiger Parameter.

#### GetVariableValue

**Methode:** `GetVariableValue`(strName als string) strValue als string

#### Rückgabewert

Ruft den Wert der als Parameter übergebenen Variablen ab.

#### Beschreibung

`GetVariableValue` ruft den Wert der Variablen im Geltungsbereich des Kontexts ab.

```
nZoom = parseInt(AuthenticView.EventContext.GetVariableValue('Zoom'));
if (nZoom > 1)
{
 AuthenticView.EventContext.SetVariableValue('Zoom', nZoom - 1);
}
```

#### Fehler

2201 Ungültiges Objekt.  
 2202 Kein Kontext.  
 2204 Keine Variable dieses Namens im Geltungsbereich  
 2205 Die Variable kann nicht ausgewertet werden  
 2206 Die Variable gibt eine Sequenz zurück  
 2209 Ungültiger Parameter

#### GetXMLNode

**Methode:** `GetXMLNode`() Node als XMLData Objekt

#### Rückgabewert

Gibt den XML-Kontext-Node oder `NULL` zurück

#### Fehler

2201 Ungültiges Objekt.  
 2202 Kein Kontext.  
 2203 Ungültiger Kontext  
 2209 Ungültiger Parameter.

#### IsAvailable

**Methode:** `IsAvailable`() als Boolean

#### Rückgabewert

Gibt "true" zurück, wenn `EventContext` definiert ist, andernfalls "false".

#### Fehler

2201 Ungültiges Objekt.

#### SetVariableValue

**Methode:** `SetVariableValue`(strName als string, strValue als string)

**Rückgabewert**

Definiert den Wert (den zweiten Parameter) der im ersten Parameter übergebenen Variablen.

**Beschreibung**

`SetVariableValue` definiert den Wert der Variablen im Geltungsbereich des Kontexts.

```
nZoom = parseInt(AuthenticView.EventContext.GetVariableValue('Zoom'));
if (nZoom > 1)
{
 AuthenticView.EventContext.SetVariableValue('Zoom', nZoom - 1);
}
```

**Fehler**

- 2201 Ungültiges Objekt.
- 2202 Kein Kontext.
- 2204 Keine Variable dieses Namens im Geltungsbereich
- 2205 Die Variable kann nicht ausgewertet werden
- 2206 Die Variable gibt eine Sequenz zurück
- 2207 Die Variable ist schreibgeschützt
- 2208 Keine Änderung zulässig

## AuthenticRange

### Siehe auch

Die erste Tabelle enthält eine Liste der Eigenschaften und Methoden von `AuthenticRange`, die zum Navigieren durch das Dokument und zum Auswählen bestimmter Abschnitte verwendet werden.

#### Eigenschaften

[Apply](#)  
[FirstPosition](#)  
[FirstXMLData](#)  
[FirstXMLDataOffset](#)  
[LastPosition](#)  
[LastXMLData](#)  
[LastXMLDataOffset](#)  
[Parent](#)

[Collapse](#)  
[CollapseToBegin](#)  
[CollapseToEnd](#)  
[ExpandTo](#)  
[Goto](#)  
[GotoNext](#)  
[GotoPrevious](#)  
[IsEmpty](#)  
[IsEqual](#)

#### Methoden

[MoveBegin](#)  
[MoveEnd](#)  
[NextCursorPosition](#)  
[PreviousCursorPosition](#)  
[Select](#)  
[SelectNext](#)  
[SelectPrevious](#)  
[SetFromRange](#)

Die folgende Tabelle enthält eine Liste der Methoden zur Bearbeitung von Inhalt. Die meisten dieser Methoden können über Kontextmenüs aufgerufen werden.

#### Eigenschaften

[Text](#)

#### Edit operations

[Copy](#)  
[Cut](#)  
[Delete](#)  
[IsCopyEnabled](#)  
[IsCutEnabled](#)  
[IsDeleteEnabled](#)  
[IsPasteEnabled](#)  
[Paste](#)

#### Dynamic table operations

[AppendRow](#)  
[DeleteRow](#)  
[DuplicateRow](#)  
[InsertRow](#)  
[IsFirstRow](#)  
[IsIndynamicTable](#)  
[IsLastRow](#)  
[MoveRowDown](#)  
[MoveRowUp](#)

Die folgenden Methoden liefern die Funktionen der Authentic-Eingabehilfenfenster für Bereichsobjekte.

### Operations of the entry helper windows

#### Elements

[CanPerformActionWith](#)  
[CanPerformAction](#)  
[PerformAction](#)

#### Attributes

[GetElementAttributeValue](#)  
[GetElementAttributeNames](#)  
[GetElementHierarchy](#)  
[HasElementAttribute](#)  
[IsTextStateApplied](#)  
[SetElementAttributeValue](#)

#### Entities

[GetEntityNames](#)  
[InsertEntity](#)

## Beschreibung

`AuthenticRange` Objekte sind die Cursor-Auswahl der Automation-Schnittstelle. Sie können damit auf jede Cursorposition in der Authentic-Ansicht zeigen oder einen Dokumentabschnitt auswählen. Die für `AuthenticRange` Objekte verfügbaren Operationen funktionieren anschließend an dieser Auswahl auf dieselbe Art wie die entsprechenden Operationen der Benutzeroberfläche an der aktuellen Auswahl des Benutzers. Der wichtigste Unterschied besteht darin, dass Sie beliebig viele `AuthenticRange` Objekte gleichzeitig verwenden können, während es auf der Benutzeroberfläche immer nur genau eine Cursorauswahl gibt.

Um zu Anfang ein Bereichsobjekt abzurufen, verwenden Sie [AuthenticView.Selection](#).

um einen Bereich zu erhalten, der der aktuellen Cursorauswahl auf der Benutzeroberfläche entspricht. Alternativ dazu stehen einige einfache Bereiche über die schreibgeschützten Eigenschaften [AuthenticView.DocumentBegin](#), [AuthenticView.DocumentEnd](#) und [AuthenticView.WholeDocument](#) zur Verfügung. Die flexibelste Methode ist [AuthenticView.Goto](#). Sie können damit in einem Aufruf zu einem bestimmten Bereich des Dokuments navigieren. Um eine komplexere Auswahl zu treffen, kombinieren Sie die obigen Methoden mit den verschiedenen Navigationsmethoden zu den in der ersten Tabelle auf dieser Seite aufgelisteten Bereichsobjekten.

Eine weitere Methode, einen Dokumentabschnitt auszuwählen, ist die Positioneigenschaften des Bereichsobjekts zu verwenden. Es stehen zwei Systeme zur Positionierung zur Verfügung, die beliebig miteinander kombiniert werden können:

- Für den Beginn und das Ende eines Bereichs können **absolute** Textcursorpositionen, beginnend mit der Position 0 am Dokumentanfang, gesetzt und abgerufen werden. Nähere Informationen dazu finden Sie unter [FirstTextPosition](#) und [LastTextPosition](#). Diese Methode erfordert komplexe interne Berechnungen und sollte nur mit Vorsicht eingesetzt werden.
- Für den Beginn und das Ende eines Bereichs können das **XMLData**-Element und eine Textposition innerhalb dieses Elements gesetzt und abgerufen werden. Nähere Informationen dazu finden Sie unter [FirstXMLData](#), [FirstXMLDataOffset](#), [LastXMLData](#) und [LastXMLDataOffset](#). Diese Methode ist äußerst effizient, erfordert aber Kenntnisse über die zugrunde liegende Dokumentstruktur. Sie kann zum Suchen von XMLData-Objekten und zum Durchführen von Operationen, die über die Benutzeroberfläche nicht aufgerufen werden können, verwendet werden.

Änderungen am Dokumentinhalt können durch verschiedene Methoden vorgenommen werden:

- Mit Hilfe der [Text](#)-Eigenschaft können Sie den durch das Bereichsobjekt ausgewählten Dokumenttext abrufen. Wenn die Eigenschaft gesetzt ist, wird der ausgewählte Dokumenttext durch den neuen Text ersetzt.
- Die Standardbearbeitungsfunktionen [Cut](#), [Copy](#), [Paste](#) und [Delete](#).
- Tabellenoperationen für Tabellen, die dynamisch wachsen können.
- Methoden, die den Funktionalitäten der Authentic-Eingabehilfefenster zugeordnet sind.
- Aufrufe der [XMLData](#) Objekte des zugrunde liegenden Dokuments, um diese direkt zu bearbeiten.

AppendRow

**Siehe auch**

**Methode:** [AppendRow\(\)](#) als Boolean

### Beschreibung

Wenn sich der Anfang eines Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine neue Zeile am Ende der ausgewählten Tabelle hinzu. Die Bereichsauswahl wird geändert, so dass sie auf den Beginn der neuen Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Anhängenoperation erfolgreich war. Anderfalls wird *false* zurückgegeben.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
```

```

' Append row at end of current dynamically growable table
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
 objRange.AppendRow
 ' objRange points to beginning of new row
 objRange.Select
End If

```

Application

### Siehe auch

**Eigenschaft:** [Application](#) als [Application](#) (schreibgeschützt)

### Beschreibung

Ruft das StyleVision application-Objekt auf.

### Fehler

- 2001 The authentic range object or its related view object is no longer valid.  
Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CanPerformAction

### Siehe auch

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

### Beschreibung

[CanPerformAction](#) und die damit in Zusammenhang stehenden Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die mühelose und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung durchgeführt wird. Der Beginn des Bereichsobjekts dient dazu, die nächste gültige Stelle zu finden, an der die angegebene Aktion durchgeführt werden kann. Sobald diese Stelle gefunden wurde, gibt die Methode *True* zurück, andernfalls *False*.

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

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

### Beispiele

Siehe [PerformAction](#).

CanPerformActionWith

### Siehe auch

**Methode:** [CanPerformActionWith](#) ([eAction](#) als SPYAuthenticActions,

`out_arrElementNames` als Variant)

### Beschreibung

`PerformActionWith` und die damit in Zusammenhang stehenden Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die mühelose und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung durchgeführt wird.

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

Tipp: Verwenden Sie zur Anwendung der Aktion [CanPerformActionWith](#).

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

### Beispiele

Siehe [PerformAction](#).

Clone

### Siehe auch

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

### Beschreibung

Gibt eine Kopie des Bereichsobjekts zurück.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CollapseToBegin

### Siehe auch

**Methode:** `CollapseToBegin()` als [AuthenticRange](#)

### Beschreibung

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

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CollapseToEnd

### Siehe auch

**Methode:** `CollapseToEnd()` als [AuthenticRange](#)

**Beschreibung**

Setzt den Beginn des Bereichsobjekts auf sein Ende. Die Methode gibt das geänderte Bereichsobjekt zurück.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Copy

**Siehe auch**

**Methode:** `Copy()` als Boolean

**Beschreibung**

Gibt *False* zurück, wenn der Bereich keine Abschnitte des Dokuments enthält, die kopiert werden können.

Gibt *True* zurück, wenn Text - und im Fall vollständig ausgewählter XML-Elemente auch die Elemente - in den Kopieren/Einfügen-Puffer kopiert wurde.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Cut

**Siehe auch**

**Methode:** `Cut()` als Boolean

**Beschreibung**

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

Gibt *True* zurück, nachdem Text - und im Fall von vollständig ausgewählten XML-Elementen auch die Elemente aus dem Dokument gelöscht und im Kopieren/Einfügen-Puffer gespeichert wurden.-

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Delete

**Siehe auch**

**Methode:** `Delete()` als Boolean

**Beschreibung**

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

Gibt *True* zurück, nachdem Text und im Fall von vollständig ausgewählten XML-Elementen auch die Elemente aus dem Dokument gelöscht wurden.

**Fehler**



- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DeleteRow

**Siehe auch**

**Methode:** DeleteRow() als Boolean

### Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, löscht diese Methode die ausgewählte Zeile. Die Bereichsauswahl wird geändert, so dass sie auf das nächste Element nach der gelöschten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Operation erfolgreich war, andernfalls ist der Rückgabewert *false*.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Delete selected row from dynamically growing table
' -----

Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we are in a table
If objRange.IsInDynamicTable Then
 objRange.DeleteRow
End If
```

DuplicateRow

**Siehe auch**

**Methode:** DuplicateRow() als Boolean

### Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine Kopie der aktuellen Zeile nach der ausgewählten ein. Die Bereichsauswahl wird geändert, so dass sie auf den Anfang der nächsten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Operation erfolgreich war, andernfalls ist der Rückgabewert *false*.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' duplicate row in current dynamically growable table
' -----

Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection
```

```

' check if we can insert something
If objRange.IsInDynamicTable Then
 objRange.DuplicateRow
 ' objRange points to beginning of new row
 objRange.Select
End If

```

## EvaluateXPath

**Methode:** [EvaluateXPath](#) (strExpression als string) strValue als string

### Rückgabewert

Die Methode gibt einen String zurück

### Beschreibung

[EvaluateXPath\(\)](#) führt einen XPath-Ausdruck aus, wobei der Kontext-Node der Anfang der Bereichsauswahl ist. Das als String zurückgegebene Ergebnis ist im Fall einer Sequenz ein durch Leerzeichen getrennter String. Wenn der XML-Node nicht relevant ist, kann der Benutzer jeden Node angeben, wie z.B. `AuthenticView.XMLDataRoot`.

### Fehler

- 2001 Ungültiges Objekt
- 2005 Ungültiger Parameter
- 2008 Interner Fehler
- 2202 Fehlender Kontext-Node
- 2211 XPath-Fehler

## ExpandTo

### Siehe auch

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

### Beschreibung

Wählt das gesamte Element vom Typ [eKind](#) aus, das an der ersten Cursorposition des Bereichs beginnt oder diese enthält. Die Methode gibt das geänderte Bereichsobjekt zurück.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2003 Die Erweiterung des Bereichs würde über das Ende des Dokuments hinausgehen.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## FirstTextPosition

### Siehe auch

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

### Beschreibung

Setzt oder ruft den am weitesten links gelegenen Positionsindex des Bereichsobjekts ab. Dieser Index ist immer kleiner oder gleich der [LastTextPosition](#). Die Indizierung beginnt am Dokumentanfang mit 0 und steigt dann mit jeder neuen Position, die der Textcursor einnehmen kann, schrittweise an. Die schrittweise Erhöhung der Textposition um 1 hat denselben Effekt, wie die nach rechts-Taste. Die schrittweise Verringerung der Textposition um 1 hat denselben Effekt, wie die nach links-Taste.

Wenn Sie `FirstTextPosition` auf einen höheren Wert als den der aktuellen `LastTextPosition` setzen, wird `LastTextPosition` auf den neuen Wert von `FirstTextPosition` gesetzt.

Tipp: Seien Sie beim Setzen der Textcursorposition vorsichtig, da dies im Vergleich zur Cursorpositionierung mittels `XMLData` eine kostspielige Operation ist.

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2006 Es wurde eine Textposition außerhalb des Dokuments definiert.

#### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
 MsgBox "Test using direct text cursor positioning was ok"
Else
 MsgBox "Oops! "
End If
```

FirstXMLData

#### Siehe auch

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

#### Beschreibung

Setzt oder ruft das erste `XMLData` Element im zugrunde liegenden Dokument ab, das durch die Bereichsauswahl teilweise oder zur Gänze ausgewählt ist. Der genaue Anfang der Auswahl wird durch das `FirstXMLDataOffset` Attribut definiert.

Immer wenn Sie `FirstXMLData` auf ein neues Datenobjekt setzen, wird `FirstXMLDataOffset` auf die erste Cursorposition innerhalb dieses Elements gesetzt. Nur `XMLData`-Objekte, die eine Cursorposition haben, können verwendet werden. Wenn Sie `FirstXMLData` / `FirstXMLDataOffset` auf eine Position setzen, die größer ist als der aktuelle Wert von `LastXMLData` / `LastXMLDataOffset` wird zweiteres auf die neue Anfangsposition verschoben.

Tipp: Sie können das zugrunde liegende XML-Dokument in Fällen, in denen die mit dem `AuthenticRange`-Objekt verfügbaren Methoden nicht ausreichen, mit Hilfe der Eigenschaften `FirstXMLData` und `LastXMLData` direkt aufrufen und bearbeiten

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2008 Interner Fehler
- 2009 Das XMLData-Objekt kann nicht aufgerufen werden.

**Beispiele**

```

' -----
' XMLSpy scripting environment - VBScript
' show name of currently selected XMLData element
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objXMLData
Set objXMLData = objAuthenticView.Selection.FirstXMLData
' authentic view adds a 'text' child element to elements
' of the document which have content. So we have to go one
' element up.
Set objXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name

```

FirstXMLDataOffset

**Siehe auch****Eigenschaft:** [FirstXMLDataOffset](#) as Long**Beschreibung**

Setzt oder ruft den Versatz der Cursorposition innerhalb des Elements [FirstXMLData](#) für den Anfang des Bereichs ab. Die Versatzpositionen basieren auf den von der Eigenschaft [Text](#) zurückgegebenen Zeichen und beginnen mit 0. Verwenden Sie beim Definieren eines neuen Versatzes -1, um den Versatz an die letzte mögliche Position im Element zu setzen. Beachten Sie bitte besonders die folgenden Fälle:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen kann anders aussehen, als angezeigt. Zwar basiert der Datenversatz auf diesem Text, doch gibt es nur zwei gültige Positionen für den Versatz, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Versatz irgendwo mitten im Eintrag zu positionieren, hat dies zur Folge, dass der Versatz an das Ende des Eintrags gesetzt wird.
- Die Textform von XML-Entities kann sich in ihrer Länge von der Darstellung auf dem Bildschirm unterscheiden. Der Versatz basiert auf der Textform.

Wenn [FirstXMLData](#) / [FirstXMLDataOffset](#) eine Position nach dem aktuellen [LastXMLData](#) / [LastXMLDataOffset](#) wählt, wird zweiteres an die neue Anfangsposition verschoben.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Es wurde ein ungültiger Versatz definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Beispiele**

```

' -----

```

```

' XMLSpy scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

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

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

' were we successful?
If objRange.IsEqual(objRange2) Then
 objRange.Select()
Else
 MsgBox "Oops"
End If

```

## GetElementAttributeNames

### Siehe auch

**Methode:** [GetElementAttributeNames](#) (*strElementName* als String,  
*out\_arrAttributeNames* als Variant)

### Beschreibung

Ruft die Namen aller Attribute für das einschließende Element mit dem angegebenen Namen auf. Verwenden Sie die Element/Attribut-Paare, um den Attributwert mit den Methoden [GetElementAttributeValue](#) und [SetElementAttributeValue](#) zu setzen oder abzurufen.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Invalid element name was specified.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

Siehe [SetElementAttributeValue](#).

## GetElementAttributeValue

### Siehe auch

**Methode:** [GetElementAttributeValue](#) (*strElementName* als String,  
*strAttributeName* als String) als String

### Beschreibung

Ruft für das in *strElementName* angegebene Element den Wert des in *strAttributeName* definierten Attributs ab. Wenn das Attribut unterstützt wird, aber keinen Wert zugewiesen hat, wird der leere String zurückgegeben. Um die Namen der von einem Element unterstützten

Attribute herauszufinden, verwenden Sie [GetElementAttributeNames](#), oder [HasElementAttribute](#).

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.  
Es wurde ein ungültiger Attributname angegeben.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### Beispiele

Siehe [SetElementAttributeValue](#).

GetElementHierarchy

#### Siehe auch

**Methode:** [GetElementHierarchy](#) (*out \_arrElementNames* als Variant)

#### Beschreibung

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

Die Namen der Elementhierarchie identifizieren zusammen mit dem Bereichsobjekt XMLData Elemente im Dokument. Die Attribute dieser Elemente können direkt von [GetElementAttributeNames](#) und damit in Zusammenhang stehenden Methoden aufgerufen werden.

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### C#-Beispiele

```

' -----
' C#
' -----

namespace ConsoleApplication1
{
 class Program
 {
 static void Main(string[] args)
 {
 XMLSpyLib.Application app = new XMLSpyLib.Application();

 app.ShowApplication(true);

 XMLSpyLib.AuthenticView view = app.ActiveDocument.AuthenticView;
 XMLSpyLib.AuthenticRange range = view.DocumentBegin;

 object o = null;
 range.GetElementHierarchy(ref o);

 object[] elements = (object[])o;
 }
 }
}

```

```

 foreach (string e in elements)
 {
 Console.WriteLine(e);
 }
 }
}

```

Also see: [SetElementAttributeValue](#).

## GetEntityNames

### Siehe auch

**Methode:** [GetEntityNames](#) (*out\_arrEntityNames* als Variant)

### Beschreibung

Ruft die Namen alle definierten Entities ab. Die Liste der abgerufenen Entities hängt nicht von der aktuellen Auswahl oder der aktuellen Position ab. Verwenden Sie einen dieser Namen mit der Funktion [InsertEntity](#).

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

Siehe [GetElementHierarchy](#) und [InsertEntity](#).

## GetVariableValue

**Methode:** [GetVariableValue](#)(*strName* als string) *strVal* als string

### Rückgabewert

Ruft den Wert der als Parameter der Methode benannten Variablen ab.

### Fehler

- 2001 Ungültiges Objekt.
- 2202 Kein Kontext.
- 2204 Es befindet sich keine Variable dieses Namens im Geltungsbereich
- 2205 Die Variable kann nicht ausgewertet werden
- 2206 Die Variable gibt eine Sequenz zurück
- 2209 Ungültiger Parameter

## Goto

### Siehe auch

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

### Beschreibung

Definiert den Bereich, der auf den Anfang des *nCount* Elements vom Typ *eKind* zeigt. Die Anfangsposition wird durch den Parameter *eFrom* definiert.

Mit Hilfe positiver Werte für *nCount* können Sie zum Ende des Dokuments navigieren. Mit Hilfe negativer Werte können Sie zum Beginn des Dokuments navigieren. die Methode gibt das geänderte Bereichsobjekt zurück.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Es wurde eine ungültige Anfangsposition definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GotoNext

**Siehe auch**

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

**Beschreibung**

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

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Beispiele**

```

' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document word-by-word
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
 objRange.GotoNext(spyAuthenticWord).Select
 If ((Err.number - vbObjectError) = 2003) Then
 bEndOfDocument = True
 Err.Clear
 ElseIf (Err.number <> 0) Then
 Err.Raise ' forward error
 End If
Wend

```

GotoNextCursorPosition

**Siehe auch**

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

**Beschreibung**

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



**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GotoPrevious

**Siehe auch**

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

**Beschreibung**

Setzt den Bereich auf den Beginn des Elements vom Typ `eKind`, welches sich vor dem Beginn des aktuellen Bereichs befindet. Die Methode geibt das geänderte Bereichsobjekt zurück.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Beispiele**

```

' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
 objRange.GotoPrevious(spyAuthenticTag).Select
 If ((Err.number - vbObjectError) = 2004) Then
 bBeginOfDocument = True
 Err.Clear
 ElseIf (Err.number <> 0) Then
 Err.Raise ' forward error
 End If
Wend

```

GotoPreviousCursorPosition

**Siehe auch**

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

**Beschreibung**

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

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

HasElementAttribute

#### Siehe auch

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

#### Beschreibung

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

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

InsertEntity

#### Siehe auch

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

#### Beschreibung

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

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Es wurde ein unbekannter Name für den Eintrag angegeben.

#### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Insert the first entity in the list of available entities
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we get the names of all available entities as they
' are shown in the entry helper of XMLSpy
Dim arrEntities
objRange.GetEntityNames arrEntities

' we insert the first one of the list
If UBound(arrEntities) >= 0 Then
 objRange.InsertEntity arrEntities(0)
Else
 MsgBox "Sorry, no entities are available for this document"
End If
```

InsertRow

**Siehe auch**

**Methode:** `InsertRow()` als Boolean

**Beschreibung**

Wenn sich der Beginn des Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine neue Zeile vor der aktuellen ein. Der Auswahlbereich wird geändert, sodass er auf den Beginn der neu eingefügten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Einfügeoperation erfolgreich war, andernfalls ist das Ergebnis *false*.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Beispiele**

```
' -----
' XMLSpy scripting environment - VBScript
' Insert row at beginning of current dynamically growing table
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
 objRange.InsertRow
 ' objRange points to beginning of new row
 objRange.Select
End If
```

IsCopyEnabled

**Siehe auch**

**Eigenschaft:** `IsCopyEnabled` als Boolean (schreibgeschützt)

**Beschreibung**

Überprüft, ob die Kopieroperation für diesen Bereich unterstützt wird.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsCutEnabled

**Siehe auch**

**Eigenschaft:** `IsCutEnabled` als Boolean (schreibgeschützt)

**Beschreibung**

Überprüft, ob die Ausschneideoperation für diesen Bereich unterstützt wird.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsDeleteEnabled

**Siehe auch**

**Eigenschaft:** [IsDeleteEnabled](#) als Boolean (schreibgeschützt)

**Beschreibung**

Überprüft, ob die Löschoperation für diesen Bereich unterstützt wird.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsEmpty

**Siehe auch**

**Methode:** [IsEmpty\(\)](#) als Boolean

**Beschreibung**

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

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsEqual

**Siehe auch**

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

**Beschreibung**

Überprüft, ob der Beginn und das Ende beider Bereiche identisch sind.

**Fehler**

- 2001 Eines der beiden verglichenen Bereichsobjekte ist ungültig.
- 2005 Es wurde eine ungültige Adresse für einen Rückgabeparameter angegeben.

IsFirstRow

**Siehe auch**

**Eigenschaft:** [IsFirstRow](#) als Boolean (schreibgeschützt)

**Beschreibung**

Überprüft, ob der Bereich in der ersten Zeile einer Tabelle ist. Welche Tabelle berücksichtigt wird, ist vom Ausmaß des Bereichs abhängig. Wenn die Auswahl über eine einzige Tabellenzeile hinausgeht, wird überprüft, ob diese Tabelle das erste Element in einer einbettenden Tabelle ist. Nähere Informationen dazu finden Sie im Benutzerhandbuch unter Eingabehilfen.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.

2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsInDynamicTable

**Siehe auch**

**Methode:** `IsInDynamicTable()` als Boolean

**Beschreibung**

Überprüft, ob sich der gesamte Bereich innerhalb einer Tabelle befindet, die die verschiedenen Zeilenoperationen wie Einfügen, Anhängen, Kopieren usw. unterstützt.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsLastRow

**Siehe auch**

**Eigenschaft:** `IsLastRow` als Boolean (schreibgeschützt)

**Beschreibung**

Überprüft, ob der Bereich in der letzten Zeile einer Tabelle ist. Welche Tabelle berücksichtigt wird, hängt vom Ausmaß des Bereichs ab. Wenn die Auswahl über eine einzige Tabellenzeile hinausgeht, wird überprüft, ob diese Tabelle das letzte Element in einer einbettenden Tabelle ist. Nähere Informationen dazu finden Sie im Benutzerhandbuch unter Eingabehilfen.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsPasteEnabled

**Siehe auch**

**Eigenschaft:** `IsPasteEnabled` als Boolean (schreibgeschützt)

**Beschreibung**

Überprüft, ob die Einfügeoperationen für diesen Bereich unterstützt wird.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsSelected

**Eigenschaft:** `IsSelected` als Boolean

**Beschreibung**

Gibt true() zurück, wenn die Auswahl vorhanden ist. Der Auswahlbereich kann dennoch leer sein. Dies passiert z.B. wenn der Cursor gesetzt ist.

IsTextStateApplied

**Siehe auch**

**Methode:** `IsTextStateApplied` (`i_strElementName` als String) als Boolean

### Beschreibung

Überprüft, ob der gesamte ausgewählte Text in ein XML-Element mit dem Namen `i_strElementName` eingebettet ist. Ein typisches Beispiel für den Parameter `i_strElementName` ist "strong", "bold" oder "italic".

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LastTextPosition

**Siehe auch**

**Eigenschaft:** `LastTextPosition` als Long

### Beschreibung

Setzt oder ruft den am weitesten rechts gelegenen Positionsindex des Bereichsobjekts ab. Dieser Index ist immer größer oder gleich `FirstTextPosition`. Die Indizierung beginnt am Dokumentanfang mit 0 und steigt dann mit jeder neuen Position, die der Textcursor einnehmen kann, schrittweise an. Die schrittweise Erhöhung der Textposition um 1 hat denselben Effekt, wie die nach rechts-Taste. Die schrittweise Verringerung der Textposition um 1 hat denselben Effekt, wie die nach links-Taste.

Wenn Sie `LastTextPosition` auf einen niedrigeren Wert als den der aktuellen `FirstTextPosition` setzen, wird `FirstTextPosition` auf den neuen Wert von `LastTextPosition` gesetzt.

Tipp: Seien Sie beim Setzen der Textcursorposition vorsichtig, da dies im Vergleich zur Cursorpositionierung mittels `XMLData` eine kostspielige Operation ist.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2006 Es wurde eine Textposition außerhalb des Dokuments definiert.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
```

```
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
 MsgBox "Test using direct text cursor positioning was ok"
Else
 MsgBox "Oops! "
End If
```

## LastXMLData

### Siehe auch

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

### Beschreibung

Setzt oder ruft das letzte [XMLData](#) Element im zugrunde liegenden Dokument ab, das durch die Bereichsauswahl teilweise oder zur Gänze ausgewählt ist. Das genaue Ende der Auswahl wird durch das [LastXMLDataOffset](#) Attribut definiert.

Immer wenn Sie [LastXMLData](#) auf ein neues Datenobjekt setzen, wird [LastXMLDataOffset](#) auf die letzte Cursorposition innerhalb dieses Elements gesetzt. Nur [XMLData](#)-Objekte, die eine Cursorposition haben, können verwendet werden. Wenn Sie [LastXMLData](#) / [LastXMLDataOffset](#) auf eine Position setzen, die niedriger ist als der aktuelle Wert von [FirstXMLData](#) / [FirstXMLDataOffset](#) wird zweiteres auf die neue Endposition verschoben.

Tipp: Sie können das zugrunde liegende XML-Dokument in Fällen, in denen die mit dem [AuthenticRange](#)-Objekt verfügbaren Methoden nicht ausreichen, mit Hilfe der Eigenschaften [FirstXMLData](#) und [LastXMLData](#) direkt aufrufen und bearbeiten

### Fehler

- |      |                                                                                                       |
|------|-------------------------------------------------------------------------------------------------------|
| 2001 | Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig. |
| 2005 | Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.                                     |
| 2008 | Interner Fehler                                                                                       |
| 2009 | Das <a href="#">XMLData</a> Objekt kann nicht aufgerufen werden.                                      |

## LastXMLDataOffset

### Siehe auch

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

### Beschreibung

Setzt oder ruft den Versatz der Cursorposition innerhalb des Elements [LastXMLData](#) für das Ende des Bereichs ab.

Die Versatzpositionen basieren auf den von der Eigenschaft [Text](#) zurückgegebenen Zeichen und beginnen mit 0. Verwenden Sie beim Definieren eines neuen Versatzes -1, um den Versatz an die letzte mögliche Position im Element zu setzen. Beachten Sie bitte besonders die folgenden Fälle:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen kann anders aussehen, als angezeigt. Zwar basiert der Datenversatz auf diesem Text, doch gibt es nur zwei gültige Positionen für den Versatz, eine am

Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Versatz irgendwo mitten im Eintrag zu positionieren, hat dies zur Folge, dass der Versatz an das Ende des Eintrags gesetzt wird.

- Die Textform von XML-Entities kann sich in ihrer Länge von der Darstellung auf dem Bildschirm unterscheiden. Der Versatz basiert auf der Textform.

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

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Es wurde ein ungültiger Versatz definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

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

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

' were we successful?
If objRange.IsEqual(objRange2) Then
 objRange.Select()
Else
 MsgBox "Ooops"
End If
```

### MoveBegin

### Siehe auch

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

### Beschreibung

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

Verwenden Sie positive Zahlen für *nCount*, um die Position zum Ende des Dokuments hin zu verschieben und negative Zahlen um die Position zum Anfang des Dokuments hin zu verschieben. Das Ende des Bereichs bleibt gleich, es sei denn, der neue Anfang hat einen höheren Wert als das Ende. In diesem Fall wird das Ende an den neuen Anfang verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.



**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveEnd

**Siehe auch**

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

**Beschreibung**

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

Verwenden Sie positive Zahlen für *nCount*, um die Position zum Ende des Dokuments hin zu verschieben und negative Zahlen um die Position zum Anfang des Dokuments hin zu verschieben. Der Anfang des Bereichs bleibt gleich, es sei denn, das neue Ende hat einen niedrigeren Wert als der Anfang. In diesem Fall wird der Anfang an das neue Ende verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveRowDown

**Siehe auch**

**Methode:** [MoveRowDown\(\)](#) als Boolean

**Beschreibung**

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile auswählt, die nicht die letzte Zeile in dieser Tabelle ist, tauscht diese Methode diese Zeile gegen die unmittelbar darunter gelegene Zeile aus. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber ansonsten nicht. Die Funktion gibt *true* zurück, wenn der Verschiebevorgang erfolgreich war, andernfalls wird *false* zurückgegeben.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveRowUp

**Siehe auch**

**Method:** [MoveRowUp\(\)](#) als Boolean

**Beschreibung**

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile auswählt, die nicht die erste Zeile in dieser Tabelle ist, tauscht diese Methode diese Zeile gegen die unmittelbar darüber gelegene Zeile aus. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber ansonsten nicht. Die Funktion gibt *true* zurück, wenn der Verschiebevorgang erfolgreich war, andernfalls wird *false* zurückgegeben

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

**Beispiele**

Siehe JScript - Bubble-Sortierung dynamischer Tabellen.

Parent

**Siehe auch**

**Eigenschaft:** `Parent` als [AuthenticView](#) (schreibgeschützt)

**Beschreibung**

Ruft die Ansicht auf, die dieses Bereichsobjekt enthält.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Paste

**Siehe auch**

**Methode:** `Paste()` als Boolean

**Beschreibung**

Gibt *False* zurück, wenn der Kopieren/Einfügen-Puffer leer ist oder sein Inhalt die aktuelle Auswahl nicht ersetzen kann.

Löscht andernfalls die aktuelle Auswahl, fügt den Inhalt des Kopieren/Einfügen-Puffers ein und gibt *True* zurück.

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

PerformAction

**Siehe auch**

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

**Beschreibung**

`PerformAction` und die dazugehörigen Methoden verschaffen Ihnen Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die einfach und einheitliche

Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung stattfinden soll. Der Anfang des Bereichsobjekts dient dazu, die nächste gültige Stelle zu finden, an der die angegebene Aktion durchgeführt werden kann. Wenn keine derartige Stelle gefunden wird, gibt die Methode *False* zurück. Andernfalls wird das Dokument geändert und der Bereich zeigt auf den Anfang der Änderung.

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

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

#### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Insert the innermost element
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' we determine the elements that can be inserted at the current position
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
If UBound(arrElements) >= 0 Then
 objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
 ' objRange now points to the beginning of the inserted element
 ' we set a default value and position at its end
 objRange.Text = "Hello"
 objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
 MsgBox "Can't insert any elements at current position"
End If
```

Select

#### Siehe auch

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

#### Beschreibung

Macht diesen Bereich zur aktuellen Auswahl der Benutzeroberfläche. Dasselbe Ergebnis können Sie mit: '*objRange.Parent.Selection = objRange*' erzielen.

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.

#### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView
```

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

SelectNext

**Siehe auch**

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

### Beschreibung

Wählt das Element vom Typ `eKind` nach dem aktuellen Ende des Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document word-by-word
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
 objRange.SelectNext(spyAuthenticWord).Select
 If ((Err.number - vbObjectError) = 2003) Then
 bEndOfDocument = True
 Err.Clear
 ElseIf (Err.number <> 0) Then
 Err.Raise ' forward error
 End If
Wend
```

SelectPrevious

**Siehe auch**

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

### Beschreibung

Wählt das Element vom Typ `eKind` vor dem aktuellen Anfang des Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments

- 2005 Es wurde eine ungültige Elementart definiert.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
 objRange.SelectPrevious(spyAuthenticTag).Select
 If ((Err.number - vbObjectError) = 2004) Then
 bBeginOfDocument = True
 Err.Clear
 ElseIf (Err.number <> 0) Then
 Err.Raise ' forward error
 End If
Wend

```

### SetElementAttributeValue

#### Siehe auch

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

#### Beschreibung

Ruft den Wert des in *strAttributeName* definierten Attributs für das mit *strElementName* identifizierte Element auf. Wenn das Attribut unterstützt wird, aber keinen Wert zugewiesen hat, wird der leere String zurückgegeben. Um die Namen der von einem Element unterstützten Attribute zu ermitteln, verwenden Sie [GetElementAttributeNames](#) oder [HasElementAttribute](#).

#### Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.  
2005 Es wurde ein ungültiger Elementname angegeben.  
Es wurde ein ungültiger Attributname definiert.  
Es wurde ein ungültiger Attributwert definiert.

### Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Get and set element attributes
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we find out all the elements below the beginning of the range
Dim arrElements
objRange.GetElementHierarchy arrElements

```

```

If IsArray(arrElements) Then
 If UBound(arrElements) >= 0 Then
 ' we use the top level element and find out its valid attributes
 Dim arrAttrs()
 objRange.GetElementAttributeNames arrElements(0), arrAttrs

 If UBound(arrAttrs) >= 0 Then
 ' we retrieve the current value of the first valid
attribute
 Dim strAttrVal
 strAttrVal = objRange.GetElementAttributeValue
(arrElements(0), arrAttrs(0))
 msgbox "current value of " & arrElements(0) & "/" &
arrAttrs(0) & " is: " & strAttrVal

 ' we change this value and read it again
 strAttrVal = "Hello"
 objRange.SetElementAttributeValue arrElements(0),
arrAttrs(0), strAttrVal
 strAttrVal = objRange.GetElementAttributeValue
(arrElements(0), arrAttrs(0))
 msgbox "new value of " & arrElements(0) & "/" &
arrAttrs(0) & " is: " & strAttrVal
 End If
 End If
End If

```

SetFromRange

**Siehe auch**

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

### Beschreibung

Setzt den Bereich des Objekts auf dieselbe Anfangs- und Endposition wie *objSrcRange* .

### Fehler

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

SetVariableValue

**Methode:** [SetVariableValue](#)(*strName* als string, *strValue* als string)

### Rückgabewert

Definiert den Wert (zweiten Parameter) der im ersten Parameter genannten Variablen.

### Fehler

- 2201 Ungültiges Objekt.
- 2202 Kein Kontext.
- 2204 Es befindet sich keine Variable dieses Namens im Geltungsbereich.
- 2205 Die Variable kann nicht ausgewertet werden.
- 2206 Die Variable gibt eine Sequenz zurück.
- 2207 Die Variable ist schreibgeschützt.
- 2208 Es ist keine Änderung zulässig.

Text

**Siehe auch**

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

**Beschreibung**

Setzt oder ruft den durch ein Bereichsobjekt ausgewählten Textinhalt ab.

Die Anzahl der abgerufenen Zeichen muss nicht unbedingt identisch sein, da es Textcursorpositionen zwischen dem Beginn und dem Ende des ausgewählten Bereichs gibt. Die meisten Dokumentelemente unterstützen eine Cursor-Endposition, die sich von der Cursor-Beginnposition des folgenden Elements unterscheidet. Dropdown-Listen haben nur eine Cursorposition, können jedoch Strings jeder Länge auswählen. Im Fall von Optionsfeldern und Kontrollkästchen enthält die Texteigenschaft den String des entsprechenden XML-Elements.

Wenn mit dem Bereich mehr als ein Element ausgewählt wird, ist der Text die Verkettung der Einzeltexte. XML Entities werden erweitert, sodass '&' als '&amp;' erwartet wird.

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

**Fehler**

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Es wurde eine ungültige Adresse für einen Rückgabeparameter angegeben.

## AuthenticView

### Siehe auch

#### Eigenschaften

[Application](#)  
[AsXMLString](#)  
[DocumentBegin](#)  
[DocumentEnd](#)  
[Event](#)  
[MakeVisible](#)  
[Parent](#)  
[Section](#)  
[XMLDataRoot](#)  
[WholeDocument](#)

#### Methoden

[Goto](#)  
[IsRedoEnabled](#)  
[IsUndoEnabled](#)  
[Print](#)  
[Redo](#)  
[Undo](#)  
[UpdateXMLInstanceEntities](#)

#### Events

[OnBeforeCopy](#)  
[OnBeforeCut](#)  
[OnBeforeDelete](#)  
[OnBeforeDrop](#)  
[OnBeforePaste](#)  
[OnDragOver](#)  
[OnKeyboardEvent](#)  
[OnMouseEvent](#)  
[OnSelectionChanged](#)

### Beschreibung

Über AuthenticView und seine untergeordneten Objekte [AuthenticRange](#) und [AuthenticDataTransfer](#) steht Ihnen eine Schnittstelle für die Authentic-Ansicht zur Verfügung, über die Sie Dokumentinhalt einfach und konsistent ändern können. Diese Schnittstellen treten nun anstelle der folgenden nun als **nicht mehr verwendet** markierten Schnittstellen:

OldAuthenticView (der alte Name war DocEditView)  
 AuthenticSelection (der alte Name war DocEditSelection, wird nun ersetzt durch [AuthenticRange](#))  
 AuthenticEvent (der alte Name war DocEditEvent) [Interfaces](#)

Mit AuthenticView erhalten Sie einfachen Zugriff auf spezielle Funktionen wie Drucken, den mehrstufigen Rückgängig-Puffer und die aktuelle Cursorauswahl oder - position.

AuthenticView verwendet zur einfachen Navigation im Dokument und zur flexiblen Auswahl logischer Textelemente Objekte vom Typ [AuthenticRange](#). Verwenden Sie für einfache Auswahldefinitionen die Eigenschaften [DocumentBegin](#), [DocumentEnd](#) oder [WholeDocument](#) und die [Goto](#) Methode für komplexere Auswahldefinitionen. Informationen zum Navigieren relativ zu einem bestimmten Dokumentbereich finden Sie unter den Methoden und Eigenschaften des [AuthenticRange](#) Objekts.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' -----

Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
 ' we have an active document, now check for view mode
 If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
 If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
 MsgBox "Active document does not support authentic view
mode"
 Else
 ' now it is safe to access the authentic view object
 Dim objAuthenticView
 Set objAuthenticView = objDocument.AuthenticView
 ' now use the authentic view object

 End If
 End If
End If
```



```
Else
 MsgBox "No document is open"
End If
```

## Events

### OnBeforeCopy

#### Siehe auch

**Event:** `OnBeforeCopy()` als Boolean

#### XMLSpy Skripting-Umgebung - VBScript:

```
Function On_AuthenticBeforeCopy()
 'On_AuthenticBeforeCopy=False 'to disable operation
EndFunction
```

#### XMLSpy Skripting-Umgebung - JScript:

```
function On_AuthenticBeforeCopy()
{
 //return false; //to disable operation
}
```

#### XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (21, ...) //nEventId=21
```

#### Beschreibung

Dieses Event wird ausgelöst, bevor eine Kopieroperation an einem Dokument durchgeführt wird. Bei Rückgabe von *True* (oder nichts) wird die Kopieroperation gestattet, bei Rückgabe von *False* wird das Kopieren deaktiviert.

### OnBeforeCut

#### Siehe auch

**Event:** `OnBeforeCut()` als Boolean

#### XMLSpy Skripting-Umgebung - VBScript:

```
Function On_AuthenticBeforeCut()
 'On_AuthenticBeforeCut=False 'to disable operation
EndFunction
```

#### XMLSpy Skripting-Umgebung - JScript:

```
function On_AuthenticBeforeCut()
{
 //return false; //to disable operation
}
```

#### XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (20, ...) //nEventId=20
```

#### Beschreibung

Dieses Event wird ausgelöst, bevor eine Ausschneideoperation an einem Dokument durchgeführt wird. Geben Sie *True* (oder nichts) zurück, um den Vorgang zu gestatten und *False* um den Vorgang zu deaktivieren.

OnBeforeDelete

**Siehe auch**

**Event:** `OnBeforeDelete()` als Boolean

**XMLSpy Skripting-Umgebung - VBScript:**

```
Function On_AuthenticBeforeDelete()
 'On_AuthenticBeforeDelete=False 'to disable operation
EndFunction
```

**XMLSpy Skripting-Umgebung - JScript:**

```
function On_AuthenticBeforeDelete()
{
 //return false; 'to disable operation;
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (22, ...) //nEventId=22
```

**Beschreibung**

Dieses Event wird ausgelöst, bevor eine Löschoperation an einem Dokument durchgeführt wird. Geben Sie *True* (oder nichts) zurück, um den Vorgang zu gestatten und *False* um den Vorgang zu deaktivieren.

OnBeforeDrop

**Siehe auch**

**Event:** `OnBeforeDrop ( i_nXmos als Long, i_nYmos als Long, i_ipRange als AuthenticRange, i_ipData als cancel Boolean`

**XMLSpy Skripting-Umgebung - VBScript:**

```
Function On_AuthenticBeforeDrop(nXmos, nYmos, objRange, objData)
 'On_AuthenticBeforeDrop=False 'to disable operation
EndFunction
```

**XMLSpy Skripting-Umgebung - JScript:**

```
function On_AuthenticBeforeDrop(nXmos, nYmos, objRange, objData)
{
 //return false; 'to disable operation;
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (11, ...) //nEventId=11
```

**Beschreibung**

Dieses Event wird ausgelöst, nachdem ein Objekt mit der Maus in das Applikationsfenster gezogen wurde. Alle mit dem Event in Zusammenhang stehenden Informationen werden als Parameter übergeben.

Die ersten beiden Parameter definieren die Mausposition zum Zeitpunkt des Auftretens des Event. Der Parameter *objRange* übergibt ein Bereichsobjekt, das das XML-Element unterhalb der Mausposition auswählt. Der Wert dieses Parameters kann *NULL* sein. Überprüfen Sie dies, bevor Sie das Bereichsobjekt aufrufen. Mit Hilfe des Parameters *objData* können Sie Informationen über das gezogene Objekt aufrufen.

Geben Sie *False* zurück, um den Drop-Vorgang abubrechen. Geben Sie *True* (oder nichts)

zurück, um mit einem normalen Vorgang fortzusetzen.

### Beispiele

```
'
'-----
' VB code snippet - connecting to object level events
'-----

' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnBeforeDrop
' event of object objView
Private Function objView_OnBeforeDrop(ByVal i_nXPos As Long, ByVal i_nYPos
As Long,
 ByVal i_ipRange As IAuthenticRange,
 ByVal i_ipData As
IAuthenticDataTransfer) As Boolean

 If (Not i_ipRange Is Nothing) Then
 MsgBox ("Dropping on content is prohibited");
 Return False;
 Else
 Return True;
 End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

### OnBeforePaste

#### Siehe auch

**Event:** `OnBeforePaste` (*objData* als Variant, *strType* als String) als Boolean

#### XMLSpy Skripting-Umgebung - VBScript:

```
Function On_AuthenticBeforePaste(objData, strType)
 'On AuthenticBeforePaste=False 'to disable operation
EndFunction
```

#### XMLSpy Skripting-Umgebung - JScript:

```
function On_AuthenticBeforePaste(objData, strType)
{
 //return false; //to disable operation
}
```

#### XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (19, ...) //nEventId=19
```

### Beschreibung

Dieses Event wird ausgelöst, bevor eine Einfügeoperation am Dokument ausgeführt wird. Der Parameter *strType* ist entweder "TEXT", "UNICODETEXT" oder "IUNKNOWN". In den ersten beiden Fällen enthält *objData* eine String-Darstellung des Objekts, das eingefügt wird. Im letzteren Fall enthält *objData* einen Pointer auf eine IUnknown COM-Schnittstelle.

Geben Sie *True* (oder nichts) zurück, um die Einfügeoperation zu gestatten. Geben Sie *False* zurück, um die Operation zu deaktivieren.

OnBeforeSave

**Event:** `OnBeforeSave` (SaveAs flag) als Boolean

**Beschreibung:** Mit `OnBeforeSave` haben Sie die Möglichkeit den Benutzer vor einem Überschreiben des bestehenden Dokuments zu warnen oder das Dokument mit einem Schreibschutz zu versehen, wenn bestimmte Bedingungen nicht erfüllt werden. Das Event wird ausgelöst, bevor das Dialogfeld "Datei" angezeigt wird. (Beachten Sie bitte, dass das Event beim Speichern des Dokuments und nicht beim Speichern des SPS-Designs in StyleVision ausgelöst wird.)

OnDragOver

**Siehe auch**

**Event:** `OnDragOver` (`nXmos` als Long, `nYmos` als Long, `eMouseEvent` als `SPYMouseEvent`, `objRange` als `AuthenticRange`, `objData` als `AuthenticDataTransfer`) als Boolean

#### XMLSpy Skripting-Umgebung - VBScript:

```
Function On_AuthenticDragOver(nXmos, nYmos, eMouseEvent, objRange,
 objData)
 'On_AuthenticDragOver=False 'to disable operation
EndFunction
```

#### XMLSpy Skripting-Umgebung - JScript:

```
function On_AuthenticDragOver(nXmos, nYmos, eMouseEvent, objRange, objData)
{
 //disable 'to disable operation'
}
}
```

#### XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (10, ...) //nEventId=10
```

#### Beschreibung

Dieses Event wird ausgelöst, wenn ein Objekt, von innerhalb oder außerhalb der Authentic-Ansicht mit der Maus über das Applikationsfenster gezogen wird. Alle mit dem Event in Zusammenhang stehenden Informationen werden als Parameter übergeben.

Die ersten drei Parameter definieren die Mausposition, den Status der Maustaste und den Status virtueller Schaltflächen zum Zeitpunkt, zu dem das Event auftritt. Der Parameter `objRange` übergibt ein Bereichsobjekt, das das XML-Element unterhalb der Mausposition auswählt. Der Wert dieses Parameters kann *NULL* sein. Überprüfen Sie dies, bevor Sie das Bereichsobjekt aufrufen. Mit Hilfe des Parameters `objData` können Sie Informationen über das gezogene Objekt aufrufen.

Geben Sie *False* zurück, um den Drag-Vorgang abubrechen. Geben Sie *True* (oder nichts) zurück, um mit einem normalen Vorgang fortzusetzen.

#### Beispiele

```
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
```

```

Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnDragOver
' event of object objView
Private Function objView_OnDragOver(ByVal i_nXPos As Long, ByVal i_nYPos As
Long,
 ByVal i_eMouseEvent As SPYMouseEvent,
 ByVal i_ipRange As IAuthenticRange,
 ByVal i_ipData As
IAuthenticDataTransfer) As Boolean

 If (((i_eMouseEvent And spyShiftKeyDownMask) <> 0) And
 (Not i_ipRange Is Nothing)) Then
 MsgBox ("Floating over element " &
i_ipRange.FirstXMLData.Parent.Name);
 End If

 Return True;
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop

```

## OnKeyboardEvent

### Siehe auch

**Event:** `OnKeyboardEvent` (`eKeyEvent` als `SPYKeyEvent`, `nKeyCode` als Long, `nVirtualKeyStatus` als Long) als Boolean

### XMLSpy Skripting-Umgebung - VBScript:

```

Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode,
nVirtualKeyStatus)
 'On_AuthenticKeyboardEvent=True 'to cancel bubbling of event
EndFunction

```

### XMLSpy Skripting-Umgebung - JScript:

```

function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
 //return true; //to cancel bubbling of event/
}

```

### XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (30, ...) //nEventId=30
```

### Beschreibung

Dieses Event wird für `WM_KEYDOWN`, `WM_KEYUP` und `WM_CHAR` Windows-Meldungen ausgelöst.

Der eigentliche Meldungstyp steht im `eKeyEvent` Parameter zur Verfügung. Der Status virtueller Tasten ist im Parameter `nVirtualKeyStatus` kombiniert. Verwenden Sie die im Enumerations-Datentyp `SPYVirtualKeyMask` definierten Bit-Masken, um eine Überprüfung der unterschiedlichen Tasten und ihrer Kombinationen durchzuführen.

## OnLoad

**Event:** OnLoad ()

**Beschreibung:** Mit OnLoad können einige Funktionalitäten der Authentic-Ansicht eingeschränkt werden, wie im Beispiel unten gezeigt:

```
function On_AuthenticLoad()
{
 // We are disabling all entry helpers in order to prevent user from
 // manipulating XML tree
 AuthenticView.DisableElementEntryHelper();
 AuthenticView.DisableAttributeEntryHelper();

 // We are also disabling the markup buttons for the same purpose
 AuthenticView.SetToolBarButtonState('AuthenticMarkupSmall',
 authenticToolBarButtonDisabled);
 AuthenticView.SetToolBarButtonState('AuthenticMarkupLarge',
 authenticToolBarButtonDisabled);
 AuthenticView.SetToolBarButtonState('AuthenticMarkupMixed',
 authenticToolBarButtonDisabled);
}
```

Im Beispiel wird der Status der Symbolleisten-Schaltflächen "Kleine Markup-Symbole", "Große Markup-Symbole" und "Gemischte Markup-Symbole" mit Hilfe der Schaltflächen-Identifizier bearbeitet. Siehe [vollständige Liste](#).

## OnMouseEvent

**Siehe auch**

**Event:** OnMouseEvent (*nXmos* als Long, *nYmos* als Long, *eMouseEvent* als SPYMouseEvent, *objRange* als [AuthenticRange](#)) als Boolean

**XMLSpy Skripting-Umgebung - VBScript:**

```
Function On_AuthenticMouseEvent(nXmos, nYmos, eMouseEvent, objRange)
 'On_AuthenticMouseEvent=True 'to cancel bubbling of event
EndFunction
```

**XMLSpy Skripting-Umgebung - JScript:**

```
function On_AuthenticMouseEvent(nXmos, nYmos, eMouseEvent, objRange)
{
 //return true 'to cancel bubbling of event
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (31, ...) //nEventId=31
```

**Beschreibung**

Dieses Event wird für jede Mausbewegung und jede Mausschaltflächen-Windows-Meldung ausgelöst.

Der eigentliche Meldungstyp und der Status der Maustaste stehen im *eMouseEvent* Parameter zur Verfügung. Verwenden Sie die im Enumerations-Datentyp *SPYMouseEvent* definierten Bit-Masken, um eine Überprüfung auf die unterschiedlichen Meldungen, Tastenstatus und ihre Kombinationen durchzuführen.

Der Parameter *objRange* definiert den Teil des Dokuments, der sich an der aktuellen Mauscursorposition befindet. Das Bereichsobjekt wählt immer einen vollständigen Tag des Dokuments aus. (Dies kann sich in zukünftigen Versionen ändern, wenn ein genauerer

Positionierungsmechanismus zur Verfügung steht). Wenn sich an der aktuellen Position kein auswählbarer Teil des Dokuments befindet, ist das Bereichsobjekt *Null*.

OnSelectionChanged

**Siehe auch**

**Event:** OnSelectionChanged (*objNewSelection* als [AuthenticRange](#))

**XMLSpy Skripting-Umgebung - VBScript:**

```
Function On_AuthenticSelectionChanged(objNewSelection)
EndFunction
```

**XMLSpy Skripting-Umgebung - JScript:**

```
function On_AuthenticSelectionChanged(objNewSelection)
{
}
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (23, ...) //nEventId=23
```

**Beschreibung**

Dieses Event wird ausgelöst, wenn die sich die Auswahl auf der Benutzeroberfläche ändert.

**Beispiele**

```
'
'-----
' VB code snippet - connecting to object level events
'
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnSelectionChanged
' event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As
XMLSpyLib.IAuthenticRange)
 MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

OnToolBarButtonClicked

**Event:** OnToolBarButtonClicked (Button identifier)

**Beschreibung:** OnToolBarButtonClicked wird ausgelöst, wenn der Benutzer auf eine Symbolleiste-Schaltfläche klickt. Über den Parameter Schaltflächen-Identifizier, kann ermittelt werden, auf welche Schaltfläche geklickt wurde. Unten sehen Sie die Liste der vordefinierten Schaltflächen-Identifizier:

- AuthenticPrint
- AuthenticPrintPreview

- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticClear
- AuthenticMarkupHide
- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticReload
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCellRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCellBelow
- AuthenticTableSplitCellHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCellVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable
- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown
- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

Der Benutzer kann für seine benutzerdefinierten Schaltflächen seine eigenen Identifier hinzufügen. Beachten Sie bitte, dass der Benutzer aufpassen muss, da die Identifier nicht auf Eindeutigkeit überprüft werden. Dieselben Identifier können zur Kennzeichnung von Schaltflächen in `Set/GetToolbarState()` COM API-Aufrufen verwendet werden. Durch Hinzufügen von Code für unterschiedliche Schaltflächen hat der Benutzer die Möglichkeit, das



Verhalten der Symbolleiste der Authentic-Ansicht vollständig neu zu definieren, indem er seine eigenen Methoden zur Tabellenbearbeitung usw. hinzufügt.

OnToolBarButtonExecuted

**Event:** [OnToolBarButtonExecuted](#) (Button identifier)

**Beschreibung:** [OnToolBarButtonClicked](#) wird ausgelöst, wenn ein Benutzer auf eine Symbolleisten-Schaltfläche klickt. Über den Parameter Schaltflächen-Identifizier, kann ermittelt werden, auf welche Schaltfläche geklickt wurde. Siehe Liste der vordefinierten [vordefinierten Schaltflächen-Identifizier](#).

[OnToolBarButtonExecuted](#) wird ausgelöst, nachdem die Symbolleisten-Aktion ausgeführt wurde und eignet sich z.B. zum Hinzufügen von aktualisiertem Code, wie im Beispiel unten gezeigt:

```
//event fired when a toolbar button action was executed
function On_AuthenticToolBarButtonExecuted(varBtnIdentifizier)
{
 // After whatever command user has executed - make sure to update
 toolbar button states
 UpdateOwnToolBarButtonStates();
}
```

In diesem Fall ist [UpdateOwnToolBarButtonStates](#) eine in den globalen Deklarationen definierte Benutzerfunktion.

OnUserAddedXMLNode

**Event:** [OnUserAddedXMLNode](#) (XML node)

**Beschreibung:** [OnUserAddedXMLNode](#) wird ausgelöst, wenn der Benutzer einen XML-Node als Primäraktion hinzufügt. Dies geschieht in Situationen, in denen der Benutzer eine der folgenden Aktionen ausführt:

- Klicken auf automatisch hinzugefügte Hyperlinks (siehe Beispiel [OnUserAddedXMLNode.sps](#))
- Auswahl der Kontextmenübefehle "Einfügen...", "Einfügen nach...", "Einfügen vor..."
- Klicken auf "Zeile anhängen", beim Einfügen neuer Symbolleisten-Schaltflächen
- Ausführen von "Einfügen nach...", "Einfügen vor..." Aktionen in Elementeingabehilfen (außerhalb von StyleVision)

Das Event wird **nicht** bei "Zeile duplizieren" oder, wenn der Node extern (z.B. über die COM API) hinzugefügt wurde, oder bei "Anwenden" (z.B. Textstatus-Symbole) oder in XML-Tabellenoperationen bzw. in Datenbankoperationen ausgelöst.

Der Event-Parameter ist das XML-Node-Objekt, das hinzugefügt wurde, wodurch der Benutzer die Möglichkeit hat, den hinzugefügten XML-Node zu bearbeiten. Ein ausführliches Beispiel für einen Event Handler finden Sie in der Datei [OnUserAddedXMLNode.sps](#).

Application

**Siehe auch**

**Eigenschaft:** [Application](#) als [Application](#) (schreibgeschützt)

**Beschreibung**

Ruft das StyleVision application-Objekt auf.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

AsXMLString

**Siehe auch**

**Eigenschaft:** [AsXMLString](#) als String

**Beschreibung**

Gibt den Dokumentinhalt als XML-String zurück oder definiert ihn als solchen. Wenn der Inhalt auf einen neuen Wert gesetzt wird, wird die gerade verwendete Schema- oder sps-Datei nicht geändert. Wenn der neue `XMLString` der aktuellen Schemadatei nicht entspricht, wird Fehler 2011 zurückgegeben.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2011 `AsXMLString` wurde auf einen Wert gesetzt, der gemäß der aktuellen Schemadatei kein gültiger XML-Code ist.

ContextMenu

**Eigenschaft:** [ContextMenu\(\)](#) als `ContextMenu`

**Beschreibung**

Über die Eigenschaft `ContextMenu` erhält der Benutzer Zugriff auf das Kontextmenü. Am besten lässt sich dies über den Event Handler `OnContextMenuActivated` durchführen.

**Fehler**

- 2000 Ungültiges Objekt.
- 2005 Ungültiger Parameter.

CreateXMLNode

**Methode:** [CreateXMLNode](#) (*nKind* als `SPYXMLDataKind`) als `XMLData`

**Rückgabewert**

Die Methode gibt das neue `XMLData` Objekt zurück.

**Beschreibung**

Um ein neues `XMLData` Objekt zu erstellen, verwenden Sie die Methode `CreateXMLNode()`. Siehe auch "Verwendung von `XMLData`".

**Fehler**

- 2000 Ungültiges Objekt.
- 2012 XML-Node kann nicht erstellt werden.

DisableAttributeEntryHelper

**Methode:** [DisableAttributeEntryHelper\(\)](#)

**Beschreibung**

`DisableAttributeEntryHelper()` deaktiviert die Attribut-Eingabehilfe in XMLSpy, Authentic Desktop und dem Authentic Browser Plug-in.

**Fehler**

- 2000 Ungültiges Objekt.

DisableElementEntryHelper

**Methode:** `DisableElementEntryHelper()`

**Beschreibung**

`DisableElementEntryHelper()` deaktiviert die Element-Eingabehilfe in XMLSpy, Authentic Desktop und dem Authentic Browser Plug-in.

**Fehler**

2000 Ungültiges Objekt.

DisableEntityEntryHelper

**Methode:** `DisableEntityEntryHelper()`

**Beschreibung**

`DisableEntityEntryHelper()` deaktiviert die Entity-Eingabehilfe in XMLSpy, Authentic Desktop und dem Authentic Browser Plug-in.

**Fehler**

2000 Ungültiges Objekt.

DocumentBegin

**Siehe auch**

**Eigenschaft:** `DocumentBegin` als [AuthenticRange](#) (schreibgeschützt)

**Beschreibung**

Ruft ein Bereichsobjekt auf, das auf den Anfang des Dokuments zeigt.

**Fehler**

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.  
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DocumentEnd

**Siehe auch**

**Eigenschaft:** `DocumentEnd` als [AuthenticRange](#) (schreibgeschützt)

**Beschreibung**

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

**Fehler**

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.  
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DoNotPerformStandardAction

**Methode** `DoNotPerformStandardAction()`

**Beschreibung**

`DoNotPerformStandardAction()` dient als Abbrech-Bubble für Makros und bricht die weitere Ausführung ab, nachdem das Makro fertig ausgeführt wurde.

**Fehler**

2000 Ungültiges Objekt.

## EvaluateXPath

**Methode:** `EvaluateXPath` (XMLData als [XMLData](#), strExpression als string) strValue als string

### Rückgabewert

Diese Methode gibt einen String zurück.

### Beschreibung

`EvaluateXPath()` führt einen XPath-Ausdruck im angegebenen XML-Kontext-Node aus. Das Ergebnis wird als String zurückgegeben. Im Fall einer Sequenz ist der String durch Leerzeichen getrennt.

### Fehler

- 2000 Ungültiges Objekt.
- 2005 Ungültiger Parameter.
- 2008 Interner Fehler.
- 2013 XPath-Fehler.

## Event

### Siehe auch

**Eigenschaft:** `Event` als `AuthenticEvent` (schreibgeschützt)

### Beschreibung

Über diese Eigenschaft erhalten Sie auf dieselbe Art wie bei `ObjAuthenticView.event` Zugriff auf die Parameter des letzten Event. Da nun alle Events für die Skripting-Umgebung und externe Clients mit Parameters verfügbar sind, sollte diese `Event` Eigenschaft nur von IDE-Plug-ins aus verwendet werden.

### Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## EventContext

**Eigenschaft:** `EventContext()` als `EventContext`

### Beschreibung

Über die Eigenschaft [EventContext](#) erhalten Sie Zugriff auf den Kontext des laufenden Makros. Nähere Informationen dazu finden Sie unter der Beschreibung zur [EventContext](#) Schnittstelle.

### Fehler

- 2000 Ungültiges Objekt.

## GetToolBarButtonState

**Methode:** `GetToolBarButtonState` (ButtonIdentifizier als `string`) als `AuthenticToolBarButtonState`

### Rückgabewert

Die Methode gibt `AuthenticToolBarButtonState` zurück.

### Beschreibung

`GetToolBarButtonState` fragt den Status der Symbolleisten-Schaltfläche ab und gestattet dem

Benutzer, die durch ihren Schaltflächen-Identifizier gekennzeichnete Schaltfläche ([siehe Liste oben](#)) zu aktivieren bzw. zu deaktivieren. Auf diese Art können die Symbolleisten-Schaltflächen z.B. permanent deaktiviert werden. Eine andere Verwendung der Methode ist `SetToolBarButtonState` in den `OnSelectionChanged` Event Handler zu setzen, da die Symbolleisten-Schaltflächen regelmäßig aktualisiert werden, wenn sich die Auswahl im Dokument ändert.

Der Status der Symbolleisten-Schaltflächen wird unter der Liste der Enumerationen aufgelistet.

Der Standardzustand bedeutet, dass die Aktivierung/Deaktivierung der Schaltfläche über `AuthenticView` erfolgt. Wenn der Benutzer den Zustand auf aktiviert oder deaktiviert setzt, bleibt die Schaltfläche so lange in diesem Zustand, solange der Benutzer den Zustand nicht ändert.

#### Fehler

- 2000 Ungültiges Objekt.
- 2005 Ungültiger Parameter.
- 2008 Interner Fehler.
- 2014 Ungültiger Schaltflächen-Identifizier.

Goto

#### Siehe auch

**Methode:** `Goto` (*eKind* als `SPYAuthenticElementKind`, *nCount* als Long, *eFrom* als `SmYAuthenticDocument position`) als `AuthenticRange`

#### Beschreibung

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

#### Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.  
Die Ausgangsposition des Dokuments ist keine `spyAuthenticDocumentBegin` oder `spyAuthenticDocumentEnd` Position.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

On Error Resume Next
Dim objRange
' goto beginning of first table in document
Set objRange = objAuthenticView.Goto (spyAuthenticTable, 1,
spyAuthenticDocumentBegin)
If (Err.number = 0) Then
 objRange.Select()
Else
 MsgBox "No table found in document"
End If
```

IsRedoEnabled

**Siehe auch**

**Eigenschaft:** [IsRedoEnabled](#) als Boolean (schreibgeschützt)

**Beschreibung**

True, wenn Wiederhol-Schritte verfügbar sind und [Redo](#) möglich ist.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsUndoEnabled

**Siehe auch**

**Eigenschaft:** [IsUndoEnabled](#) als Boolean (schreibgeschützt)

**Beschreibung**

True, wenn Rückgängig-Schritte verfügbar sind und [Undo](#) möglich ist.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MarkupVisibility

**Siehe auch**

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

**Beschreibung**

Setzt oder ruft die aktuelle Sichtbarkeit des Markup-Codes ab.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Enumerationswert angegeben.  
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

**Siehe auch**

**Eigenschaft:** [Parent](#) als [Document](#) (schreibgeschützt)

**Beschreibung**

Ruft das in dieser Ansicht angezeigte Dokument auf.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Print

**Siehe auch**

**Methode:** `Print` (*bWithmreview* als Boolean, *bmromptUser* als Boolean)

### Beschreibung

Druckt das in dieser Ansicht angezeigte Dokument. Wenn *bWithPreview* auf *True* gesetzt ist, wird das Dialogfeld "Druckvorschau" angezeigt. Wenn *bPromptUser* auf *True* gesetzt ist, wird das Dialogfeld "Drucken" angezeigt. Wenn beide Parameter auf *False* gesetzt sind, wird das Dokument ohne weitere Eingabe durch den Benutzer gedruckt.

### Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

Redo

**Siehe auch**

**Methode:** `Redo()` als Boolean

### Beschreibung

Stellt die durch den letzten Rückgängig-Befehl rückgängig gemachte Änderung wieder her.

### Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Selection

**Siehe auch**

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

### Beschreibung

Setzt oder ruft die aktuelle Textauswahl auf der Benutzeroberfläche ab.

### Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

2002 Es ist keine Cursorauswahl aktiv.

2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' if we are the end of the document, re-start at the beginning
If (objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then
 objAuthenticView.Selection = objAuthenticView.DocumentBegin
Else
 ' objAuthenticView.Selection =
objAuthenticView.Selection.GotoNextCursorPosition()
 ' or shorter:
 objAuthenticView.Selection.GotoNextCursorPosition().Select
End If
```

## SetToolBarButtonState

**Methode:** `SetToolBarButtonState` (ButtonIdentifier als string, AuthenticToolBarButtonState state)

### Beschreibung

`SetToolBarButtonState` fragt den Status der Symbolleisten-Schaltfläche ab und gestattet dem Benutzer, die durch ihren Schaltflächen-Identifizier gekennzeichnete Schaltfläche ([siehe Liste oben](#)) zu aktivieren bzw. zu deaktivieren. Auf diese Art können die Symbolleisten-Schaltflächen z.B. permanent deaktiviert werden. Eine andere Verwendung der Methode ist `SetToolBarButtonState` in den `OnSelectionChanged` Event Handler zu setzen, da die Symbolleisten-Schaltflächen regelmäßig aktualisiert werden, wenn sich die Auswahl im Dokument ändert.

Der Status der Symbolleisten-Schaltflächen wird unter der Liste der Enumerationen aufgelistet.

Der Standardzustand bedeutet, dass die Aktivierung/Deaktivierung der Schaltfläche über `AuthenticView` erfolgt. Wenn der Benutzer den Zustand auf aktiviert oder deaktiviert setzt, bleibt die Schaltfläche so lange in diesem Zustand, solange der Benutzer den Zustand nicht ändert.

### Fehler

- 2000 Ungültiges Objekt.
- 2008 Interner Fehler.
- 2014 Ungültiger Schaltflächen-Identifizier.

## Undo

### Siehe auch

**Methode:** `Undo()` als Boolean

### Beschreibung

Macht die letzte Änderung des Dokuments innerhalb dieser Ansicht rückgängig.

### Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## UpdateXMLInstanceEntities

### Siehe auch

**Methode:** `UpdateXMLInstanceEntities()`

### Beschreibung

Aktualisiert die interne Darstellung der deklarierten Entities und befüllt die Eingabehilfen erneut. Zusätzlich dazu wird der Validator neu geladen, sodass die XML-Datei korrekt validiert werden kann. Beachten Sie bitte, dass dadurch auch Schemadateien neu geladen werden können.

### Fehler

Die Methode gibt niemals einen Fehler zurück.

### Beispiel

```
// -----
// XMLSpy scripting environment - JavaScript
// -----
if(Application.ActiveDocument &&
(Application.ActiveDocument.CurrentViewMode == 4))
```



```
{
 var objDocType;
 objDocType =
Application.ActiveDocument.DocEditView.XMLRoot.GetFirstChild(10);

 if(objDocType)
 {
 var objEntity = Application.ActiveDocument.CreateChild(14);
 objEntity.Name = "child";
 objEntity.TextValue = "SYSTEM \"child.xml\"";
 objDocType.AppendChild(objEntity);

Application.ActiveDocument.AuthenticView.UpdateXMLInstanceEntities();
 }
}
```

WholeDocument

**Siehe auch**

**Eigenschaft:** `WholeDocument` als [AuthenticRange](#) (schreibgeschützt)

**Beschreibung**

Ruft ein Bereichsobjekt ab, das das gesamte Dokument auswählt.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLDataRoot

**Siehe auch**

**Eigenschaft:** `XMLDataRoot` als [XMLData](#) (schreibgeschützt)

**Beschreibung**

Gibt das am höchsten gestufte XMLData-Element des aktuellen Dokuments zurück oder definiert dieses. Dieses Element beschreibt normalerweise die Dokumentstruktur und wäre von der Art `spyXMLDataXMLDocStruct`, `spyXMLDataXMLEntityDocStruct` oder `spyXMLDataDTDDocStruct`.

**Fehler**

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## Document

Die `Document` Schnittstelle verfügt über die folgenden Methoden und Eigenschaften:

### Methoden

- [Activate](#)
- [AssignWorkingXMLFile](#)
- [Close](#)
- [Save](#)
- [SaveAs](#)
- [Saved](#)
- [SaveGeneratedFOFile](#)
- [SaveGeneratedFOFileEx](#)
- [SaveGeneratedHTMLFile](#)
- [SaveGeneratedHTMLFileEx](#)
- [SaveGeneratedPDFFile](#)
- [SaveGeneratedPDFFileEx](#)
- [SaveGeneratedRTFFile](#)
- [SaveGeneratedRTFFileEx](#)
- [SaveGeneratedWord2007File](#)
- [SaveGeneratedWord2007FileEx](#)
- [SaveGeneratedXSLTFOFile](#)
- [SaveGeneratedXSLTFOFileEx](#)
- [SaveGeneratedXSLTHTMLFile](#)
- [SaveGeneratedXSLTHTMLFileEx](#)
- [SaveGeneratedXSLTRTFFile](#)
- [SaveGeneratedXSLTRTFFileEx](#)
- [SaveGeneratedXSLTWord2007File](#)
- [SaveGeneratedXSLTWord2007FileEx](#)

### Eigenschaften

- [Application](#)
- [FullName](#)
- [Name](#)
- [Parameters](#)
- [Parent](#)
- [Path](#)
- [SchemaSources](#)

### Events

#### OnDocumentClosed

**Event:** `OnDocumentClosed`(*Document* als [Document](#))

#### Beschreibung

Dieses Event wird durch das Schließen eines Dokuments ausgelöst.

#### OnModifiedFlagChanged

**Event:** `OnModifiedFlagChanged`(als Boolean)

#### Beschreibung

Gibt "true" zurück, wenn das "Modified" Flag geändert wurde.

Activate

**Methode:** `Activate ()`

**Beschreibung**

Aktiviert den Dokumentrahmen.

**Fehler**

- 1200 Das Document-Objekt ist ungültig.
- 1201 Ungültiger Eingabeparameter.

Application

**Siehe auch**

**Eigenschaft:** `Application` als `Application` (schreibgeschützt)

**Beschreibung**

Ruft das StyleVision application-Objekt auf.

**Fehler**

- 1400 Das Objekt ist nicht mehr gültig.
- 1111
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1100

AssignWorkingXMLFile

**Methode:** `AssignWorkingXMLFile` (`strWorkingXMLFileName` als String, `strSchemaSourceName` als String)

**Beschreibung**

Weist die XML-Arbeitsdatei und die Schemaquelle durch Angabe der URIs als Strings zu. Der Name der Schemaquelle ist der Name, der dem Schema im SPS zugewiesen wurde; (der Standard-Schemaname in einem SPS, das in StyleVision neu erstellt wurde, ist: `XML`).

**Fehler**

- 1200 Das document-Objekt ist ungültig.
- 1201 Ungültiger Eingabeparameter.
- 1203 Fehler beim Zuweisen der XML-Arbeitsdatei
- 1404 Fehlendes XML-Schema oder DTD

Close

**Siehe auch**

**Methode:** `Close` (`bDiscardChanges` als Boolean)

**Beschreibung**

Rufen Sie diese Methode auf, um ein Dokument zu schließen. Wenn `bDiscardChanges` "true" ist und das Dokument geändert wurde, wird das Dokument geschlossen aber nicht gespeichert.

**Fehler**

- 1400 Das Objekt ist nicht mehr gültig.
- 1401 Das Dokument muss zuerst gespeichert werden.

FullName

**Siehe auch**

**Eigenschaft:** [FullName](#) als String

### Beschreibung

Mit Hilfe dieser Eigenschaft kann der vollständige Dateiname - einschließlich des Pfads - unter dem das Dokument gespeichert wird, abgerufen oder definiert werden. Die Gültigkeit des Namens wird vor dem nächsten Speichervorgang nicht überprüft.

Diese Eigenschaft macht die Methoden `GetPathName` und `SetPathName` überflüssig.

### Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1402 Es wurde ein leerer String als vollständiger Dateiname definiert.

Name

**Siehe auch**

**Eigenschaft:** [Name](#) als String (schreibgeschützt)

### Beschreibung

Verwenden Sie diese Eigenschaft, um den Namen - ohne Pfad - der Dokumentdatei abzurufen. Um den Dateinamen für ein Dokument zu ändern, verwenden Sie die Eigenschaft [FullName](#).

### Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parameters

**Eigenschaft:** [Parameters](#) als [Parameters](#) (schreibgeschützt)

### Beschreibung

Referenz zum aktuellen `Parameters` Objekt.

### Fehler

- 1200 Das document-Objekt ist ungültig.
- 1201 Ungültiger Eingabeparameter.

Parent

### Beschreibung

Ruft das `StyleVision application` Objekt auf.

### Fehler

- 1111 Das document-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Path

**Siehe auch**

**Eigenschaft:** `Path` als String (schreibgeschützt)

**Beschreibung**

Verwenden Sie diese Eigenschaft, um den Pfad der Dokumentdatei - ohne Dateinamen - abzurufen. Um den Dateinamen und Pfad für ein Dokument zu ändern, verwenden Sie die Eigenschaft [FullName](#).

**Fehler**

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Save

**Siehe auch**

**Methode:** `Save()`

**Beschreibung**

Die Methode schreibt alle Änderungen des Dokuments in die dazugehörige Datei.

**Fehler**

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurde ein leerer Dateiname angegeben.
- 1403 Fehler beim Speichern der Datei. Wahrscheinlich ist der Dateiname ungültig.

SaveAs

**Siehe auch**

**Methode:** `SaveAs` (`strFileName` als String)

**Beschreibung**

Speichert das Dokument in der angegebenen Datei.

**Fehler**

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurde ein leerer Dateiname angegeben.
- 1403 Fehler beim Speichern der Datei. Wahrscheinlich ist der Dateiname ungültig.

Saved

**Siehe auch**

**Eigenschaft:** `Saved` als Boolean (schreibgeschützt)

**Beschreibung**

Mit Hilfe dieser Eigenschaft können Sie überprüfen, ob das Dokument seit der letzten Änderung gespeichert wurde.

**Fehler**

- 1400 Das document-Objekt ist nicht mehr gültig.

1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SaveGeneratedFOFile

**Methode:** `SaveGeneratedFOFile` (*strFileName* als String)

**Beschreibung**

Speichert die generierte Datei unter dem angegebenen Pfad.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedFOFileEx

**Methode:** `SaveGeneratedFOFileEx` (*strFileName* als String, *pbError* als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedHTMLFile

**Methode:** `SaveGeneratedHTMLFile` (*strFileName* als String)

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedHTMLFileEx

**Methode:** `SaveGeneratedHTMLFileEx` (*strFileName* als String, *pbError* als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedPDFFile

**Methode:** `SaveGeneratedPDFFile` (*strFileName* als String)

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

## SaveGeneratedPDFFileEx

**Methode:** `SaveGeneratedPDFFileEx` (`strFileName` als String, `pbError` als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

## SaveGeneratedRTFFile

**Methode:** `SaveGeneratedRTFFile` (`strFileName` als String)

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

## SaveGeneratedRTFFileEx

**Methode:** `SaveGeneratedRTFFileEx` (`strFileName` als String, `pbError` als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

## SaveGeneratedWord2007File

**Methode:** `SaveGeneratedWord2007File` (`strFileName` als String)

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

## SaveGeneratedWord2007FileEx

**Methode:** `SaveGeneratedWord2007FileEx` (`strFileName` als String, `pbError` als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedXSLTFOFile

**Methode:** `SaveGeneratedXSLTFOFile` (*strFileName* als String)

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedXSLTFOFileEx

**Methode:** `SaveGeneratedXSLTFOFileEx` (*strFileName* als String, *pbError* als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedXSLTHTMLFile

**Methode:** `SaveGeneratedXSLTHTMLFile` (*strFileName* als String)

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

SaveGeneratedXSLTHTMLFileEx

**Methode:** `SaveGeneratedXSLTHTMLFileEx` (*strFileName* als String, *pbError* als Variant) als AppOutputLines

**Beschreibung**

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

**Fehler**

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.



### SaveGeneratedXSLTRTFFFile

**Methode:** `SaveGeneratedXSLTRTFFFile` (*strFileName* als String)

#### Beschreibung

Speichert die generierte Datei im angegebenen Ordner

#### Fehler

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

### SaveGeneratedXSLTRTFFFileEx

**Methode:** `SaveGeneratedXSLTRTFFFileEx` (*strFileName* als String, *pbError* als Variant) als AppOutputLines

#### Beschreibung

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

#### Fehler

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

### SaveGeneratedXSLTWord2007File

**Methode:** `SaveGeneratedXSLTWord2007File` (*strFileName* als String)

#### Beschreibung

Speichert die generierte Datei im angegebenen Ordner

#### Fehler

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

### SaveGeneratedXSLTWord2007FileEx

**Methode:** `SaveGeneratedXSLTWord2007FileEx` (*strFileName* als String, *pbError* als Variant) als AppOutputLines

#### Beschreibung

Speichert die generierte Datei im angegebenen Ordner und gibt eine Fehlerbeschreibung.

#### Fehler

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

### SchemaSources

**Eigenschaft:** `SchemaSources` als `SchemaSources` (schreibgeschützt)

#### Beschreibung

Referenz zum aktuellen `SchemaSources` Objekt

#### Fehler

- 1201 Ungültiges document-Objekt.
- 1204 Die Ausgabedatei kann nicht generiert werden.

## Documents

Die `Documents` Schnittstelle verfügt über die folgenden Methoden und Eigenschaften:

### Methoden

- [ActiveDocument](#)
- `Item`
- [NewDocument](#)
- [OpenDocument](#)

### Eigenschaften

- [Application](#)
- `Count`
- [Parent](#)

#### ActiveDocument

**Eigenschaft:** `ActiveDocument` als [Document](#)

#### Beschreibung

Referenz zum aktiven Dokument. Wenn kein Dokument geöffnet ist, ist `ActiveDocument` `Null` (nichts).

#### Fehler

- |      |                                                                   |
|------|-------------------------------------------------------------------|
| 1111 | Das application-Objekt ist nicht mehr gültig.                     |
| 1100 | Für den Rückgabeparameter wurde eine ungültige Adresse angegeben. |
| 1600 | Ungültiges <code>Documents</code> Objekt                          |
| 1601 | Ungültiger Eingabeparameter                                       |

#### Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

#### Beschreibung

Ruft das `StyleVision` application-Objekt auf.

#### Fehler

- |      |                                                                  |
|------|------------------------------------------------------------------|
| 1111 | Das application-Objekt ist nicht mehr gültig.                    |
| 1100 | Für den Rückgabeparameter wurde eine ungültige Adresse angegeben |

#### NewDocument

**Methode:** `NewDocument()` als [Document](#)

#### Rückgabewert

Keiner

#### Beschreibung

Erstellt ein neues leeres Dokument anhand der vorherigen Vorlage.

#### Fehler

- 1000 Das application-Objekt ist nicht mehr gültig.
- 1005 Fehler beim Erstellen eines neuen Dokuments
- 1006 Das Dokument kann nicht erstellt werden
- 1600 Ungültiges `Documents` Objekt
- 1601 Ungültiger Eingabeparameter

#### OpenDocument

**Methode:** `OpenDocument(strFileName als String)` als [`Document`](#)

#### Rückgabewert

Keiner

#### Beschreibung

Öffnet eine vorhandene SPS-Datei.

#### Fehler

- 1000 Das application-Objekt ist ungültig.
- 1002 Ungültige Dateierweiterung.
- 1003 Fehler beim Öffnen des Dokuments.
- 1004 Das Dokument kann nicht geöffnet werden.
- 1600 Ungültiges `Documents` Objekt
- 1601 Ungültiges Eingabeparameter

#### Parent

**Eigenschaft:** `Parent` als [`Application`](#) (schreibgeschützt)

#### Beschreibung

Ruft das StyleVision application-Objekt auf.

#### Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## Parameter

Die `Parameter` Schnittstelle verfügt über die folgenden Eigenschaften:

### Eigenschaften

- [Application](#)
- [Name](#)
- [Parent](#)
- [Value](#)

#### Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

#### Beschreibung

Ruft das StyleVision application-Objekt auf.

#### Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### Name

**Eigenschaft:** `Name` als String (schreibgeschützt)

#### Fehler

- 1600 Ungültiges `Documents` Objekt
- 1601 Ungültiger Eingabeparameter

#### Parent

**Eigenschaft:** `Parent` als [Application](#) (schreibgeschützt)

#### Beschreibung

Ruft das StyleVision application Objekt auf.

#### Fehler

- 1111 Das document-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### Value

**Eigenschaft:** `Value` als String (schreibgeschützt)

#### Fehler

- 1600 Ungültiges `Documents` Objekt
- 1601 Ungültiger Eingabeparameter

## Parameters

Die `Parameters` Schnittstelle verfügt über die folgenden Eigenschaften:

### Eigenschaften

- [Application](#)
- [Count](#)
- [Item](#)
- [Parent](#)

#### Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

#### Beschreibung

Ruft das StyleVision application Objekt auf.

#### Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

#### Count

**Eigenschaft:** `Count` als long

#### Beschreibung

Anzahl der Parameter

#### Fehler

- 1600 Ungültiges `Documents` Objekt
- 1601 Ungültiger Eingabeparameter

#### Item

**Methode:** `Item` (`n` als long) als [Document](#)

#### Beschreibung

Ruft das Dokument mit dem Index `n` in dieser Sammlung ab. Der Index ist 1-basiert.

#### Fehler

- 1600 Ungültiges `Documents` Objekt
- 1601 Ungültiger Eingabeparameter

#### Parent

**Eigenschaft:** `Parent` als [Application](#) (schreibgeschützt)

#### Beschreibung

Ruft das StyleVision application Objekt auf.

#### Fehler

- 1111 Das document-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## SchemaSource

Die `SchemaSource` Schnittstelle verfügt über die folgenden Parameter:

### Eigenschaften

- [Application](#)
- [IsMainSchemaSource](#)
- [Name](#)
- [Parent](#)
- [SchemaFileName](#)
- [TemplateFileName](#)
- [Type](#)
- [TypeName](#)
- [WorkingXMLFileName](#)

Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

### Beschreibung

Ruft das `StyleVision application`-Objekt auf.

### Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsMainSchemaSource

**Eigenschaft:** `IsMainSchemaSource` als Boolean (schreibgeschützt)

### Beschreibung

Gibt "true" zurück, wenn die Schemaquelle die Hauptschemaquelle ist.

### Fehler

- 1400 Ungültiges Schemaquell-Objekt.
- 1401 Ungültiger Parameter

Name

**Eigenschaft:** `Name` als String (schreibgeschützt)

### Beschreibung

Über diese Eigenschaft können Sie den Namen der Schemaquelle abrufen.

### Fehler

- 1400 Das Schemaquellobjekt ist nicht gültig
- 1401 Ungültiger Parameter

Parent

**Eigenschaft:** `Parent` als `SchemaSource` (schreibgeschützt)

### Beschreibung

Das Parent-Objekt gemäß dem Objektmodell.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SchemaFileName

**Eigenschaft:** [SchemaFileName](#) als String

**Beschreibung**

Über diese Eigenschaft können Sie den Namen der Schemadatei abrufen.

**Fehler**

- 1400 Ungültiges Schemaquelle-Objekt.
- 1401 Ungültiger Parameter.
- 1403 Fehlendes XML-Schema oder fehlende DTD
- 1406 Fehler beim Zuweisen der Schemadatei

TemplateFileName

**Eigenschaft:** [TemplateFileName](#) als String

**Beschreibung**

Über diese Eigenschaft können Sie den Namen der XML-Arbeitsdatei abrufen.

**Fehler**

- 1400 Ungültiges Schemaquelle-Objekt.
- 1401 Ungültiger Parameter.
- 1403 Fehlendes XML-Schema oder fehlende DTD
- 1407 Fehler beim Zuweisen der XML-Vorlagendatei

Type

**Eigenschaft:** [Type](#) als ENUMSchemaSourceType (schreibgeschützt)

**Beschreibung**

Über diese Eigenschaft können Sie den Typ der Schemaquelle abrufen.

**Fehler**

- 1400 Ungültiges Schemaquelle-Objekt.
- 1401 Ungültiger Parameter

TypeName

**Eigenschaft:** [Type](#) als String (schreibgeschützt)

**Beschreibung**

Über diese Eigenschaft können Sie den Typ der Schemaquelle abrufen.

**Fehler**

- 1400 Ungültiges Schemaquelle-Objekt.
- 1401 Ungültiger Parameter

WorkingXMLFileName

**Eigenschaft:** [WorkingXMLFileName](#) als String

**Beschreibung**

Über diese Eigenschaft können Sie den Namen der XML-Arbeitsdatei abrufen.

**Fehler**

- 1400 Ungültiges Schemaquelle-Objekt.
- 1401 Ungültiger Parameter.
- 1403 Fehlendes XML-Schema oder fehlende DTD
- 1203 Fehler beim Zuweisen der XML-Arbeitsdatei



## SchemaSources

Die `SchemaSources` Schnittstelle verfügt über die folgenden Eigenschaften:

### Eigenschaften

- [Application](#)
- [MainSchemaSource](#)
- [Parent](#)

Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

### Beschreibung

Ruft das StyleVision application-Objekt auf.

### Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Count

**Eigenschaft:** `Count` als long (schreibgeschützt)

### Beschreibung

Anzahl der Schemaquellen.

### Fehler

- 1300 Ungültiges `SchemaSources` Objekt
- 1301 Ungültiger Eingabeparameter

Item

**Method:** `Item` (`n` as long) as [Document](#)

### Beschreibung

Ruft das Dokument mit dem Index `n` in dieser Sammlung ab. Der Index ist 1-basiert.

### Fehler

- 1300 Ungültiges `SchemaSources` Objekt
- 1301 Ungültiger Eingabeparameter

MainSchemaSource

**Eigenschaft:** `MainSchemaSource` als `SchemaSource` (schreibgeschützt)

### Fehler

- 1300 Ungültiges `SchemaSources` Objekt
- 1301 Ungültiger Eingabeparameter

Parent

**Eigenschaft:** `Parent` als SchemaSources (schreibgeschützt)

**Beschreibung**

Das Parent-Objekt gemäß dem Objektmodell.

**Fehler**

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## StyleSheet

Die `StyleSheet` verfügt über die folgenden Methoden und Eigenschaften:

### Methoden

- `Save`
- `SaveAs`
- `SaveGeneratedXSLTFile`
- `SaveGeneratedXSLTFOFile`

### Eigenschaften

- [Application](#)
- [FullName](#)
- `Name`
- `Parent`
- `Path`
- `Saved`

Application

**Eigenschaft:** `Application` als [Application](#) (schreibgeschützt)

### Beschreibung

Ruft das `StyleVision` application-Objekt auf.

### Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FullName

### Siehe auch

**Eigenschaft:** `FullName` als String

### Beschreibung

Mit Hilfe dieser Eigenschaft kann der vollständige Dateiname - einschließlich des Pfads - unter dem das Dokument gespeichert wird, abgerufen oder definiert werden. Die Gültigkeit des Namens wird vor dem nächsten Speichervorgang nicht überprüft.

Diese Eigenschaft macht die Methoden `GetPathName` und `SetPathName` überflüssig.

### Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1402 Es wurde ein leerer String als vollständiger Dateiname definiert.

**XMLData****Siehe auch****Eigenschaften**[Kind](#)[Name](#)[TextValue](#)[HasChildren](#)[MayHaveChildren](#)[Parent](#)**Methoden**[GetFirstChild](#)[GetNextChild](#)[GetCurrentChild](#)[InsertChild](#)[InsertChildAfter](#)[InsertChildBefore](#)[AppendChild](#)[EraseAllChildren](#)[EraseChild](#)[EraseCurrentChild](#)[IsSameNode](#)[CountChildren](#)[CountChildrenKind](#)[GetChild](#)[GetChildAttribute](#)[GetChildElement](#)[GetChildKind](#)[GetNamespacePrefixForURI](#)[HasChildrenKind](#)[SetTextValueXMLEncoded](#)**Beschreibung**

Über die XMLData-Schnittstelle haben Sie auf XML-Ebene direkten Zugriff auf das Dokument. Sie können die XML-Darstellung des Dokuments anzeigen und direkt bearbeiten. Beachten Sie jedoch die folgenden Einschränkungen:

- Die XMLData-Darstellung ist nur gültig, wenn das Dokument in der Grid-Ansicht oder der Authentic-Ansicht angezeigt wird.
- In der Authentic-Ansicht werden zusätzliche XMLData-Elemente automatisch als übergeordnete Elemente der einzelnen sichtbaren Dokumentelemente eingefügt. Normalerweise handelt es sich dabei um ein XMLData-Element von der Art `spyXMLDataElement`, wobei die [Name](#) Eigenschaft auf 'Text' gesetzt ist.
- Wenn Sie die XMLData-Schnittstelle von einem anderen Ansichtsmodus aus verwenden, werden zwar keine Fehler angezeigt, aber die Änderungen werden in der Ansicht nicht angezeigt und gehen u.U. beim nächsten Ansichtswechsel verloren.

Beachten Sie bitte außerdem:

- Das Definieren eines neuen Textwerts für ein XML-Element ist möglich, wenn das Element keine Unterelemente hat, die keinen Text enthalten. Ein Textwert kann sogar definiert werden, wenn das Element Attribute enthält.
- Wenn Sie für ein XML-Element, das mehr als ein Text-Child-Element hat, einen neuen Textwert definieren, so wird das letzte davon gelöscht und durch ein neues Text-Child-Element ersetzt.
- Wenn der Textwert eines XML-Elements mit mehr als einem Text-Child-Element gelesen wird, so wird nur der Wert des ersten Text-Child-Elements zurückgegeben.

Objekte dieser Klasse repräsentieren die unterschiedlichen Bestandteile eines XML-Dokuments. Informationen zu den verfügbaren Teiltypen finden Sie unter dem Enumerationstyp `SPYXMLDataKind`. Jeder Teil kennt seine untergeordneten Elemente, wodurch eine XMLData-Struktur mit `DocumentRootElement` auf der obersten Ebene gebildet wird. Um das oberste Element des Dokumentinhalts (ohne XML Header) abzurufen, verwenden Sie `DocumentDataRoot`. Beispiele, wie man die XMLData-Struktur durchquert, finden Sie unter `UsingXMLData to modify document structure` und [GetNextChild](#).

## AppendChild

### Siehe auch

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

### Beschreibung

`AppendChild` hängt *pNewData* als letztes Child-Element an das `XMLData`-Objekt an. Siehe auch `UsingXMLData`.

### Fehler

- 1500 Das XMLData-Element ist nicht mehr gültig.
- 1505 Es wurde eine ungültige XMLData-Art definiert.
- 1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1507 Das Element kann keine untergeordneten Elemente haben.
- 1512 Zyklische Einfügung - ein neues Datenelement ist bereits Teil des Dokuments.
- 1514 Es wurde eine ungültige XMLData-Art für diese Position definiert.
- 1900 Das Dokument darf nicht geändert werden.

### Beispiel

```
Dim objCurrentParent As XMLData
Dim objNewChild As XMLData

Set objNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)
Set objCurrentParent = objSpy.ActiveDocument.RootElement

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

## CountChildren

### Siehe auch

**Deklaration:** `CountChildren` als long

### Beschreibung

`CountChildren` ruft die Anzahl der Child-Elemente ab.

Verfügbar mit Typbibliothek Version 1.5

**Fehler**

1500 Das XMLData-Objekt ist nicht mehr gültig.

CountChildrenKind

**Siehe auch**

**Deklaration:** CountChildrenKind (*nKind* als SPYXMLDataKind) als long

**Beschreibung**

CountChildrenKind ruft die Anzahl der Child-Elemente dieser bestimmten Art ab.

Verfügbar mit Typbibliothek Version 1.5

**Fehler**

1500 Das XMLData-Objekt ist nicht mehr gültig.

EraseAllChildren

**Siehe auch**

**Deklaration:** EraseAllChildren

**Beschreibung**

EraseAllChildren löscht alle zum XMLData Objekt gehörigen Child-Elemente.

**Fehler**

1500 Das XMLData-Objekt ist nicht mehr gültig.

1900 Das Dokument darf nicht geändert werden.

**Beispiel**

In diesem Beispiel werden alle Elemente des aktiven Dokuments gelöscht.

```
Dim objCurrentParent As XMLData

Set objCurrentParent = objSpy.ActiveDocument.RootElement
objCurrentParent.EraseAllChildren
```

EraseChild

**Methode:** EraseChild (Child als [XMLData](#))

**Beschreibung**

Löscht den angegebenen Child Node.

**Fehler**

1500 Ungültiges Objekt.

1506 Ungültige XML-Input-Datei.

1510 Ungültiger Parameter.

## EraseCurrentChild

### Siehe auch

**Deklaration:** [EraseCurrentChild](#)

### Beschreibung

[EraseCurrentChild](#) löscht das aktuelle `XMLData` Child-Objekt. Bevor Sie [EraseCurrentChild](#) aufrufen, müssen Sie einen internen Iterator mit [XMLData.GetFirstChild](#) initialisieren. Nachdem Sie das aktuelle Child-Element gelöscht haben, erhöht [EraseCurrentChild](#) den internen Iterator des `XMLData`-Elements. Es wird kein Fehler zurückgegeben, wenn das letzte Child-Element gelöscht wird und der Iterator über das Ende der Child-Liste hinaus verschoben wird. Der nächste Aufruf von [EraseCurrentChild](#) gibt allerdings den Fehler 1503 zurück.

### Fehler

- 1500 Das `XMLData`-Objekt ist nicht mehr gültig.
- 1503 Es wurde für dieses `XMLData`-Objekt kein Iterator initialisiert oder der Iterator zeigt hinter das letzte Child-Element.
- 1900 Das Dokument darf nicht geändert werden.

### Beispiele

```
// -----
// XMLSpy scripting environment - JScript
// erase all children of XMLData
// -----
// let's get an XMLData element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application.ActiveDocument.GridView.CurrentFocus;

// the following line would be shorter, of course
//objList.EraseAllChildren();

// but we want to demonstrate the usage of EraseCurrentChild
if ((objList != null) && (objList.HasChildren))
{
 try
 {
 objEle = objList.GetFirstChild(-1);
 while (objEle != null)
 {
 objList.EraseCurrentChild();
 // no need to call GetNextChild
 }
 }
 catch (err)
 {
 // 1503 - we reached end of child list
 if ((err.number & 0xffff) != 1503) throw (err);
 }
}
```

## GetChild

### Siehe auch

**Deklaration:** [GetChild](#) (*position* als long) als [XMLData](#)

### Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

### Beschreibung

[GetChild](#) gibt eine Referenz auf das Child am angegebenen Index (nullbasiert) zurück.

Verfügbar mit Typbibliothek Version 1.5

**Fehler**

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetChildAttribute

**Methode:** `GetChildAttribute` (strName als string) Child als XMLData Objekt (NULL bei Fehler)

**Beschreibung**

Ruft das Attribut mit dem angegebenen Namen auf.

**Fehler**

- 1500 Ungültiges Objekt.
- 1510 Ungültiger Parameter.

## GetChildElement

**Methode:** `GetChildElement` (strName als string, nIndex als long) Child als XMLData-Objekt (NULL bei Fehler)

**Beschreibung**

Ruft das nte Child Element des angegebenen Namens auf.

**Fehler**

- 1500 Ungültiges Objekt.
- 1510 Ungültiger Parameter.

## GetChildKind

**Siehe auch**

**Deklaration:** `GetChildKind` (*position* als long, *nKind* als `SPYXMLDataKind`) als [XMLData](#)

**Rückgabewert**

Gibt ein XML-Element als `XMLData` Objekt zurück.

**Beschreibung**

`GetChildKind` gibt eine Referenz auf ein Child dieser Art am angegebenen Index (nullbasiert) zurück. Der Positionsparameter ist relativ zur Anzahl der Child-Elemente der angegebenen Art und nicht zu allen Child-Elementen des Objekts.

Verfügbar mit Typbibliothek Version 1.5

**Fehler**

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

## GetCurrentChild

**Siehe auch**



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

### Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

### Beschreibung

`GetCurrentChild` ruft das aktuelle Child-Element ab. Bevor Sie `GetCurrentChild` aufrufen, müssen Sie einen internen Iterator mit [XMLData.GetFirstChild](#) initialisieren.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

`GetFirstChild`

### Siehe auch

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

### Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

### Beschreibung

`GetFirstChild` initialisiert einen neuen Iterator und gibt das erste Child-Element zurück. Setzen Sie `nKind = -1`, um einen Iterator für alle Arten von Child-Elementen abzurufen.  
Anmerkung: Der Iterator wird innerhalb des XMLData-Objekts gespeichert und wird zerstört, wenn das XMLData-Objekt zerstört wird. Stellen Sie sicher, dass Sie eine Referenz auf dieses Objekt beibehalten, solange Sie [GetCurrentChild](#), [GetNextChild](#) oder [EraseCurrentChild](#) verwenden möchten.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1501 Es wurde eine ungültige XMLData-Art definiert.
- 1504 Das Element hat keine untergeordneten Elemente der angegebenen Art.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiel

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

`GetNamespacePrefixForURI`

**Methode:** `GetNamespacePrefixForURI` (`strURI` als `string`) `strNS` als `string`

### Beschreibung

Gibt das Namespace-Präfix der vorgegebenen URI zurück.

### Fehler

- 1500 Ungültiges Objekt.
- 1510 Ungültiger Parameter.

`GetNextChild`

### Siehe auch

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

### Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

### Beschreibung

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

Überprüfen Sie das letzte Child-Element des Elements, wie im Beispiel unten gezeigt.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1503 Für dieses XMLData-Objekt wurde kein Iterator initialisiert.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

```
' -----
' VBA code snippet - iterate XMLData children
' -----

On Error Resume Next
Set objParent = objSpy.ActiveDocument.RootElement

'get elements of all kinds
Set objCurrentChild = objParent.GetFirstChild(-1)

Do
 'do something useful with the child

 'step to next child
 Set objCurrentChild = objParent.GetNextChild
Loop Until (Err.Number - vbObjectError = 1503)

// -----
// XMLSpy scripting environment - JScript
// iterate through children of XMLData
// -----
try
{
 var objXMLData = ... // initialize somehow
 var objChild = objXMLData.GetFirstChild(-1);

 while (true)
 {
 // do something usefull with objChild

 objChild = objXMLData.GetNextChild();
 }
}
catch (err)
{
 if ((err.number & 0xffff) == 1504)
 ; // element has no children
 else if ((err.number & 0xffff) == 1503)
 ; // last child reached
 else
 throw (err);
}
```

GetTextValueXMLDecoded

**Methode:** `GetTextValueXMLDecoded()` als string

**Beschreibung**

Ruft den dekodierten Textwert der XML-Datei ab.

**Fehler**

- 1500 Ungültiges Objekt.
- 1510 Ungültiger Parameter.

HasChildren

**Siehe auch**

**Deklaration:** `HasChildren` als Boolean

**Beschreibung**

Die Eigenschaft ist true, wenn das Objekt das übergeordnete Objekt von anderen `XMLData` Objekten ist. Diese Eigenschaft ist schreibgeschützt.

**Fehler**

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

HasChildrenKind

**Siehe auch**

**Deklaration:** `HasChildrenKind` (`nKind` als `SPYXMLDataKind`) als Boolean

**Beschreibung**

Die Methode gibt true zurück, wenn das Objekt das übergeordnete Objekt von anderen `XMLData` Objekten dieser Art ist.

Verfügbar mit Typbibliothek Version 1.5

**Fehler**

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

InsertChild

**Siehe auch**

**Deklaration:** `InsertChild` (`pNewData` als `XMLData`)

**Beschreibung**

`InsertChild` fügt das neue Child-Element vor dem aktuellen Child-Element ein (Informationen zum Setzen des aktuellen Child-Elements finden Sie auch unter [XMLData.GetFirstChild](#), [XMLData.GetNextChild](#)).

**Fehler**

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert.

- 1505 Es wurde eine ungültige XMLData-Art definiert.
- 1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1507 Das Element kann keine Child-Elemente haben.
- 1512 Zyklische Einfügung - ein neues Datenelement ist bereits Teil des Dokuments.
- 1514 Es wurde eine ungültige XMLData-Art für diese Position definiert.
- 1900 Das Dokument darf nicht geändert werden.

**Beispiele**

Siehe `Using XMLData to modify document structure`.

**InsertChildAfter**

**Methode:** `InsertChildAfter` (Node als XMLData, NewData als XMLData)

**Beschreibung**

Fügt einen neuen XML-Node (definiert im zweiten Parameter) nach dem (mit dem ersten Parameter) angegebenen Node ein.

**Errors**

- 1500 Ungültiges Objekt.
- 1506 Ungültige XML-Input-Datei
- 1507 Es sind keine Child-Nodes zulässig
- 1510 Ungültiger Parameter.
- 1512 Das Child wurde bereits hinzugefügt
- 1514 Ungültige Art an dieser Stelle

**InsertChildBefore**

**Methode:** `InsertChildBefore` (Node als XMLData, NewData als XMLData)

**Beschreibung**

Fügt einen neuen XML-Node (definiert im zweiten Parameter) vor dem (im ersten Parameter) angegebenen Node ein.

**Errors**

- 1500 Ungültiges Objekt.
- 1506 Ungültige XML-Input-Datei
- 1507 Es sind keine Child-Nodes zulässig
- 1510 Ungültiger Parameter.
- 1512 Das Child wurde bereits hinzugefügt
- 1514 Ungültige Art an dieser Stelle

**IsSameNode****Siehe auch**

**Deklaration:** `IsSameNode` (`pNodeToCompare` als `XMLData`) als Boolean

**Beschreibung**

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

**Fehler**

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Kind

**Siehe auch**

**Deklaration:** [Kind](#) als [SPYXMLDataKind](#)

### Beschreibung

Art dieses `XMLData` Objekts. Diese Eigenschaft ist schreibgeschützt.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MayHaveChildren

**Siehe auch**

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

### Beschreibung

Gibt an, ob Child-Elemente zu diesem `XMLData` Objekt hinzugefügt werden dürfen. Diese Eigenschaft ist schreibgeschützt.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Name

**Siehe auch**

**Deklaration:** [Name](#) als String

### Beschreibung

Dient zum Bearbeiten und Abrufen des Namens des `XMLData` Objekts.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

**Siehe auch**

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

### Rückgabewert

Übergeordnetes Element als `XMLData` Objekt. Nichts (oder NULL), wenn es kein übergeordnetes Element gibt.

### Beschreibung

Diesem Element übergeordnetes Element. Diese Eigenschaft ist schreibgeschützt.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SetTextValueXMLEncoded

**Methode:** `SetTextValueXMLEncoded ( strVal als String )`

### Beschreibung

Definiert den kodierten Textwert der XML-Datei.

### Fehler

- 1500 Ungültiges Objekt.
- 1513 Keine Änderung zulässig.

TextValue

### Siehe auch

**Deklaration:** `TextValue` als String

### Beschreibung

Dient zur Bearbeitung und Abrufen des Textwerts dieses `XMLData` Objekts.

### Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

### Beispiele

Siehe „[ndern der Dokumentstruktur](#)“ in [Hilfe](#) von XMLData .

## Enumerationen

Dies ist eine Liste aller von der StyleVision API verwendeten Enumerationen. Wenn Ihre Skripting-Umgebung Enumerationen nicht unterstützt, verwenden Sie stattdessen die Zahlenwerte.

**ENUMApplicationStatus****Beschreibung**

Enumeration zur Definition des aktuellen Applikationsstatus.

**Mögliche Werte:**

|                                          |     |
|------------------------------------------|-----|
| eApplicationRunning                      | = 0 |
| eApplicationAfterLicenseCheck            | = 1 |
| eApplicationBeforeLicenseCheck           | = 2 |
| eApplicationConcurrentLicenseCheckFailed | = 3 |
| eApplicationProcessingCommandLine        | = 4 |



**ENUMAppOutputLine\_Severity****Beschreibung**

Enumerationswerte zur Kennzeichnung des Schweregrads einer AppOutputLine.

**Mögliche Werte:**

|                                          |      |
|------------------------------------------|------|
| eSeverity_Undefined                      | = -1 |
| eSeverity_Info                           | = 0  |
| eSeverity_Warning                        | = 1  |
| eSeverity_Error                          | = 2  |
| eSeverity_CriticalError                  | = 3  |
| eSeverity_Success                        | = 4  |
| eSeverity_Summary                        | = 5  |
| eSeverity_Progress                       | = 6  |
| eSeverity_DataEdit                       | = 7  |
| eSeverity_ParserInfo                     | = 8  |
| eSeverity_PossibleInconsistencyWarning   | = 9  |
| eSeverity_Message                        | = 10 |
| eSeverity_Document                       | = 11 |
| eSeverity_Rest                           | = 12 |
| eSeverity_NoSelect                       | = 13 |
| eSeverity_Select                         | = 14 |
| eSeverity_Autoinsertion                  | = 15 |
| eSeverity_GlobalResources_DefaultWarning | = 16 |

**ENUMAppOutputLine\_TextDecoration****Beschreibung**

Enumerationenwerte für die verschiedenen Arten der Textdarstellung einer AppOutputLine.

**Mögliche Werte:**

|                                         |      |
|-----------------------------------------|------|
| eTextDecorationDefault                  | = 0  |
| eTextDecorationBold                     | = 1  |
| eTextDecorationDebugValues              | = 2  |
| eTextDecorationDB_ObjectName            | = 3  |
| eTextDecorationDB_ObjectLink            | = 4  |
| eTextDecorationDB_ObjectKind            | = 5  |
| eTextDecorationDB_TimeoutValue          | = 6  |
| eTextDecorationFind_MatchingString      | = 7  |
| eTextDecorationValidation_Speclink      | = 8  |
| eTextDecorationValidation_ErrorPosition | = 9  |
| eTextDecorationValidation_UnkownParam   | = 10 |

**ENUMSchemaSourceType****Beschreibung**

Enumeration zur Definition des Quellschematyps: XML-Schema, DTD, DB, DB-Zellen, benutzerdefiniert oder XBRL.

**Mögliche Werte:**

|                            |     |
|----------------------------|-----|
| eSchemaSourceType_XSDorDTD | = 0 |
| eSchemaSourceType_DB       | = 1 |
| eSchemaSourceType_DBCell   | = 2 |
| eSchemaSourceType_User     | = 3 |
| eSchemaSourceType_XBRL     | = 4 |

**ENUMSchemaType****Beschreibung**

Enumeration zur Definition des Schematyps: W3C XML-Schema oder DTD.

**Mögliche Werte:**

|                      |     |
|----------------------|-----|
| eSchemaTypeW3CSchema | = 0 |
| eSchemaTypeDTD       | = 1 |

## Applikations-API für Java

Die in diesem Abschnitt (Applikation API für Java) beschriebenen Objekte werden ab v2012 nicht mehr verwendet.

Informationen zum Aufrufen der Application API von Java-Code aus finden Sie im Abschnitt: [Programmiersprachen | Java](#).

Die Applikations-API in Java hat eine Benutzeroberfläche, die auf Java-Klassen aufbaut, wobei jede dieser Klassen einem Objekt in der [Applikations-API](#) entspricht. Entwickler können mit Hilfe dieser Java-Klassen mit der COM API interagieren. Diese Klassen sind unten aufgelistet und in den Unterabschnitten näher beschrieben. Eine Beschreibung der [Applikations-API](#)-Objekte selbst finden Sie in der [Dokumentation zur Applikations-API](#). Beachten Sie, dass einige API-Funktionen nur in Skripting-Umgebungen zur Verfügung stehen; diese wurden daher nicht nach Java portiert.

### Java-Klassen

- [SpyApplication](#)
- [SpyProject](#)
  - [SpyProjectItems](#)
  - [SpyProjectItem](#)
- [SpyDocuments](#)
- [SpyDoc](#)
  - [SpyAuthenticView](#)
  - [SpyAuthenticRange](#)
  - [SpyDocEditView](#)
  - [SpyDocEditSelection](#)
  - [SpyGridView](#)
  - [SpyTextView](#)
  - [SpyXMLData](#)
- [SpyDialogs](#)
  - [SpyCodeGeneratorDlg](#)
  - [SpyDTDSchemaGeneratorDlg](#)
  - [SpyFileSelectionDlg](#)
  - [SpyFindInFilesDlg](#)
  - [SpyGenerateSampleXMLDlg](#)
  - [SpySchemaDocumentationDlg](#)
  - [SpyWSDL20DocumentationDlg](#)
  - [SpyWSDLDocumentationDlg](#)
  - [SpyXBRLDocumentationDlg](#)
- [SpyDatabaseConnection](#)
- [SpyElementList](#)
  - [SpyElementListItem](#)
- [SpyExportSettings](#)
- [SpyFindInFilesResults](#)
  - [SpyFindInFilesResult](#)
  - [SpyFindInFilesMatch](#)
- [SpyTextImportExportSettings](#)

### Implementierung von COM-Eigenschaften in Java

Die Eigenschaften in Java wurden so definiert, dass Sie sowohl eine `set` als auch eine `get` Methode enthalten (`set` wenn dies die COM-Implementierung zulässt). So enthält z.B. die

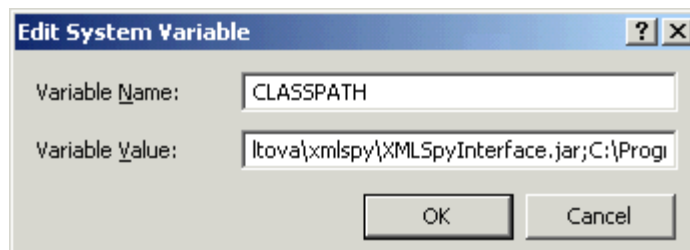
COM-Klasse `Document` die Eigenschaft `GridView`. In Java heißt die Methode `SpyDoc` und die Eigenschaft ist als eine `GetGridView` Methode definiert.

Überprüfen Sie bei Kompilierproblemen die folgenden Punkte:

- Die `xmlspylib.dll` muss in `..\windows\system32` verfügbar sein.
- Die Datei `XMLSpyInterface.jar` muss in die `ClassPath`-Umgebungsvariable eingesetzt werden.

#### Setzen der `ClassPath` Variable in Windows XP

1. Klicken Sie auf **Start | Einstellungen | Systemsteuerung | System | Erweitert | Umgebungsvariablen**. Daraufhin wird das Dialogfeld "Umgebungsvariablen" geöffnet.
2. Wenn in der Gruppe der Systemvariablen bereits ein `ClassPath` Eintrag vorhanden ist, wählen Sie den Eintrag `ClassPath` aus und klicken Sie auf die Schaltfläche **Bearbeiten**. Bearbeiten Sie den Pfad zu: "`C:\Program Files\Altova\xmlspy\XMLSpyInterface.jar`".



Wenn in der Gruppe der Systemvariablen noch kein Eintrag `ClassPath` vorhanden ist, klicken Sie auf die Schaltfläche **Neu**. Daraufhin erscheint das Dialogfeld "Neue Systemvariable". Geben Sie als Name der Variable `CLASSPATH` ein und "`C:\Program Files\Altova\xmlspy\XMLSpyInterface.jar`" als die `ClassPath` Variable (ändern Sie gegebenenfalls den Pfad zu ihrer Installation).

## Beispielquellcode

Der Befehlsparameter "SpyDoc doc = app.GetDocuments().OpenFile(...)" muss für Ihre Umgebung angepasst werden.

Das Beispiel führt folgende Schritte durch:

- Es startet eine neue XMLSpy-Instanz
- Es öffnet die Datei Datasheet.xml (ändern Sie den Pfad hier...)
- Es wechselt in die Grid-Ansicht
- Es hängt ein neues Child-Element namens "NewChild" mit dem Textwertelement "NewValuE" an das Root-Element an
- Es überprüft, ob das Dokument gültig ist und gibt eine Meldung an die Java-Konsole aus
- Es schließt die XMLSpy-Applikation und gibt diese frei

```
import XMLSpyInterface.*;

public class TestSpyInterface
{
 public TestSpyInterface() {}

 public static void main(String[] args)
 {
 SpyApplication app = null;
 SpyDoc oDoc = null;
 SpyXMLData oData = null;
 SpyXMLData oNewChild = null;

 try
 {
 app = new SpyApplication();
 app.ShowApplication(true);

 oDoc = app.GetDocuments().OpenFile("C:\\FilePath\\OrgChart.xml",
 true);

 // OrgChart.xml is in the folder C:\Documents and
 // Settings\<username>\My Documents\Altova\XMLSpy\. The filepath should
 // be in
 // the form: C:\\Documents and
 // Settings\\Username\\Folder\\Filename.xml

 if (oDoc != null)
 {
 oDoc.SwitchViewMode(SPYViewModes.spyViewGrid);
 oData = oDoc.GetRootElement();
 oNewChild = oDoc.CreateChild(SPYXMLDataKind.spyXMLDataElement);

 oNewChild.SetName("NewChild");
 oNewChild.SetTextValue("newValuE");
 oData.AppendChild(oNewChild);

 if (oDoc.IsValid() == false)
 {
 // is to be expected after above insertion
 System.out.println("!!!!!! validation error: " +
 oDoc.GetErrorString());
 System.out.println("!!!!!! validation error: " +
 oDoc.GetErrorPos());
 System.out.println("!!!!!! validation error: " +
 oDoc.GetBadData());
 }
 }
 }
 }
}
```

```
 }
 }

 app.Quit();
}
finally
{
 // Free any allocated resources by calling ReleaseInstance().
 if (oNewChild != null)
 oNewChild.ReleaseInstance();

 if (oData != null)
 oData.ReleaseInstance();

 if (oDoc != null)
 oDoc.ReleaseInstance();

 if (app != null)
 app.ReleaseInstance();
}
}
```

Wenn Sie Probleme haben, dieses Beispiel zu kompilieren, versuchen Sie bitte in der Befehlszeile (**Start | Ausführen | cmd**) die folgenden Befehle. Stellen Sie vorher sicher, dass Sie sich gerade in dem Ordner befinden, der die Java-Beispieldatei enthält.

#### Kompilierung

```
javac -classpath c:\IhrPfadhier\XMLSpyInterface.jar testspyinterface.java
```

#### Ausführung

```
java -classpath c:\IhrPfadhier\XMLSpyInterface.jar testspyinterface
```



## SpyApplication

```
public class SpyApplication
{
 public void ReleaseInstance();
 public void ShowApplication(boolean bShow);
 public void Quit();
 public void AddMacroMenuItem(String sMacro, String sDisplayText);
 public void ClearMacroMenu();
 public SpyDoc GetActiveDocument();
 public SpyProject GetCurrentProject();
 public SpyDocuments GetDocuments();
 public SpyElementList GetDatabaseImportElementList(SpyDatabaseConnection
oImportSettings);
 public SpyDatabaseConnection GetDatabaseSettings();
 public SpyElementList GetDatabaseTables(SpyDatabaseConnection
oImportSettings);
 public SpyExportSettings GetExportSettings();
 public SpyElementList GetTextImportElementList(SpyTextImportExportSettings
oImportSettings);
 public SpyTextImportExportSettings GetTextImportExportSettings();
 public SpyDoc ImportFromDatabase(SpyDatabaseConnection oImportSettings,
SpyElementList oElementList);
 public SpyDoc ImportFromSchema(SpyDatabaseConnection oImportSettings,
String strTable, SpyDoc oSchemaDoc);
 public SpyDoc ImportFromText(SpyTextImportExportSettings oImportSettings,
SpyElementList oElementList);
 public SpyDoc ImportFromWord(String sFile);
 public void NewProject(String sPath, boolean bDiscardCurrent);
 public void OpenProject(String sPath, boolean bDiscardCurrent, boolean
bDialog);
 public long ShowForm(String sName);
 public void URLDelete(String sURL, String sUser, String sPassword);
 public void URLMakeDirectory(String sURL, String sUser, String sPassword);

 public int GetWarningNumber();
 public String GetWarningText();

 // since Version 2004R4
 public SpyApplication GetApplication();
 public SpyApplication GetParent();
 public SpyDialogs GetDialogs();
 public boolean GetVisible();
 public void SetVisible(boolean i_bVisibility);
 public long GetWindowHandle();

 public void ReloadSettings();
 public SpyFindInFilesResults FindInFiles(SpyFindInFilesDlg dlgSettings);
 public boolean ShowFindInFiles(SpyFindInFilesDlg dlgSettings);
 public void Selection(String sVal);

 public long Status();
 public int MajorVersion();
 public int MinorVersion();
 public String Edition();
 public boolean IsAPISupported();
 public long ServicePackVersion();
 public void CreateXMLSchemaFromDBStructure(SpyDatabaseConnection
oConnection, SpyElementList oTables);
}
```

## SpyCodeGeneratorDlg

Steht nur in der Enterprise Edition zur Verfügung. Bei Aufruf über eine andere Version wird eine Fehlermeldung ausgegeben.

```
// since version 2004R4
public class SpyCodeGeneratorDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public SpyDialogs GetParent();
 public long GetProgrammingLanguage();
 public void SetProgrammingLanguage(long i_eVal);
 public String GetTemplateFileName();
 public void SetTemplateFileName(String i_strVal);
 public String GetOutputPath();
 public void SetOutputPath(String i_strVal);
 public long GetOutputPathDialogAction();
 public void SetOutputPathDialogAction(long i_eVal);
 public long GetPropertySheetDialogAction();
 public void SetPropertySheetDialogAction(long i_eVal);
 public long GetOutputResultDialogAction();
 public void SetOutputResultDialogAction(long i_eVal);
 public long GetCPPSettings_DOMType();
 public void SetCPPSettings_DOMType(long i_eVal);
 public long GetCPPSettings_LibraryType();
 public void SetCPPSettings_LibraryType(long i_eVal);
 public boolean GetCPPSettings_UseMFC();
 public void SetCPPSettings_UseMFC(boolean i_bVal);
 public long GetCSharpSettings_ProjectType();
 public void SetCSharpSettings_ProjectType(long i_eVal);
}
```

## SpyDatabaseConnection

```
public class SpyDatabaseConnection
{
 public void ReleaseInstance();
 public String GetADOConnection();
 public void SetADOConnection(String sValue);
 public boolean GetAsAttributes();
 public void SetAsAttributes(boolean bValue);
 public boolean GetCreateMissingTables();
 public void SetCreateMissingTables(boolean bValue);
 public boolean GetCreateNew();
 public void SetCreateNew(boolean bValue);
 public boolean GetExcludeKeys();
 public void SetExcludeKeys(boolean bValue);
 public String GetFile();
 public void SetFile(String sValue);
 public boolean GetIncludeEmptyElements();
 public void SetIncludeEmptyElements(boolean bValue);
 public long GetNumberDateTimeFormat();
 public void SetNumberDateTimeFormat(long nValue);
 public String GetODBCConnection();
 public void SetODBCConnection(String sValue);
 public String GetSQLSelect();
 public void SetSQLSelect(String sValue);
 public long GetTextFieldLen();
 public void SetTextFieldLen(long nValue);

 // since version 2004R4
 public long GetDatabaseKind();
 public void SetDatabaseKind(long nValue);

 // since version 2008R2
 public boolean GetCommentIncluded();
 public void SetCommentIncluded(boolean bValue);
 public String GetNullReplacement();
 public void SetNullReplacement(String sValue);
 public String GetDatabaseSchema();
 public void SetDatabaseSchema(String sValue);

 // since version 2010r3
 public boolean GetPrimaryKeys();
 public void SetPrimaryKeys(boolean bValue)
 public boolean GetForeignKeys();
 public void SetForeignKeys(boolean bValue)
 public boolean GetUniqueKeys();
 public void SetUniqueKeys(boolean bValue)
 public long GetSchemaExtensionType();
 public void SetSchemaExtensionType(long nValue)
 public long GetSchemaFormat();
 public void SetSchemaFormat(long nValue)
 public long GetImportColumnsType();
 public void SetImportColumnsType(long nValue)
}
```

## SpyDialogs

```
// Since version 2004R4
public class SpyDialogs
{
 public SpyApplication GetApplication();
 public SpyApplication GetParent();
 public SpyCodeGeneratorDlg GetCodeGeneratorDlg();
 public SpyFileSelectionDlg GetFileSelectionDlg();
 public SpySchemaDocumentationDlg GetSchemaDocumentationDlg();
 public SpyGenerateSampleXMLDlg GetGenerateSampleXMLDlg();
 public SpyDTDSchemaGeneratorDlg GetDTDSchemaGeneratorDlg();
 public SpyFindInFilesDlg GetFindInFilesDlg();
 public SpyWSDLDocumentationDlg GetWSDLDocumentationDlg();

 // Since version 2010
 public SpyWSDL20DocumentationDlg GetWSDL20DocumentationDlg();
 public SpyXBRLDocumentationDlg GetXBRLDocumentationDlg();
}
```

## SpyDoc

```
public class SpyDoc
{
 public void ReleaseInstance();
 public void SetEncoding(String strEncoding);
 public void SetPathName(String strPath);
 public String GetPathName();
 public String GetTitle();
 public boolean IsModified();
 public void Save();
 public void Close(boolean bDiscardChanges);
 public void UpdateViews();
 public long GetCurrentViewMode();
 public boolean SwitchViewMode(long nMode);
 public SpyGridView GetGridView();
 public void SetActiveDocument();
 public void StartChanges();
 public void EndChanges();
 public void TransformXSL();
 public void AssignDTD(String sDTDFile, boolean bDialog);
 public void AssignSchema(String sSchemaFile, boolean bDialog);
 public void AssignXSL(String sXSLFile, boolean bDialog);
 public void ConvertDTDOrSchema(long nFormat, long nFrequentElements);
 public SpyXMLData CreateChild(long nKind);
 public void CreateSchemaDiagram(long nKind, String sName, String sFile);
 public SpyDocEditView GetDocEditView();
 public void ExportToDatabase(SpyXMLData oFromChild, SpyExportSettings
oExportSettings, SpyDatabaseConnection oDatabaseConnection);
 public void ExportToText(SpyXMLData oFromChild, SpyExportSettings
oExportSettings, SpyTextImportExportSettings oTextSettings);
 public void GenerateDTDOrSchema(long nFormat, int nValuesList, long
nDetection, long nFrequentElements);
 public SpyElementList GetExportElementList(SpyXMLData oFromChild,
SpyExportSettings oExportSettings);
 public SpyXMLData GetRootElement();
 public String SaveInString(SpyXMLData oData, boolean bMarked);
 public void SaveToURL(String sUrl, String sUser, String sPassword);
 public String GetErrorString(); // See IsValid() or IsWellFormed()
 public int GetErrorPos(); // See IsValid() or IsWellFormed()
 public SpyXMLData GetBadData(); // See IsValid() or IsWellFormed()
 public boolean IsValid();
 public boolean IsWellFormed(SpyXMLData oData, boolean bWithChildren);

 // Since version 2004R3
 public SpyAuthenticView GetAuthenticView()

 // Since version 2004R4
 public SpyApplication GetApplication();
 public SpyDocuments GetParent();
 public String GetFullName();
 public void SetFullName(String i_strName);
 public String GetName();
 public String GetPath();
 public boolean GetSaved();
 public void SaveAs(String i_strFileNameOrPath);
 public String GetEncoding();
 public SpyXMLData GetDataRoot();
 public void GenerateProgramCode(SpyCodeGeneratorDlg i_dlg);
 public void AssignXSLFO(String i_strFile, boolean i_bUseDialog);
 public void TransformXSLFO();
 public void GenerateSchemaDocumentation(SpySchemaDocumentationDlg i_dlg);
}
```

```
public void ExecuteXQuery(String i_strXMLSourceFile);
public void SetExternalIsValid(boolean bIsValid);
public SpyDoc GenerateSampleXML(SpyGenerateSampleXMLDlg ipGenerateXMLDlg);
public boolean UpdateXMLData();
public String GetAsXMLString();
public void SetAsXMLString(String newVal);
public SpyDoc GenerateDTDOrSchemaEx(SpyDTDSchemaGeneratorDlg
ipDTDSchemaGeneratorDlg);
public SpyDoc ConvertDTDOrSchemaEx(long nFormat, long nFrequentElements,
String sOutputPath, long nOutputPathDialogAction);
public SpyTextView GetTextView();
public String[] GetSuggestions();
public void SetSuggestions(String[] aList);
public void SetSelection(String sVal);

// Since version 2009
public void GenerateWSDLDocumentation(SpyWSDLDocumentationDlg
ipWSDLDocumenationDlg);
public void TransformXSLEx(long nDialogAction);

// Since version 2010
public void GenerateWSDL20Documentation(SpyWSDL20DocumentationDlg
ipWSD20DocumenationDlg);
public void GenerateXBRLDocumentation(SpyXBRLDocumentationDlg
ipXBRLDocumentationDlg);
public SpyDoc ConvertToWSDL20(String sFilePath, boolean bShowDialogs);

// Since version 2010r3
public String CreateDBStructureFromXMLSchema(SpyDatabaseConnection
oConnection, SpyElementList oTables, boolean bDropTableWithExistingName);
public SpyElementList GetDBStructureList(SpyDatabaseConnection oConnection
);
}
```

## SpyDocuments

```
public class SpyDocuments
{
 public void ReleaseInstance\(\);
 public long Count();
 public SpyDoc GetItem(long nNo);
 public SpyDoc NewFile(String strFile, String strType);
 public SpyDoc NewFileFromText(String nSource, String strType);
 public SpyDoc OpenFile(String sPath, boolean bDialog);
 public SpyDoc OpenURL(String sUrl, long nURLType, long nLoading, String
sUser, String sPassword);
 public SpyDoc OpenURLDialog(String sURL, long nURLType, long nLoading,
String sUser, String sPassword);
 // Since version 2011r2
 public SpyDoc NewAuthenticFile(String strSPSPPath, String strXMLPath);
 public SpyDoc OpenAuthenticFile(String strSPSPPath, String strXMLPath);
}
```

**SpyDTDSchemaGeneratorDlg**

```
public class SpyDTDSchemaGeneratorDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public long GetDTDSchemaFormat();
 public void SetDTDSchemaFormat(long newVal);
 public short GetValueList();
 public void SetValueList(short newVal);
 public long GetTypeDetection();
 public void SetTypeDetection(long newVal);
 public long GetFrequentElements();
 public void SetFrequentElements(long newVal);
 public boolean GetMergeAllEqualNamed();
 public void SetMergeAllEqualNamed(boolean newVal);
 public boolean GetResolveEntities();
 public void SetResolveEntities(boolean newVal);
 public long GetAttributeTypeDefinition();
 public void SetAttributeTypeDefinition(long newVal);
 public boolean GetGlobalAttributes();
 public void SetGlobalAttributes(boolean newVal);
 public boolean GetOnlyStringEnums();
 public void SetOnlyStringEnums(boolean newVal);
 public long GetMaxEnumLength();
 public void SetMaxEnumLength(long newVal);
 public String GetOutputPath();
 public void SetOutputPath(String newVal);
 public long GetOutputPathDialogAction();
 public void SetOutputPathDialogAction(long newVal);
}
```



## SpyElementList

```
public class SpyElementList
{
 public void ReleaseInstance\(\);
 public long GetCount();
 public SpyElementListItem GetItem(long nIndex);
 public void RemoveElement(long nIndex);
}
```

**SpyElementListItem**

```
public class SpyElementListItem
{
 public void ReleaseInstance();
 public long GetElementKind();
 public void SetElementKind(long nKind);
 public long GetFieldCount();
 public String GetName();
 public long GetRecordCount();
}
```

## SpyExportSettings

```
public class SpyExportSettings
{
 public void ReleaseInstance();
 public boolean GetCreateKeys();
 public void SetCreateKeys(boolean bValue);
 public SpyElementList GetElementList();
 public void SetElementList(SpyElementList obj);
 public boolean GetEntitiesToText ();
 public void SetEntitiesToText(boolean bValue);
 public boolean GetExportAllElements();
 public void SetExportAllElements(boolean bValue);
 public boolean GetFromAttributes();
 public void SetFromAttributes(boolean bValue);
 public boolean GetFromSingleSubElements();
 public void SetFromSingleSubElements(boolean bValue);
 public boolean GetFromTextValues();
 public void SetFromTextValues(boolean bValue);
 public boolean GetIndependentPrimaryKey();
 public void SetIndependentPrimaryKey(boolean bValue);
 public long GetNamespace();
 public void SetNamespace(long nValue);
 public int GetSubLevelLimit();
 public void SetSubLevelLimit(int nValue);
}
```

**SpyFileSelectionDlg**

```
// Since version 2004R4
public class SpyFileSelectionDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public SpyDialogs GetParent();
 public String GetFullName();
 public void SetFullName(String i_strName);
 public long GetDialogAction();
 public void SetDialogAction(long i_eAction);
}
```

**SpyFindInFilesDlg**

```
public class SpyFindInFilesDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public String GetFind();
 public void SetFind(String sNewVal);
 public boolean GetRegularExpression();
 public void SetRegularExpression(boolean bNewVal);
 public String GetReplace();
 public void SetReplace(String sNewVal);
 public boolean GetReplaceOnDisk();
 public void SetReplaceOnDisk(boolean bNewVal);
 public boolean GetDoReplace();
 public void SetDoReplace(boolean bNewVal);
 public boolean GetMatchWholeWord();
 public void SetMatchWholeWord(boolean bNewVal);
 public boolean GetMatchCase();
 public void SetMatchCase(boolean bNewVal);
 public long GetSearchLocation();
 public void SetSearchLocation(long nPosition);
 public String GetStartFolder();
 public void SetStartFolder(String sNewVal);
 public boolean GetIncludeSubfolders();
 public void SetIncludeSubfolders(boolean bNewVal);
 public boolean GetSearchInProjectFilesDoExternal();
 public void SetSearchInProjectFilesDoExternal(boolean bNewVal);
 public String GetFileExtension();
 public void SetFileExtension(String sNewVal);
 public boolean GetAdvancedXMLSearch();
 public void SetAdvancedXMLSearch(boolean bNewVal);
 public boolean GetXMLElementNames();
 public void SetXMLElementNames(boolean bNewVal);
 public boolean GetXMLElementContents();
 public void SetXMLElementContents(boolean bNewVal);
 public boolean GetXMLAttributeNames();
 public void SetXMLAttributeNames(boolean bNewVal);
 public boolean GetXMLAttributeContents();
 public void SetXMLAttributeContents(boolean bNewVal);
 public boolean GetXMLComments();
 public void SetXMLComments(boolean bNewVal);
 public boolean GetXMLCDATA();
 public void SetXMLCDATA(boolean bNewVal);
 public boolean GetXMLPI();
 public void SetXMLPI(boolean bNewVal);
 public boolean GetXMLRest();
 public void SetXMLRest(boolean bNewVal);
 public boolean GetShowResult();
 public void SetShowResult(boolean bNewVal);
}
```

**SpyFindInFilesMatch**

```
public class SpyFindInFilesMatch
{
 public void ReleaseInstance\(\);
 public long Line();
 public long Position();
 public long Length();
 public String LineText();
 public boolean Replaced();
}
```

**SpyFindInFilesResult**

```
public class SpyFindInFilesResult
{
 public void ReleaseInstance\(\);
 public long Count();
 public SpyFindInFilesMatch GetItem(long nNo);
 public String GetPath();
 public SpyDoc GetDocument();
}
```

**SpyFindInFilesResults**

```
public class SpyFindInFilesResults
{
 public void ReleaseInstance\(\);
 public long Count();
 public SpyFindInFilesResult GetItem(long nNo);
}
```



**SpyGenerateSampleXMLDlg**

```
public class SpyGenerateSampleXMLDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public boolean GetNonMandatoryAttributes();
 public void SetNonMandatoryAttributes(boolean newVal);
 public boolean GetNonMandatoryElements();
 public void SetNonMandatoryElements(boolean newVal);
 public boolean GetTakeFirstChoice();
 public void SetTakeFirstChoice(boolean newVal);
 public long GetRepeatCount();
 public void SetRepeatCount(long newVal);
 public boolean GetFillWithSampleData();
 public void SetFillWithSampleData(boolean newVal);
 public boolean GetFillElementsWithSampleData();
 public void SetFillElementsWithSampleData(boolean newVal);
 public boolean GetFillAttributesWithSampleData();
 public void SetFillAttributesWithSampleData(boolean newVal);
 public boolean GetContentOfNillableElementsIsNonMandatory();
 public void SetContentOfNillableElementsIsNonMandatory(boolean newVal);
 public boolean GetTryToUseNonAbstractTypes();
 public void SetTryToUseNonAbstractTypes(boolean newVal);
 public long GetOptimization();
 public void SetOptimization(long newVal);
 public long GetSchemaOrDTDAssignment();
 public void SetSchemaOrDTDAssignment(long newVal);
 public String GetLocalNameOfRootElement();
 public void SetLocalNameOfRootElement(String newVal);
 public String GetNamespaceURIOfRootElement();
 public void SetNamespaceURIOfRootElement(String newVal);
 public long GetOptionsDialogAction();
 public void SetOptionsDialogAction(long newVal);
}
```

## SpyGridView

```
public class SpyGridView
{
 public void ReleaseInstance\(\);
 public SpyXMLData GetCurrentFocus();
 public void Deselect(SpyXMLData oData);
 public boolean GetIsVisible();
 public void Select(SpyXMLData oData);
 public void SetFocus(SpyXMLData oData);
}
```

## SpyProject

```
public class SpyProject
{
 public void ReleaseInstance();
 public void CloseProject(boolean bDiscardChanges, boolean bCloseFiles,
 boolean bDialog);
 public String GetProjectFile();
 public void SetProjectFile(String sFile);
 public SpyProjectItems GetRootItems();
 public void SaveProject();
 public void SaveProjectAs(String sPath, boolean bDialog);
}
```

## SpyProjectItem

```
public class SpyProjectItem
{
 public void ReleaseInstance();
 public SpyProjectItems GetChildItems();
 public String GetFileExtensions();
 public void SetFileExtensions(String sExtensions);
 public long GetItemType();
 public String GetName();
 public SpyDoc Open();
 public SpyProjectItem GetParentItem();
 public String GetPath();
 public String GetValidateWith();
 public void SetValidateWith(String sVal);
 public String GetXMLForXSLTransformation();
 public void SetXMLForXSLTransformation(String sVal);
 public String GetXSLForXMLTransformation();
 public void SetXSLForXMLTransformation(String sVal);
 public String GetXSLTransformationFileExtension();
 public void SetXSLTransformationFileExtension(String sVal);
 public String GetXSLTransformationFolder();
 public void SetXSLTransformationFolder(String sVal);
}
```

## SpyProjectItems

```
public class SpyProjectItems
{
 public void ReleaseInstance();
 public void AddFile(String sPath);
 public void AddFolder(String sName);
 public void AddURL(String sURL, long nURLType, String sUser, String
sPassword, boolean bSave);
 public long Count();
 public SpyProjectItem GetItem(long nNumber);
 public void RemoveItem(SpyProjectItem oItemToRemove);
}
```

**SpySchemaDocumentationDlg**

```
// Since version 2004R4
public class SpySchemaDocumentationDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public SpyDialogs GetParent();

 public String GetOutputFile();
 public void SetOutputFile(String i_strVal);
 public long GetOutputFormat();
 public void SetOutputFormat(long i_eVal);

 public boolean GetShowResult();
 public void SetShowResult(boolean i_bVal);
 public long GetOptionsDialogAction();
 public void SetOptionsDialogAction(long i_eVal);
 public long GetOutputFileDialogAction();
 public void SetOutputFileDialogAction(long i_eVal);
 public boolean GetShowProgressBar();
 public void SetShowProgressBar(boolean i_bVal);

 public void IncludeAll(boolean i_bInclude);
 public boolean GetIncludeIndex();
 public void SetIncludeIndex(boolean i_bVal);
 public boolean GetIncludeGlobalElements();
 public void SetIncludeGlobalElements(boolean i_bVal);
 public boolean GetIncludeLocalElements();
 public void SetIncludeLocalElements(boolean i_bVal);
 public boolean GetIncludeGroups();
 public void SetIncludeGroups(boolean i_bVal);
 public boolean GetIncludeComplexTypes();
 public void SetIncludeComplexTypes(boolean i_bVal);
 public boolean GetIncludeSimpleTypes();
 public void SetIncludeSimpleTypes(boolean i_bVal);
 public boolean GetIncludeAttributeGroups();
 public void SetIncludeAttributeGroups(boolean i_bVal);
 public boolean GetIncludeRedefines();
 public void SetIncludeRedefines(boolean i_bVal);

 public void AllDetails(boolean i_bDetailsOn);
 public boolean GetShowDiagram();
 public void SetShowDiagram(boolean i_bVal);
 public boolean GetShowNamespace();
 public void SetShowNamespace(boolean i_bVal);
 public boolean GetShowType();
 public void SetShowType(boolean i_bVal);
 public boolean GetShowChildren();
 public void SetShowChildren(boolean i_bVal);
 public boolean GetShowUsedBy();
 public void SetShowUsedBy(boolean i_bVal);
 public boolean GetShowProperties();
 public void SetShowProperties(boolean i_bVal);
 public boolean GetShowSingleFacets();
 public void SetShowSingleFacets(boolean i_bVal);
 public boolean GetShowPatterns();
 public void SetShowPatterns(boolean i_bVal);
 public boolean GetShowEnumerations();
 public void SetShowEnumerations(boolean i_bVal);
 public boolean GetShowAttributes();
 public void SetShowAttributes(boolean i_bVal);
}
```

```
public boolean GetShowIdentityConstraints();
public void SetShowIdentityConstraints(boolean i_bVal);
public boolean GetShowAnnotations();
public void SetShowAnnotations(boolean i_bVal);
public boolean GetShowSourceCode();
public void SetShowSourceCode(boolean i_bVal);

// Since version 2009
public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams(boolean i_bVal);
public long GetDiagramFormat();
public void SetDiagramFormat(long i_nVal);
public boolean GetIncludeGlobalAttributes();
public void SetIncludeGlobalAttributes(boolean i_bVal);
public boolean GetIncludeLocalAttributes();
public void SetIncludeLocalAttributes(boolean i_bVal);
public boolean GetIncludeReferencedSchemas();
public void SetIncludeReferencedSchemas(boolean i_bVal);
public boolean GetMultipleOutputFiles();
public void SetMultipleOutputFiles(boolean i_bVal);

// Since version 2010
public boolean GetEmbedCSSInHTML();
public void SetEmbedCSSInHTML(boolean i_bVal);
public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder(boolean i_bVal);

// Since version 2010r3
public boolean GetGenerateRelativeLinks();
public void SetGenerateRelativeLinks(boolean i_bVal);

// Since version 2011r2
public boolean GetUseFixedDesign();
public void SetUseFixedDesign(boolean i_bVal);
public String GetSPSFile();
public void SetSPSFile(String i_strVal);
}
```

**SpyTextImportExportSettings**

```
public class SpyTextImportExportSettings
{
 public void ReleaseInstance();
 public String GetDestinationFolder();
 public void SetDestinationFolder(String sVal);
 public long GetEnclosingCharacter();
 public void SetEnclosingCharacter(long nEnclosing);
 public String GetEncoding();
 public void SetEncoding(String sVal);
 public long GetEncodingByteOrder();
 public void SetEncodingByteOrder(long nByteOrder);
 public long GetFieldDelimiter();
 public void SetFieldDelimiter(long nDelimiter);
 public String GetFileExtension ();
 public void SetFileExtension(String sVal);
 public boolean GetHeaderRow();
 public void SetHeaderRow(boolean bVal);
 public String GetImportFile();
 public void SetImportFile(String sVal);
}
```



## SpyTextView

```
public class SpyTextView
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public SpyDoc GetParent();
 public long LineFromPosition(long nCharPos);
 public long PositionFromLine(long nLine);
 public long LineLength(long nLine);
 public String GetSelText();
 public void SetSelText(String sText);
 public String GetRangeText(long nPosFrom, long nPosTill);
 public void ReplaceText(long nPosFrom, long nPosTill, String sText);
 public void MoveCaret(long nDiff);
 public void GoToLineChar(long nLine, long nChar);
 public void SelectText(long nPosFrom, long nPosTill);
 public long GetSelectionStart();
 public void SetSelectionStart(long nNewVal);
 public long GetSelectionEnd();
 public void SetSelectionEnd(long nNewVal);
 public String GetText();
 public void SetText(String sText);
 public long LineCount();
 public long Length();
}
```

**SpyWSDL20DocumentationDlg**

```
// Since version 2010
public class SpyWSDL20DocumentationDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();

 public long GetOptionsDialogAction();
 public void SetOptionsDialogAction(long nNewVal);

 public long GetOutputFileDialogAction();
 public void SetOutputFileDialogAction(long nNewVal);

 public boolean GetShowProgressBar();
 public void SetShowProgressBar(boolean bNewVal);

 public String GetOutputFile();
 public void SetOutputFile(String sNewVal);

 public long GetOutputFormat();
 public void SetOutputFormat(long nNewVal);

 public boolean GetMultipleOutputFiles();
 public void SetMultipleOutputFiles(boolean bNewVal);

 public boolean GetEmbedCSSInHTML();
 public void SetEmbedCSSInHTML(boolean bNewVal);

 public long GetDiagramFormat();
 public void SetDiagramFormat(long nNewVal);

 public boolean GetEmbedDiagrams();
 public void SetEmbedDiagrams(boolean bNewVal);

 public boolean GetCreateDiagramsFolder();
 public void SetCreateDiagramsFolder(boolean bNewVal);

 public boolean GetShowResult();
 public void SetShowResult(boolean bNewVal);

 public void IncludeAll(boolean bNewVal);
 public void AllDetails(boolean bNewVal);

 public boolean GetIncludeOverview();
 public void SetIncludeOverview(boolean bNewVal);

 public boolean GetIncludeService();
 public void SetIncludeService(boolean bNewVal);

 public boolean GetIncludeBinding();
 public void SetIncludeBinding(boolean bNewVal);

 public boolean GetIncludeInterface();
 public void SetIncludeInterface(boolean bNewVal);

 public boolean GetIncludeTypes();
}
```

```
public void SetIncludeTypes(boolean bNewVal);

public boolean GetIncludeImportedWSDLFiles();
public void SetIncludeImportedWSDLFiles(boolean bNewVal);

public boolean GetShowServiceDiagram();
public void SetShowServiceDiagram(boolean bNewVal);

public boolean GetShowBindingDiagram();
public void SetShowBindingDiagram(boolean bNewVal);

public boolean GetShowInterfaceDiagram();
public void SetShowInterfaceDiagram(boolean bNewVal);

public boolean GetShowTypesDiagram();
public void SetShowTypesDiagram(boolean bNewVal);

public boolean GetShowEndpoint();
public void SetShowEndpoint(boolean bNewVal);

public boolean GetShowSourceCode();
public void SetShowSourceCode(boolean bNewVal);

public boolean GetShowExtensibility();
public void SetShowExtensibility(boolean bNewVal);

public boolean GetShowUsedBy();
public void SetShowUsedBy(boolean bNewVal);

public boolean GetShowOperation();
public void SetShowOperation(boolean bNewVal);

public boolean GetShowFault();
public void SetShowFault(boolean bNewVal);

// Since version 2011r2
public boolean GetUseFixedDesign();
public void SetUseFixedDesign(boolean i_bVal);

public String GetSPSFile();
public void SetSPSFile(String i_strVal);

}
```

**SpyWSDLDocumentationDlg**

```
// Since version 2008r2spl
public class SpyWSDLDocumentationDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();

 public String GetOutputFile();
 public void SetOutputFile(String sNewVal);

 public long GetOutputFileDialogAction();
 public void SetOutputFileDialogAction(long nNewVal);

 public long GetOptionsDialogAction();
 public void SetOptionsDialogAction(long nNewVal);

 public boolean GetShowProgressBar();
 public void SetShowProgressBar(boolean bNewVal);

 public boolean GetShowResult();
 public void SetShowResult(boolean bNewVal);

 public long GetOutputFormat();
 public void SetOutputFormat(long nNewVal);

 public boolean GetEmbedDiagrams();
 public void SetEmbedDiagrams(boolean bNewVal);

 public long GetDiagramFormat();
 public void SetDiagramFormat(long nNewVal);

 public boolean GetMultipleOutputFiles();
 public void SetMultipleOutputFiles(boolean bNewVal);

 public void IncludeAll(boolean bNewVal);

 public boolean GetIncludeBinding();
 public void SetIncludeBinding(boolean bNewVal);

 public boolean GetIncludeImportedWSDLFiles();
 public void SetIncludeImportedWSDLFiles(boolean bNewVal);

 public boolean GetIncludeMessages();
 public void SetIncludeMessages(boolean bNewVal);

 public boolean GetIncludeOverview();
 public void SetIncludeOverview(boolean bNewVal);

 public boolean GetIncludePortType();
 public void SetIncludePortType(boolean bNewVal);

 public boolean GetIncludeService();
 public void SetIncludeService(boolean bNewVal);

 public boolean GetIncludeTypes();
 public void SetIncludeTypes(boolean bNewVal);

 public void AllDetails(boolean bNewVal);

 public boolean GetShowBindingDiagram();
 public void SetShowBindingDiagram(boolean bNewVal);
}
```

```
public boolean GetShowExtensibility();
public void SetShowExtensibility(boolean bNewVal);

public boolean GetShowMessageParts();
public void SetShowMessageParts(boolean bNewVal);

public boolean GetShowPort();
public void SetShowPort(boolean bNewVal);

public boolean GetShowPortTypeDiagram();
public void SetShowPortTypeDiagram(boolean bNewVal);

public boolean GetShowPortTypeOperations();
public void SetShowPortTypeOperations(boolean bNewVal);

public boolean GetShowServiceDiagram();
public void SetShowServiceDiagram(boolean bNewVal);

public boolean GetShowSourceCode();
public void SetShowSourceCode(boolean bNewVal);

public boolean GetShowTypesDiagram();
public void SetShowTypesDiagram(boolean bNewVal);

public boolean GetShowUsedBy();
public void SetShowUsedBy(boolean bNewVal);

// Since version 2010
public boolean GetEmbedCSSInHTML();
public void SetEmbedCSSInHTML(boolean i_bVal);

public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder(boolean i_bVal);

// Since version 2011r2
public boolean GetUseFixedDesign();
public void SetUseFixedDesign(boolean i_bVal);

public String GetSPSFile();
public void SetSPSFile(String i_strVal);

}
```

**SpyXBRLDocumentationDlg**

```
// Since version 2010
public class SpyXBRLDocumentationDlg
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();

 public long GetOptionsDialogAction();
 public void SetOptionsDialogAction(long nNewVal);

 public long GetOutputDialogAction();
 public void SetOutputDialogAction(long nNewVal);

 public boolean GetShowProgressBar();
 public void SetShowProgressBar(boolean bNewVal);

 public String GetOutputFile();
 public void SetOutputFile(String sNewVal);

 public long GetOutputFormat();
 public void SetOutputFormat(long nNewVal);

 public boolean GetEmbedCSSInHTML();
 public void SetEmbedCSSInHTML(boolean bNewVal);

 public long GetDiagramFormat();
 public void SetDiagramFormat(long nNewVal);

 public boolean GetEmbedDiagrams();
 public void SetEmbedDiagrams(boolean bNewVal);

 public boolean GetCreateDiagramsFolder();
 public void SetCreateDiagramsFolder(boolean bNewVal);

 public boolean GetShowResult();
 public void SetShowResult(boolean bNewVal);

 public void IncludeAll(boolean bNewVal);
 public void AllDetails(boolean bNewVal);

 public boolean GetIncludeOverview();
 public void SetIncludeOverview(boolean bNewVal);

 public boolean GetIncludeNamespacePrefixes();
 public void SetIncludeNamespacePrefixes(boolean bNewVal);

 public boolean GetIncludeGlobalElements();
 public void SetIncludeGlobalElements(boolean bNewVal);

 public boolean GetIncludeDefinitionLinkroles();
 public void SetIncludeDefinitionLinkroles(boolean bNewVal);

 public boolean GetIncludePresentationLinkroles();
 public void SetIncludePresentationLinkroles(boolean bNewVal);

 public boolean GetIncludeCalculationLinkroles();
 public void SetIncludeCalculationLinkroles(boolean bNewVal);
}
```

```
public boolean GetShowDiagram();
public void SetShowDiagram(boolean bNewVal);

public boolean GetShowSubstitutiongroup();
public void SetShowSubstitutiongroup(boolean bNewVal);

public boolean GetShowItemtype();
public void SetShowItemtype(boolean bNewVal);

public boolean GetShowBalance();
public void SetShowBalance(boolean bNewVal);

public boolean GetShowPeriod();
public void SetShowPeriod(boolean bNewVal);

public boolean GetShowAbstract();
public void SetShowAbstract(boolean bNewVal);

public boolean GetShowNillable();
public void SetShowNillable(boolean bNewVal);

public boolean GetShowLabels();
public void SetShowLabels(boolean bNewVal);

public boolean GetShowReferences();
public void SetShowReferences(boolean bNewVal);

public boolean GetShowLinkbaseReferences();
public void SetShowLinkbaseReferences(boolean bNewVal);

public boolean GetShortQualifiedname();
public void SetShortQualifiedname(boolean bNewVal);

public boolean GetShowImportedElements();
public void SetShowImportedElements(boolean bNewVal);

// Since version 2011r2
public boolean GetUseFixedDesign();
public void SetUseFixedDesign(boolean i_bVal);

public String GetSPSFile();
public void SetSPSFile(String i_strVal);
};
```

## SpyXMLData

```
public class SpyXMLData
{
 public void ReleaseInstance\(\);
 public void AppendChild(SpyXMLData oNewData);
 public void EraseAllChildren\(\);
 public void EraseCurrentChild\(\);
 public SpyXMLData GetCurrentChild\(\);
 public SpyXMLData GetFirstChild(long nKind);
 public SpyXMLData GetNextChild\(\);
 public boolean GetHasChildren\(\);
 public void InsertChild(SpyXMLData oNewData);
 public boolean IsSameNode(SpyXMLData oToComp);
 public long GetKind\(\);
 public boolean GetMayHaveChildren\(\);
 public String GetName\(\);
 public void SetName(String sValue);
 public SpyXMLData GetParent\(\);
 public String GetTextValue\(\);
 public void SetTextValue(String sValue);
}
```



**Authentic****SpyAuthenticRange**

```

// Since version 2004R3
public class SpyAuthenticRange
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public SpyAuthenticView GetParent();
 public SpyAuthenticRange GotoNext(long eKind);
 public SpyAuthenticRange GotoPrevious(long eKind);
 public void Select();
 public long GetFirstTextPosition();
 public void SetFirstTextPosition(long nTextPosition);
 public long GetLastTextPosition();
 public void SetLastTextPosition(long nTextPosition);
 public String GetText();
 public void SetText(String strText);
 public boolean PerformAction(long eAction, String strElementName);
 public boolean CanPerformAction(long eAction, String strElementName);
 public String[] CanPerformActionWith(long eAction);
 public SpyAuthenticRange GoTo(long eKind, long nCount, long nFrom);
 public SpyAuthenticRange SelectNext(long eKind);
 public SpyAuthenticRange SelectPrevious(long eKind);
 public SpyAuthenticRange MoveBegin(long eKind, long nCount);
 public SpyAuthenticRange MoveEnd(long eKind, long nCount);
 public SpyAuthenticRange ExpandTo(long eKind);
 public SpyAuthenticRange CollapsToBegin();
 public SpyAuthenticRange CollapsToEnd();
 public SpyAuthenticRange GotoNextCursorPosition();
 public SpyAuthenticRange GotoPreviousCursorPosition();
 public boolean IsEmpty();
 public boolean IsEqual(SpyAuthenticRange ipCmp);
 public SpyAuthenticRange Clone();
 public SpyAuthenticRange SetFromRange(SpyAuthenticRange ipSrc);
 public boolean Delete();
 public boolean Cut();
 public boolean Copy();
 public boolean Paste();
 public SpyXMLData GetFirstXMLData();
 public void SetFirstXMLData(SpyXMLData objXMLDataPtr);
 public long GetFirstXMLDataOffset();
 public void SetFirstXMLDataOffset(long nOffset);
 public SpyXMLData GetLastXMLData();
 public void SetLastXMLData(SpyXMLData objXMLDataPtr);
 public long GetLastXMLDataOffset();
 public void SetLastXMLDataOffset(long nOffset);
 public String[] GetElementHierarchy();
 public String[] GetElementAttributeNames(String strElementName);
 public boolean HasElementAttribute(String strElementName, String
strAttributeName);
 public String GetElementAttributeValue(String strElementName, String
strAttributeName);
 public void SetElementAttributeValue(String strElementName, String
strAttributeName, String strNewValue);
 public String[] GetEntityNames();
 public void InsertEntity(String strEntityName);
 public boolean IsInDynamicTable();
 public boolean AppendRow();
 public boolean InsertRow();
 public boolean DuplicateRow();
}

```

```

 public boolean DeleteRow();
 public boolean MoveRowUp();
 public boolean MoveRowDown();

 // Since version 2004R4
 public boolean IsCopyEnabled();
 public boolean IsCutEnabled();
 public boolean IsPasteEnabled();
 public boolean IsDeleteEnabled();
 public boolean IsTextStateApplied(String i_strElementName);
 public boolean IsFirstRow();
 public boolean IsLastRow();
}

```

### SpyAuthenticView

```

// Since version 2004R3
public class SpyAuthenticView
{
 public void ReleaseInstance();
 public SpyApplication GetApplication();
 public SpyDoc GetParent();
 public SpyAuthenticRange GetSelection();
 public void SetSelection(SpyAuthenticRange obj);
 public SpyAuthenticRange GetDocumentBegin();
 public SpyAuthenticRange GetDocumentEnd();
 public SpyAuthenticRange GetWholeDocument();
 public long GetMarkupVisibility();
 public void SetMarkupVisibility(long eSpyAuthenticMarkupVisibility);
 public SpyAuthenticRange GoTo(long eKind, long nCount, long nFrom);
 public void Print(boolean bWithPreview, boolean bPromptUser);
 public boolean Undo();
 public boolean Redo();
 public void UpdateXMLInstanceEntities();

 // Since version 2004R4
 public String GetAsXMLString();
 public void SetAsXMLString(String i_strXML);
 public SpyXMLData GetXMLDataRoot();
 public boolean IsUndoEnabled();
 public boolean IsRedoEnabled();
}

```

### SpyDocEditSelection

```

public class SpyDocEditSelection
{
 public void ReleaseInstance();
 public SpyXMLData GetEnd();
 public long GetEndTextPosition();
 public SpyXMLData GetStart();
 public long GetStartTextPosition();
}

```

### SpyDocEditView

```

public class SpyDocEditView
{
 public void ReleaseInstance();
 public void ApplyTextState(String sElementName);
 public SpyDocEditSelection GetCurrentSelection();
 public void EditClear();
}

```

```
public void EditCopy();
public void EditCut();
public void EditPaste();
public void EditRedo();
public void EditSelectAll();
public void EditUndo();
public SpyXMLData GetNextVisible(SpyXMLData oElement);
public SpyXMLData GetPreviousVisible(SpyXMLData oElement);
public boolean GetIsEditClearEnabled();
public boolean GetIsEditCopyEnabled();
public boolean GetIsEditCutEnabled();
public boolean GetIsEditPasteEnabled();
public boolean GetIsEditRedoEnabled();
public boolean GetIsEditUndoEnabled();
public boolean GetIsRowAppendEnabled();
public boolean GetIsRowDeleteEnabled();
public boolean GetIsRowDuplicateEnabled();
public boolean GetIsRowInsertEnabled();
public boolean GetIsRowMoveDownEnabled();
public boolean GetIsRowMoveUpEnabled();
public boolean IsTextStateApplied(String sElementName);
public boolean IsTextStateEnabled(String sElementName);
public void LoadXML(String sXML);
public void MarkupView(long nKind);
public void RowAppend();
public void RowDelete();
public void RowDuplicate();
public void RowInsert();
public void RowMoveDown();
public void RowMoveUp();
public String SaveXML();
public void SelectionMoveTabOrder(boolean bForward, boolean bTag);
public boolean SelectionSet(SpyXMLData oStart, long nStartPos, SpyXMLData
oEndElement, long nEndPos);
public SpyXMLData GetXMLRoot();
public String[] GetAllowedElements(long nAction, SpyXMLData oStartPtr,
SpyXMLData oEndPtr);
}
```

### Vordefinierte Konstanten

In diesem Abschnitt sind alle Klassen aufgelistet, die die von der Java-Schnittstelle verwendeten vordefinierten Konstanten verwenden.

#### SPYApplicationStatus

```
public class SPYApplicationStatus
{
 public final static long spyApplicationStatus_Running = 0;
 public final static long spyApplicationStatus_AfterLicenseCheck = 1;
 public final static long spyApplicationStatus_BeforeLicenseCheck = 2;
 public final static long spyApplicationStatus_ConcurrentLicenseCheckFailed = 3;
 public final static long spyApplicationStatus_ProcessingCommandLine = 4;
}
```

#### SPYAttributeTypeDefinition

```
public class SPYAttributeTypeDefinition
{
 public final static long spyMergedGlobal = 0;
 public final static long spyDistinctGlobal = 1;
 public final static long spyLocal = 2;
}
```

#### SPYAuthenticActions

```
public class SPYAuthenticActions
{
 public final static long spyAuthenticInsertAt = 0;
 public final static long spyAuthenticApply = 1;
 public final static long spyAuthenticClearSurr = 2;
 public final static long spyAuthenticAppend = 3;
 public final static long spyAuthenticInsertBefore = 4;
 public final static long spyAuthenticRemove = 5;
}
```

#### SPYAuthenticDocumentPosition

```
public class SPYAuthenticDocumentPosition
{
 public final static long spyAuthenticDocumentBegin = 0;
 public final static long spyAuthenticDocumentEnd = 1;
}
```

```
 public final static long = 2;
 spyAuthenticRangeBegin
 public final static long = 3;
 spyAuthenticRangeEnd
}
```

### SPYAuthenticElementKind

```
public class SPYAuthenticElementKind
{
 public final static long = 0;
 spyAuthenticChar
 public final static long = 1;
 spyAuthenticWord
 public final static long = 3;
 spyAuthenticLine
 public final static long = 4;
 spyAuthenticParagraph
 public final static long = 6;
 spyAuthenticTag
 public final static long = 8;
 spyAuthenticDocument
 public final static long = 9;
 spyAuthenticTable
 public final static long = 10;
 spyAuthenticTableRow
 public final static long = 11;
 spyAuthenticTableColumn
}
```

### SPYAuthenticMarkupVisibility

```
public class SPYAuthenticMarkupVisibility
{
 public final static long = 0;
 spyAuthenticMarkupHidden
 public final static long = 1;
 spyAuthenticMarkupSmall
 public final static long = 2;
 spyAuthenticMarkupLarge
 public final static long = 3;
 spyAuthenticMarkupMixed
}
```

### SPYDatabaseKind

```
public class SPYLoading
{
 public final static long = 0;
 spyDB_Access
 public final static long = 1;
 spyDB_SQLServer
 public final static long = 2;
 spyDB_Oracle
 public final static long = 3;
 spyDB_Sybase
}
```

```
 public final static long = 4;
 spyDB MySQL
 public final static long spyDB_DB2 = 5;
 public final static long = 6;
 spyDB Other
 public final static long = 7;
 spyDB_Unspecified
}
```

#### SPYDialogAction

```
public class SPYDialogAction
{
 public final static long = 0;
 spyDialogOK
 public final static long = 1;
 spyDialogCancel
 public final static long = 2;
 spyDialogUserInput
}
```

#### SPYDOMType

```
public class SPYDOMType
{
 public final static long = 0;
 spyDOMType_msxml4
 public final static long = 1;
 spyDOMType_xerces
}
```

#### SPYDTDSchemaFormat

```
public class SPYDTDSchemaFormat
{
 public final static long spyDTD = 0;
 public final static long spyDCD = 1;
 public final static long spyXMLData = 2;
 public final static long spyBizTalk = 3;
 public final static long spyW3C = 4;
}
```

#### SPYEncodingByteOrder

```
public class SPYEncodingByteOrder
{
 public final static long spyNONE = 0;
 public final static long = 1;
 spyLITTLE_ENDIAN
 public final static long = 2;
 spyBIG_ENDIAN
}
```

#### SPYExportNamespace

```
public class SPYExportNamespace
```

```
{
 public final static long spyNoNamespace = 0;
 public final static long spyReplaceColonWithUnderscore = 1;
}
```

#### SPYFindInFilesSearchLocation

```
public class SPYFindInFilesSearchLocation
{
 public final static long spyFindInFiles_Documents = 0;
 public final static long spyFindInFiles_Project = 1;
 public final static long spyFindInFiles_Folder = 2;
}
```

#### SPYFrequentElements

```
public class SPYFrequentElements
{
 public final static long spyGlobalElements = 0;
 public final static long spyGlobalComplexType = 1;
}
```

#### SPYImageKind

```
public class SPYImageKind
{
 public final static long spyImageType_PNG = 0;
 public final static long spyImageType_EMF = 1;
}
```

#### SPYLibType

```
public class SPYLibType
{
 public final static long spyLibType_static = 0;
 public final static long spyLibType_dll = 1;
}
```

#### SPYLoading

```
public class SPYLoading
{
 public final static long spyUseCacheProxy = 0;
 public final static long spyReload = 1;
}
```

## SPYNumberDateTimeFormat

```
public class SPYNumberDateTimeFormat
{
 public final static long = 0;
 spySystemLocale
 public final static long = 1;
 spySchemaCompatible
}
```

## SPYProgrammingLanguage

```
public class SPYLoading
{
 public final static long spyUndefinedLanguage = -1;
 public final static long spyJava = 0;
 public final static long spyCpp = 1;
 public final static long spyCSharp = 2;
}
```

## SPYProjectItemTypes

```
public class SPYProjectItemTypes
{
 public final static long = 0;
 spyUnknownItem
 public final static long = 1;
 spyFileItem
 public final static long = 2;
 spyFolderItem
 public final static long = 3;
 spyURLItem
}
```

## SPYProjectType

```
public class SPYProjectType
{
 public final static long = 0;
 spyVisualStudioProject
 public final static long = 1;
 spyVisualStudio2003Project
 public final static long = 2;
 spyBorlandProject
 public final static long = 3;
 spyMonoMakefile
}
```

## SPYSampleXMLGenerationOptimization

```
public class SPYSampleXMLGenerationOptimization
{
 public final static long = 0;
 spySampleXMLGen_Optimized
 public final static long = 1;
 spySampleXMLGen NonMandatoryElements
}
```



```

 public final static long = 2;
 spySampleXMLGen Everything
}

```

#### SPYSampleXMLGenerationSchemaOrDTDAssignment

```

public class SPYSampleXMLGenerationOptimization
{
 public final static long = 0;
 spySampleXMLGen_AssignRelatively
 public final static long = 1;
 spySampleXMLGen_AssignAbsolutely
 public final static long = 2;
 spySampleXMLGen DoNotAssign
}

```

#### SPYSchemaDefKind

```

public class SPYSchemaDefKind
{
 public final static long spyKindElement = 0;
 public final static long = 1;
 spyKindComplexType
 public final static long = 2;
 spyKindSimpleType
 public final static long spyKindGroup = 3;
 public final static long spyKindModel = 4;
 public final static long spyKindAny = 5;
 public final static long spyKindAttr = 6;
 public final static long = 7;
 spyKindAttrGroup
 public final static long spyKindAttrAny = 8;
 public final static long = 9;
 spyKindIdentityUnique
 public final static long = 10;
 spyKindIdentityKey
 public final static long = 11;
 spyKindIdentityKeyRef
 public final static long = 12;
 spyKindIdentitySelector
 public final static long = 13;
 spyKindIdentityField
 public final static long spyKindNotation = 14;
 public final static long spyKindInclude = 15;
 public final static long spyKindImport = 16;
 public final static long spyKindRedefine = 17;
 public final static long spyKindFacet = 18;
 public final static long spyKindSchema = 19;
 public final static long spyKindCount = 20;
}

```

#### SPYSchemaDocumentationFormat

```

public class SPYSchemaDocumentationFormat
{
 public final static long = 0;
 spySchemaDoc_HTML
 public final static long = 1;
 spySchemaDoc MSWord
}

```

```

 public final static long = 2;
 spySchemaDoc RTF
 public final static long = 3;
 spySchemaDoc_PDF
}

```

### SPYSchemaExtensionType

```

public class SPYSchemaExtensionType
{
 public final static long = 0;
 spySchemaExtension_None
 public final static long = 1;
 spySchemaExtension_SQL_XML
 public final static long = 2;
 spySchemaExtension_MS_SQL_Server
 public final static long = 3;
 spySchemaExtension_Oracle
}

```

### SPYSchemaFormat

```

public class SPYSchemaFormat
{
 public final static long = 0;
 spySchemaFormat_Hierarchical
 public final static long = 1;
 spySchemaFormat_Flat
}

```

### SPYTextDelimiters

```

public class SPYTextDelimiters
{
 public final static long = 0;
 spyTabulator
 public final static long = 1;
 spySemicolon
 public final static long = 2;
 spyComma
 public final static long = 3;
 spySpace
}

```

### SPYTextEnclosing

```

public class SPYTextEnclosing
{
 public final static long = 0;
 spyNoEnclosing
 public final static long = 1;
 spySingleQuote
 public final static long = 2;
 spyDoubleQuote
}

```

### SPYTypeDetection

```

public class SPYTypeDetection

```

```
{
 public final static long = 0;
 spyBestPossible
 public final static long = 1;
 spyNumbersOnly
 public final static long = 2;
 spyNoDetection
}
```

### SPYURLTypes

```
public class SPYURLTypes
{
 public final static long = (-1);
 spyURLTypeAuto
 public final static long = 0;
 spyURLTypeXML
 public final static long = 1;
 spyURLTypeDTD
}
```

### SpyViewModes

```
public class SPYViewModes
{
 public final static long = 0;
 spyViewGrid
 public final static long = 1;
 spyViewText
 public final static long = 2;
 spyViewBrowser
 public final static long = 3;
 spyViewSchema
 public final static long = 4;
 spyViewContent
 public final static long = 4;
 spyViewAuthentic
 public final static long = 5;
 spyViewWSDL
 public final static long spyViewZIP = 6;
 public final static long = 7;
 spyViewEditionInfo
}
```

### SPYWhitespaceComparison

```
public class SPYWhitespaceComparison
{
 public final static long = 0;
 spyCompareAsIs
 public final static long = 1;
 spyCompareNormalized
 public final static long = 2;
 spyStripAll
}
```

## SPYXMLDataKind

```
public class SPYXMLDataKind
{
 public final static long = 0;
 spyXMLDataXMLDocStruct
 public final static long = 1;
 spyXMLDataXMLEntityDocStruct
 public final static long = 2;
 spyXMLDataDTDDocStruct
 public final static long = 3;
 spyXMLDataXML
 public final static long = 4;
 spyXMLDataElement
 public final static long = 5;
 spyXMLDataAttr
 public final static long = 6;
 spyXMLDataText
 public final static long = 7;
 spyXMLDataCDATA
 public final static long = 8;
 spyXMLDataComment
 public final static long = 9;
 spyXMLDataPI
 public final static long = 10;
 spyXMLDataDefDoctype
 public final static long = 11;
 spyXMLDataDefExternalID
 public final static long = 12;
 spyXMLDataDefElement
 public final static long = 13;
 spyXMLDataDefAttlist
 public final static long = 14;
 spyXMLDataDefEntity
 public final static long = 15;
 spyXMLDataDefNotation
 public final static long = 16;
 spyXMLDataKindsCount
}
```

## 20.3 ActiveX Integration

StyleVisionControl ist ein Control, das ein Mittel zur Integration der StyleVision Benutzeroberfläche und der in diesem Abschnitt beschriebenen Funktionalitäten in die meisten Applikationsarten darstellt. Es wurde die ActiveX-Technologie gewählt, um die Integration mittels jeder beliebigen der vielen Programmiersprachen zu ermöglichen; dadurch können für die Integration C++, C#, VisualBasic oder HTML verwendet werden. ActiveX-Komponenten funktionieren offiziell nur mit Microsoft Internet Explorer. Alle Komponenten sind vollständige OLE Controls, wodurch die Integration so einfach wie möglich gemacht wird. Es stehen zwei verschiedene Ebenen der Integration zur Verfügung, sodass die Integration an die verschiedensten Bedürfnisse angepasst werden kann.

Um StyleVision zu integrieren, müssen Sie das StyleVision Integrationspaket installieren. Stellen Sie sicher, dass Sie StyleVision zuerst installieren und dann erst das StyleVision Integrationspaket.

Für eine erfolgreiche Integration müssen beim Design die folgenden Hauptfaktoren berücksichtigt werden:

- Welche Technologie oder Programmiersprache kann die Host-Applikation zur Integration des StyleVisionControl verwenden?
- Soll die integrierte Benutzeroberfläche genau wie StyleVision mit all seinen Menüs, Symbolleisten und Fenstern aussehen oder wäre eine Untermenge dieser Elemente - z.B. dass nur ein Dokument und eine eingeschränkte Anzahl an Befehlen verwendet werden kann - sinnvoller?
- Wie tief wird die Integration sein? Soll die StyleVision Benutzeroberfläche, wie sie ist, verwendet werden? Sind Erweiterungen zur Benutzeroberfläche und/oder Einschränkungen erforderlich? Können einige häufig verwendete Aufgaben automatisiert werden?

Die Abschnitte [Integration auf Applikationsebene](#) und [Integration auf Dokumentebene](#), von denen jeder Beispiele in den verschiedenen Programmiersprachen enthält, sollen Ihnen dabei helfen, rasch die richtige Entscheidung zu treffen. Der Abschnitt [Objektreferenz](#) enthält eine Beschreibung aller COM-Objekte mit ihren Eigenschaften und Methoden, die für die Integration verwendet werden können.

Für Automationsaufgaben kann die [StyleVision Automation-Schnittstelle](#) auch über das StyleVisionControl aufgerufen werden.

Informationen zum Integrieren von StyleVision in Microsoft Visual Studio finden Sie im Abschnitt [StyleVision in Visual Studio](#).

## Integration auf Applikationsebene

Die Integration auf Applikationsebene ist einfach. Sie können dadurch die gesamte Benutzeroberfläche von StyleVision in ein Fenster Ihrer Applikation einbetten. Da Sie die gesamte Benutzeroberfläche von StyleVision zur Verfügung haben, stehen Ihnen alle Menüs, Symbolleisten, die Statusleiste, die Dokumentfenster und Eingabehilfen zur Verfügung. Die Anpassung der Benutzeroberfläche der Applikation ist auf die Optionen eingeschränkt, die StyleVision bietet. Dazu gehören die Neuordnung und Anpassung der Größe der Eingabehilfen und die Anpassung von Menüs und Symbolleisten.

Das einzige ActiveX Control, das Sie integrieren müssen, ist [StyleVisionControl](#). Der Wert seiner Eigenschaft [IntegrationLevel](#) ist standardmäßig "application-level". Sie können das Aussehen des Wrapper-Fensters des Control mit Hilfe von [Appearance](#) und [BorderStyle](#) konfigurieren. Bei Integration auf Applikationsebene dürfen [StyleVisionControlDocument](#) oder [StyleVisionControlPlaceholder](#) ActiveX Controls nicht instantiiert oder aufgerufen werden.

Wenn Sie Initialisierungen vornehmen müssen oder ein bestimmtes Verhalten von StyleVision automatisieren wollen, verwenden Sie die für [StyleVisionControl](#) beschriebenen Eigenschaften, Methoden und Events. Um komplexere StyleVision Funktionsaufrufe auszuführen, sollten Sie eventuell [StyleVisionControl.Application](#) verwenden.

In diesem Abschnitt finden Sie ein Beispiel ([Beispiel: HTML](#)), in dem gezeigt wird, wie die StyleVision Applikation in eine HTML-Seite eingebettet werden kann. Eine Beschreibung zur Verwendung mit anderen Programmiersprachen oder kompliziertere Aufrufe finden Sie unter den [Beispielen](#) zur Integration auf Dokumentebene.

**Beispiel: HTML**

In diesem Beispiel wird eine einfache Integration des StyleVision Control auf Applikationsebene in eine HTML-Seite beschrieben. Die Integration wird in den folgenden Abschnitten beschrieben:

- Instantiieren Sie ein StyleVisionControl in HTML-Code.
- Implementieren Sie Schaltflächen zum Laden von Dokumenten und automatisieren Sie Codegenerierungsaufgaben.
- Definieren Sie Aktionen für einige Applikationsevents.

Den Code für dieses Beispiel finden Sie in Ihrer StyleVision Installation im folgenden Ordner: StyleVisionExamples\ActiveX\HTML\StyleVisionActiveX\_ApplicationLevel.htm.

**Anmerkung:** Dieses Beispiel funktioniert nur in Internet Explorer.

**Instantiieren des Control**

Der HTML Object Tag dient zum Erstellen einer Instanz des StyleVisionControl. Die Classid ist die von StyleVisionControl. Mit "width" und "height" wird die Fenstergröße definiert. Es sind keine weiteren Parameter erforderlich, da die Standardeinstellung "application-level" ist.

```
<OBJECT id="objStyleVisionControl"
 Classid="clsid: 559db547-71fc-435d-aa8e-7faa321d0c6e"
 width="1000"
 height="700"
 VIEWASTEXT>
</OBJECT>
```

**Hinzufügen einer Schaltfläche zum Öffnen eines Standarddokuments**

Als einfaches Beispiel, wie man manche Aufgaben automatisiert, fügen wir eine Schaltfläche zur Seite hinzu:

```
<input type="button" value="Open Marketing Expenses"
onclick="BtnOpenMEFile()" ">
```

Wenn Sie darauf klicken, wird ein vordefiniertes Dokument im StyleVisionControl geöffnet. Wir verwenden eine Methode, um die Datei relativ zum StyleVisionControl anzugeben, daher kann das Beispiel in unterschiedlichen Installationen ausgeführt werden.

```
<SCRIPT ID="Javahandlers" LANGUAGE="javascript">
// -----
// open a pre-defined document
function BtnOpen()
{
 if(strPath.value.length > 0)
 {
 var absolutePath = MakeAbsolutePath(strPath.value);
 var objDoc = objStyleVisionControl.Open(absolutePath);
 if (objDoc == null)
 alert("Unable to locate " + absolutePath + " at: " +
objStyleVisionControl.BaseHref);
 }
 else
 {
 alert("Please set path for the document first!");
 strPath.focus();
 }
}
</SCRIPT>
```

## Überprüfen der Gültigkeit des aktuellen Dokuments

Die Gültigkeit des aktuellen Dokuments wird mit Hilfe des folgenden Skripts überprüft:

```
// check validity of current document.
// if validation fails, show validation result in alert box .
function BtnTest()
{
 // get top-level object of automation interface
 var objApp = objStyleVisionControl.Application;

 // get the active document
 var objDocument = objApp.ActiveDocument;

 if (objDocument == null)
 alert("no active document found");
 else
 {
 // define as arrays to support their usage as return parameters
 var errorText = new Array(1);
 var errorPos = new Array(1);
 var badData = new Array(1);

 var valid = objDocument.IsValid(errorText, errorPos, badData);

 if (! valid)
 {
 // compose the error description
 var text = errorText;

 // access that XMLData object only if filled in
 if (badData[0] != null)
 text += "(" + badData[0].Name + "/" + badData[0].TextValue +
") ";

 alert("Validation error[" + errorPos + "]: " + text);
 }
 else
 alert("Document is valid");
 }
}
```

## Verbindung zu benutzerdefinierten Events herstellen

In diesem Beispiel werden zur Veranschaulichung des Prinzips zwei Event Callbacks für benutzerdefinierte StyleVisionControl Events implementiert:

```
<!-- ----- -->
<!-- custom event 'OnDocumentOpened' of StyleVisionControl object -->
<SCRIPT LANGUAGE="javascript">
 function objStyleVisionControl::OnDocumentOpened(objDocument)
 {
 // alert("Document '" + objDocument.Name + "' opened!");
 }
</SCRIPT>

<!-- ----- -->
<!-- custom event 'OnDocumentClosed' of StyleVisionControl object -->
<SCRIPT LANGUAGE="javascript">
```



```
function objStyleVisionControl::OnDocumentClosed(objDocument)
{
 // alert("Document '" + objDocument.Name + "' closed!");
}
</SCRIPT>
```

## Integration auf Dokumentebene

Bei der Integration auf Dokumentebene haben Sie Freiheit bei der Instantiierung und Platzierung der folgenden Teile der StyleVision Benutzeroberfläche:

Falls erforderlich, muss in Ihrer Applikation ein Ersatz für die Menüs und Symbolleisten von StyleVision zur Verfügung gestellt werden.

Je nachdem, welche Teile der Benutzeroberfläche Sie wiederverwenden möchten, müssen Sie mehrere ActiveX Controls instantiieren und aufrufen. Alle diese Controls sind in der StyleVisionControl OCX-Datei enthalten.

- [Mit Hilfe von StyleVisionControl](#) können Sie die Integrationsebene definieren und auf applikationsweite Funktionalitäten zugreifen.
- [Mit Hilfe von StyleVisionControlDocument](#) können Sie beliebig viele Editor-Fenster erstellen. Eventuell genügt es, nur ein Fenster zu erstellen und dieses wiederzuverwenden.
- Verwenden Sie optional [StyleVisionControlPlaceholder](#) um StyleVision Eingabehilfenfenster, Validierungsausgabefenster oder andere oben erwähnte Fenster einzubetten.
- Rufen Sie Laufzeitinformationen über in StyleVisionControl verfügbare Befehle, Menüs und Symbolleisten auf, um diese Befehle nahtlos in Ihre Applikationsmenüs und Symbolleisten zu integrieren. Nähere Informationen dazu finden Sie unter [Abfrage von StyleVision Befehlen](#).

Wenn Sie ein bestimmtes Verhalten von StyleVision automatisieren möchten, verwenden Sie die für [StyleVisionControl](#), [StyleVisionControlDocument](#) und [StyleVisionControlPlaceholder](#) beschriebenen Eigenschaften, Methoden und Events. Bei Aufruf komplexerer StyleVision Funktionalitäten sollten Sie eventuell [StyleVisionControl.Application](#), [StyleVisionControlDocument.Document](#) und [StyleVisionControlPlaceholder.Project](#) verwenden. Verwenden Sie allerdings zum Öffnen eines Dokuments immer [StyleVisionControlDocument.Open](#) oder [StyleVisionControlDocument.New](#) am entsprechenden Document Control. Verwenden Sie zum Öffnen eines Projekts immer [StyleVisionControlPlaceholder.OpenProject](#) an einem Placeholder Control, das ein StyleVision Projektfenster einbettet.

Beschreibungen zum Instantiieren und Aufrufen der erforderlichen Controls in verschiedenen Programmierungsumgebungen finden Sie unter [Beispiele](#).

## Verwendung von StyleVisionControl

Zur Integration auf Dokumentebene instantiieren Sie zuerst ein [StyleVisionControl](#). Setzen Sie die Eigenschaft [IntegrationLevel](#) auf `IActiveXIntegrationOnDocumentLevel(=1)`. Setzen Sie die Fenstergröße des einbettenden Fensters auf `0x0`, damit eine andere Benutzeroberfläche gegebenenfalls vom Control verdeckt wird. Sie können das Aussehen des Wrapper-Fensters für das Control mittels [Appearance](#) und [BorderStyle](#) konfigurieren.

Vermeiden Sie nach Möglichkeit die Verwendung der Methode [Open](#), da dies zu unerwarteten Ergebnissen führen könnte. Verwenden Sie stattdessen die entsprechenden Öffnungsmethoden von [StyleVisionControlDocument](#) und [StyleVisionControlPlaceholder](#).

Eine Beschreibung zur Integration von StyleVision Befehlen in Ihre Applikation finden Sie unter [Abfragen von StyleVision Befehlen](#). Senden Sie Befehle über die Methode [Exec](#) an StyleVision. Führen Sie mit Hilfe der Methode [QueryStatus](#) eine Abfrage durch, um zu sehen, ob ein Befehl derzeit aktiviert oder deaktiviert ist.

### Verwendung von StyleVisionControlDocument

Mit Hilfe einer Instanz des `StyleVisionControlDocument` ActiveX Control können Sie ein StyleVision Mapping-Bearbeitungsfenster in Ihre Applikation einbetten. Sie können so viele Instanzen verwenden, so viele Sie benötigen.

Verwenden Sie die Methode [Open](#), um eine beliebige andere bereits vorhandene Datei zu laden.

Das Control unterstützt keinen schreibgeschützten Modus. Der Wert der Eigenschaft [ReadOnly](#) wird ignoriert.

Verwenden Sie zum Aufrufen von Dokumentfunktionalitäten [Path](#) und [Save](#) oder Methoden und Eigenschaften, die über die Eigenschaft [Document](#) aufgerufen werden.

### Verwendung von StyleVisionControlPlaceHolder

Mit Hilfe von Instanzen des `StyleVisionControlPlaceHolder` ActiveX Control können Sie selektiv weitere Eingabehilfenfenster von StyleVision in Ihre Applikation einbetten. Die Eigenschaft [PlaceholderWindowID](#) wählt das StyleVision Eingabehilfenfenster aus, um es einzubetten. Verwenden Sie für jeden Fenster-Identifizier nur einen `StyleVisionControlPlaceHolder`. Eine Beschreibung gültiger Fenster-Identifizier finden Sie unter .

Für Platzhalter-Controls, die das StyleVision Projektfenster auswählen, stehen zusätzliche Methoden zur Verfügung. Verwenden Sie [OpenProject](#), um ein StyleVision Projekt zu laden. Verwenden Sie die Eigenschaft [Project](#) und die Methoden und Eigenschaften aus dem StyleVision Automation Interface, um andere projektbezogene Operationen auszuführen.

**Abfragen von StyleVision Befehlen**

Bei der Integration auf Dokumentebene stehen Ihrer Applikation kein Menü und keine Symbolleiste aus StyleVision zur Verfügung. Stattdessen können Sie alle Befehle und die Struktur des Applikationsmenüs zur Laufzeit abfragen. Bei professionellen Applikationen muss dieses Menü auf raffinierte Art in die eigene Menüstruktur integriert werden. In Ihrer Installation von StyleVision stehen Ihnen sogar Bilder für in StyleVision verwendete Befehle zur Verfügung. Symbole im GIF-Format finden Sie in Ihrem StyleVision Installationsordner unter `StyleVisionExamples\ActiveX\Images` of your StyleVision. Die Dateinamen entsprechen den [labels](#) von Befehlen.

## Beispiele

Dieser Abschnitt enthält Beispiele für die Integration von StyleVision auf Dokumentebene über unterschiedliche Container-Umgebungen und Programmiersprachen. Der Quellcode für alle Beispiele steht im Ordner `StyleVisionExamples\ActiveX` Ihrer StyleVision Installation zur Verfügung.

### C#

Im C#-Beispiel wird gezeigt, wie man das `StyleVisionControl` in eine allgemeine Desktop-Applikation integriert, die mit C# über Visual Studio 2008 erstellt wurde.

- 

Bitte beachten Sie, dass die Beispielapplikation bereits fertig ist. Sie müssen keine Änderungen vornehmen, wenn Sie sie ausführen und sehen möchten, wie sie funktioniert. In den folgenden Schritten wird beschrieben, welche Aktionen durchgeführt und welche Dinge berücksichtigt werden müssen, um ein Projekt wie dieses zu erstellen.

### Einführung

#### Hinzufügen der StyleVision -Komponenten zur Toolbox

Bevor Sie einen Blick in das Beispielprojekt werfen, fügen Sie bitte die Assemblies zur .NET IDE Toolbox hinzu. Der StyleVision Installer hat die Assemblies bereits im .NET Global Assembly Cache (GAC) installiert. Wenn Sie das Toolbox-Dialogfeld unter **Tools | Add/Remove Toolbox Items** öffnen, werden die Controls als `AxStyleVisionControl`, `AxStyleVisionControlDocument` und `AxStyleVisionControlPlaceholder` auf dem Register ".NET Framework Components" angezeigt. Aktivieren Sie alle, damit Sie in der IDE zur Verfügung stehen.

### Platzieren des StyleVisionControl

Sie benötigen eine `StyleVisionControl` Instanz, um die Integrationsebene zu definieren und die Dokument- und Platzhalter-Controls der StyleVision Bibliothek zu verwalten. Das Control kann über den Abschnitt "General" des Toolbox-Eingabehilfefensters in der IDE aufgerufen werden. Um es hinzuzufügen, müssen Sie die Komponente im Fenster "Toolbox" auswählen und ein Rechteck an der Stelle aufziehen, an der Sie es im Zielfenster haben möchten. Bei einer Applikation, die beim Start kein Fenster öffnet, können Sie ein einfaches unsichtbares Formular mit dem Control verwenden, an dem es im Code manuell erstellt wird.

Im Beispielprojekt wird diese Instanz zum Haupt-MdiContainer `MDIMain` hinzugefügt. Wenn Sie `MDIMain` in der Design-Ansicht über den Solution Explorer öffnen, sehen Sie im Client-Bereich des Frame-Fensters an der linken oberen Seite ein hellblaues Rechteck. Wenn Sie das Rechteck auswählen, sehen Sie die Eigenschaften des `StyleVisionControl`. Die Eigenschaft `IntegrationLevel` muss unbedingt auf `ICActiveXIntegrationOnDocumentLevel` gesetzt werden, um die Unterstützung der StyleVision Bibliothek für Dokumente und Platzhalter zu aktivieren.

### HTML

In diesem Beispiel wird eine Integration des StyleVision Control auf Dokumentebene in eine HTML-Seite gezeigt. Es werden die folgenden Themen behandelt:

- Instantiieren eines `StyleVisionControl` ActiveX Control-Objekts in HTML-Code
- Instantiieren eines `StyleVisionControlDocument` ActiveX Control für die Bearbeitung eines StyleVision Mappings
- Instantiieren eines `StyleVisionControlPlaceHolder` für ein StyleVision-Projektfenster
- Instantiieren eines `StyleVisionControlPlaceHolder` ActiveX Control für die Aufnahme eines der StyleVision-Eingabehilfefenster

- Erstellen einer benutzerdefinierten Symbolleiste für einige häufig gebrauchte StyleVision-Befehle
- Hinzufügen von weiteren Schaltflächen, die das COM Automation Interface von StyleVision verwenden
- Verwenden von Event Handlern zum Aktualisieren von Befehlsschaltflächen

Dieses Beispiel ist vollständig in der Datei `StyleVisionActiveX_ApplicationLevel.htm` im Ordner `C:\Program Files\Altova\StyleVision2012\Examples\ActiveX\HTML\ Ihrer StyleVision Installation` enthalten.

**Anmerkung:** Dieses Beispiel funktioniert nur in Internet Explorer.

#### Instantiieren des StyleVisionControl

Zum Erstellen einer Instanz des StyleVisionControl wird der HTML OBJECT Tag verwendet. Die Classid ist die von StyleVisionControl. "width" und "height" werden auf 0 gesetzt, da wir dieses Control als Manager Control ohne Verwendung für seine Benutzeroberfläche verwenden. Die Integrationsebene wird in Form eines Parameters im OBJECT Tag definiert.

```
<OBJECT id=""
 Classid="clsid: 559db547-71fc-435d-aa8e-7faa321d0c6e"
 width="0"
 height="0"
 VIEWASTEXT>
 <PARAM NAME="IntegrationLevel" VALUE="1">
</OBJECT>
```

#### Erstellen eines Bearbeitungsfensters

Der HTML OBJECT Tag wird zum Einbetten eines Dokumentenbearbeitungsfensters verwendet. Der zusätzliche benutzerdefinierte Parameter definiert, dass das Control mit einem neuen leeren Mapping initialisiert werden muss.

```
<OBJECT id="objDoc1"
 Classid="clsid: 8EE4E77E-C793-468B-A0D8-5D6334CD6986"
 width="600"
 height="500"
 VIEWASTEXT>
 <PARAM NAME="NewDocument">
</OBJECT>
```

#### Erstellen eines Projektfensters

Der HTML OBJECT Tag wird zum Erstellen eines StyleVisionControlPlaceholder-Fensters verwendet. Der erste zusätzliche benutzerdefinierte Parameter definiert, welcher Platzhalter das StyleVision-Projektfenster anzeigen soll. Der zweite Parameter lädt eines der Beispielpunkte aus Ihrer StyleVision Installation (aus dem Ordner `<yourusername>/MyDocuments`).

```
<OBJECT id="objProjectWindow"
 Classid="clsid: E6ECBA9C-0E01-4FD5-98C3-C08B3D4824BC"
 width="200"
 height="200"
 VIEWASTEXT>
 <PARAM name="PlaceholderWindowID" value="-1">
 <PARAM name="FileName" value="StyleVisionExamples/Examples.svp">
</OBJECT>
```



### Erstellen eines Placeholder für Eingabehilfenfenster

Der HTML OBJECT Tag wird zum Instantiieren eines StyleVisionControlPlaceholder ActiveX Control verwendet, das die verschiedenen StyleVision Hilfsfenster aufnehmen kann. Anfangs wird kein Hilfsfenster angezeigt. Siehe Beispieldatei:

```
<OBJECT id="objEHWindow"
 Classid="clsid: E6ECBA9C-0E01-4FD5-98C3-C08B3D4824BC"
 width="200"
 height="200"
 VIEWASTEXT>
 <PARAM name="PlaceholderWindowID" value="-1">
</OBJECT>
```

Mit Hilfe von drei Schaltflächen kann zwischen dem tatsächlichen Fenster, das angezeigt werden soll, gewechselt werden. Mit der JavaScript-Anweisung die bei Klick auf eine Schaltfläche ausgeführt wird, wird die Eigenschaft `PlaceHolderWindowID` auf den entsprechenden in definierten Wert gesetzt.

```
<input type="button" value="New File" onclick="BtnNewFile(objDoc1) ">
<input type="button" value="Open file" onclick="BtnOpenFile(objDoc1) ">
<input type="button" value="Save File" onclick="BtnSaveFile(objDoc1) ">
<input type="button" value="Open project" onclick="BtnOpenProject(objDoc1) ">
<input type="button" value="Save project" onclick="BtnSaveProject() ">
<input type="button" value="Close project" onclick="BtnCloseProject() ">
```

### Obligatorische Event Handler

Ein obligatorischer Event Handler, z.B. für eine bereits offene Datei:

```
<SCRIPT LANGUAGE="javascript">
 function objStyleVisionControl::OnOpenedOrFocused(strFilePath,
bOpenWithThisControl, bFileAlreadyOpened)
 {
 alert("Opening...");
 if (!bFileAlreadyOpened)
 DoOpenFile(objDoc1, strFilePath);
 }
</SCRIPT>
```

## Befehlstabelle für StyleVision

In den Tabellen in diesem Abschnitt werden die Namen und Identifier aller Befehle aufgelistet, die in StyleVision zur Verfügung stehen. In den einzelnen Unterabschnitten werden jeweils die Befehle aus dem entsprechenden Menü von StyleVision aufgelistet. In der Spalte ganz links wird der Text des Menübefehls angezeigt, damit Sie sofort wissen, um welche Funktionalität es sich hierbei handelt. Im letzten Unterabschnitt finden Sie all jene Befehle, die über das Hauptmenü nicht aufgerufen werden können.

Einige dieser Befehle werden je nach installierter StyleVision Edition eventuell nicht unterstützt. Eine Beschreibung, wie man die aktuelle Ressourcenstruktur und die Verfügbarkeit von Befehlen abfragt, finden Sie unter [Abfragen von StyleVision Befehlen](#).

Verwenden Sie die Befehls-Identifier mit [StyleVisionControl.QueryStatus](#) oder [StyleVisionControlDocument.QueryStatus](#) um den aktuellen Status eines Befehls zu überprüfen. Verwenden Sie [StyleVisionControl.Exec](#) oder [StyleVisionControlDocument.Exec](#) um einen Befehl auszuführen.

[Menü "Datei"](#)  
[Menü "Bearbeiten"](#)  
[Menü "Projekt"](#)  
[Menü "Ansicht"](#)  
[Menü "Einfügen"](#)  
[Menü "Tabelle"](#)  
[Menü "Authentic"](#)  
[Menü "Eigenschaften"](#)  
[Menü "Extras"](#)  
[Menü "Fenster"](#)  
[Menü "Hilfe"](#)  
[Diverse andere Menübefehle](#)

**Menü "Datei"**

Befehle aus dem Menü "Datei" (Einige davon stehen in der Professional und der Basic Edition nicht zur Verfügung):

|                                                                             |                                 |       |
|-----------------------------------------------------------------------------|---------------------------------|-------|
| Neu/Neu von XML Schema/DTD/XML...                                           | IDC_FILE_NEW_FROM_SCHEMA        | 37579 |
| Neu/Neu von DB...                                                           | IDC_FILE_NEW_FROM_DB            | 37577 |
| Neu/Neu vom XML-Spalte in DB-Tabelle...                                     | IDC_FILE_NEW_FROM_DBCCELL       | 37861 |
| Neu/Neu von XBRL-Taxonomie...                                               | IDC_FILE_NEW_FROM_XBRL_TAXONOMY | 37912 |
| Neu/Neu von HTML-Datei...                                                   | IDC_FILE_NEW_FROM_HTML          | 37578 |
| Neu/Neu (leer)                                                              | IDC_FILE_NEW_EMPTY              | 37576 |
| Öffnen...                                                                   | ID_FILE_OPEN                    | 57601 |
| Neu laden                                                                   | IDC_FILE_RELOAD                 | 36002 |
| Schließen                                                                   | ID_FILE_CLOSE                   | 57602 |
| Alle schließen                                                              | ID_FILE_CLOSE_ALL               | 37777 |
| Design speichern                                                            | ID_FILE_SAVE                    | 57603 |
| Speichern unter...                                                          | ID_FILE_SAVE_AS                 | 57604 |
| Alles speichern                                                             | ID_FILE_SAVE_ALL                | 37778 |
| XML-Authentic-Daten speichern                                               | ID_SAVE_AUTHENTIC_XML           | 32860 |
| XML-Authentic-Daten speichern unter ...                                     | ID_SAVE_AUTHENTIC_XML_AS        | 32812 |
| Generierte Dateien speichern/<br>Generierte XSLT-HTML-Datei<br>speichern... | IDC_SAVE_GEN_XSLT_HTML          | 37642 |
| Generierte Dateien speichern/<br>Generierte HTML-Datei speichern...         | IDC_SAVE_GEN_HTML               | 37636 |
| Generierte Dateien speichern/<br>Generierte XSLT-RTF-Datei<br>speichern...  | IDC_SAVE_GEN_XSLT_RTF           | 37643 |
| Generierte Dateien<br>speichern/Generierte RTF-Datei<br>speichern...        | IDC_SAVE_GEN_RTF                | 37638 |
| Generierte Dateien<br>speichern/Generierte XSLT-FO-Datei<br>speichern...    | IDC_SAVE_GEN_XSLT_FO            | 37641 |
| Generierte Dateien<br>speichern/Generierte FO-Datei<br>speichern...         | IDC_SAVE_GEN_FO                 | 37635 |

Forts....

|                                                                      |                  |       |
|----------------------------------------------------------------------|------------------|-------|
| Generierte Dateien<br>speichern/Generierte PDF-Datei<br>speichern... | IDC_SAVE_GEN_PDF | 37637 |
|----------------------------------------------------------------------|------------------|-------|

|                                                                                   |                                |       |
|-----------------------------------------------------------------------------------|--------------------------------|-------|
| Generierte Dateien speichern/Generierte XSLT-Word 2007+-Datei speichern...        | IDC_SAVE_GEN_XSLT_WORDML       | 37523 |
| Generierte Dateien speichern/Generierte Word 2007+-Datei speichern...             | IDC_SAVE_GEN_WORDML            | 37529 |
| Generierte Dateien speichern/Generiertes DB-Schema speichern...                   | IDC_SAVE_GEN_DB_SCHEMA         | 37633 |
| Generierte Dateien speichern/Generierte DB-XML-Daten speichern...                 | IDC_SAVE_GEN_DB_XML            | 37634 |
| Generierte Dateien speichern/Generiertes benutzerdefiniertes Schema speichern...  | IDC_SAVE_GEN_USERDEF_SCHEMA    | 37639 |
| Generierte Dateien speichern/Generierte benutzerdefinierte XML-Daten speichern... | IDC_SAVE_GEN_USERDEF_XML       | 37640 |
| XML-Arbeitsdatei zuweisen...                                                      | IDC_ASSIGN_WORKING_XML_FILE    | 37526 |
| Zuweisung von XML-Arbeitsdatei aufheben                                           | IDC_UNASSIGN_WORKING_XML_FILE  | 37683 |
| XML-Vorlagendatei zuweisen...                                                     | IDC_ASSIGN_TEMPLATE_XML_FILE   | 37525 |
| Zuweisung von XML-Vorlagendatei aufheben                                          | IDC_UNASSIGN_TEMPLATE_XML_FILE | 37682 |
| Eigenschaften...                                                                  | IDC_FILE_PROPERTIES            | 36027 |
| Druckvorschau...                                                                  | ID_FILE_PRINT_PREVIEW          | 57609 |
| Drucken...                                                                        | ID_FILE_PRINT                  | 57607 |
| Letzte Datei                                                                      | ID_FILE_MRU_FILE1              | 57616 |
| Beenden                                                                           | ID_APP_EXIT                    | 57665 |

**Menü "Bearbeiten"**

Befehle aus dem Menü "Bearbeiten":

|                                |                                   |       |
|--------------------------------|-----------------------------------|-------|
| Rückgängig                     | ID_EDIT_UNDO                      | 57643 |
| Wiederherstellen               | ID_EDIT_REDO                      | 57644 |
| Ausschneiden                   | ID_EDIT_CUT                       | 57635 |
| Kopieren                       | ID_EDIT_COPY                      | 57634 |
| Einfügen                       | ID_EDIT_PASTE                     | 57637 |
| Löschen                        | ID_EDIT_CLEAR                     | 57632 |
| Suchen                         | ID_EDIT_FIND                      | 57636 |
| Weitersuchen                   | ID_EDIT_FINDNEXT                  | 36802 |
| Ersetzen...                    | ID_EDIT_REPLACE                   | 57641 |
| Stylesheet-Parameter           | IDC_EDIT_PARAMETERS               | 37573 |
| Markup<br>reduzieren/erweitern | IDC_SPSGUI_COLLAPSE_EXPAND_MARKUP | 36804 |
| Alles markieren                | ID_EDIT_SELECT_ALL                | 57642 |

**Menü "Projekt"**

Befehle aus dem Menü "Projekt":

|                                                    |                                                          |       |
|----------------------------------------------------|----------------------------------------------------------|-------|
| Neues Projekt                                      | IDC_ICPROJECTGUI_NEW                                     | 37200 |
| Projekt öffnen...                                  | IDC_ICPROJECTGUI_OPEN                                    | 37201 |
| Projekt neu laden                                  | IDC_ICPROJECTGUI_RELOAD                                  | 37202 |
| Projekt schließen                                  | IDC_ICPROJECTGUI_CLOSE                                   | 37203 |
| Projekt speichern                                  | IDC_ICPROJECTGUI_SAVE                                    | 37204 |
| Dateien zu Projekt hinzufügen...                   | IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT                    | 37205 |
| Globale Ressource zu Projekt hinzufügen...         | IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT          | 37239 |
| URL zu Projekt hinzufügen...                       | IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT                      | 37206 |
| Aktive Datei zu Projekt hinzufügen                 | IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT              | 37208 |
| Aktive und verwandte Dateien zu Projekt hinzufügen | IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT | 37209 |
| Projektordner zu Projekt hinzufügen...             | IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT                   | 37210 |
| Externes Verzeichnis zum Projekt hinzufügen...     | IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT               | 37211 |
| Externes Webverzeichnis zum Projekt hinzufügen...  | IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT           | 37212 |
| Letztes Projekt                                    | IDC_ICPROJECTGUI_RECENT                                  | 37224 |

**Menü "Ansicht"**

Befehle aus dem Menü "Ansicht":

|                                                          |                                |       |
|----------------------------------------------------------|--------------------------------|-------|
| Symbolleisten/Dummy-Eintrag                              | ID_VIEW_TOOLBARS               | 37807 |
| Projekt                                                  | ID_VIEW_PROJECT                | 57682 |
| Design-Übersicht                                         | ID_VIEW_DESIGNOVERVIEW         | 37883 |
| Schema-Struktur                                          | ID_VIEW_SCHEMASOURCES          | 37805 |
| Design-Struktur                                          | ID_VIEW_DESIGNTREE             | 37804 |
| Globale Stile                                            | ID_VIEW_STYLEREPOSITORY        | 37806 |
| Kontext-Eigenschaften                                    | ID_VIEW_CONTEXTPROPERTY_COMMON | 37802 |
| Kontext-Stile                                            | ID_VIEW_CONTEXTPROPERTY_STYLES | 37803 |
| Design-Filter/Immer nur eine Vorlage auf einmal anzeigen | IDC_DESIGN_FILTER_ONE          | 37704 |
| Design-Filter/Alle Vorlagentypen anzeigen                | IDC_DESIGN_FILTER_ALL          | 37700 |
| Design-Filter/Importierte Vorlagen anzeigen              | IDC_DESIGN_FILTER_IMPORTED     | 37870 |
| Design-Filter/Hauptvorlage ein-/ausblenden               | IDC_DESIGN_FILTER_MAIN         | 37701 |
| Design-Filter/Globale Vorlagen ein-/ausblenden           | IDC_DESIGN_FILTER_MATCH        | 37702 |
| Design-Filter/Design-Fragmente ein-/ausblenden           | IDC_DESIGN_FILTER_NAMED        | 37703 |
| Design-Filter/Seiten-Layout-Vorlagen ein-/ausblenden     | IDC_DESIGN_FILTER_PAGELAYOUT   | 37705 |
| Vergrößern/Verkleinern/Vergrößern                        | IDC_SPSGUI_ZOOM_IN             | 36805 |
| Vergrößern/Verkleinern/Verkleinern                       | IDC_SPSGUI_ZOOM_OUT            | 36806 |
| Vergrößern/Verkleinern/Zoom 500 %                        | IDC_SPSGUI_ZOOM_500            | 36807 |
| Vergrößern/Verkleinern/Zoom 400 %                        | IDC_SPSGUI_ZOOM_400            | 36808 |
| Vergrößern/Verkleinern/Zoom 200 %                        | IDC_SPSGUI_ZOOM_200            | 36809 |
| Vergrößern/Verkleinern/Zoom 150 %                        | IDC_SPSGUI_ZOOM_150            | 36810 |
| Vergrößern/Verkleinern/Zoom 100 %                        | IDC_SPSGUI_ZOOM_100            | 36811 |
| Vergrößern/Verkleinern/Zoom 75 %                         | IDC_SPSGUI_ZOOM_75             | 36812 |
| Vergrößern/Verkleinern/Zoom 50 %                         | IDC_SPSGUI_ZOOM_50             | 36813 |
| Statusleiste                                             | ID_VIEW_STATUS_BAR             | 59393 |

**Menü "Einfügen"**

Befehle aus dem Menü "Einfügen" (Einige davon stehen in der Professional und der Basic Edition nicht zur Verfügung):

|                                                              |                                     |       |
|--------------------------------------------------------------|-------------------------------------|-------|
| Inhalt einfügen                                              | IDC_INSERT_CONTENTS                 | 37846 |
| Restlichen Inhalt einfügen                                   | IDC_INSERT_REST_OF_CONTENTS         | 37617 |
| Formular-Steuerelemente einfügen/Eingabefeld                 | IDC_INSERT_EDITFIELD                | 37854 |
| Formular-Steuerelemente einfügen/Mehrzeiliges Eingabefeld    | IDC_INSERT_MULTILINEEDITFIELD       | 37855 |
| Formular-Steuerelemente einfügen/Kontrollkästchen...         | IDC_INSERT_CHECKBOX                 | 37857 |
| Formular-Steuerelemente einfügen/Auswahlliste...             | IDC_INSERT_COMBOBOX                 | 37856 |
| Formular-Steuerelemente einfügen/Optionsfeld...              | IDC_INSERT_RADIOBUTTON              | 37858 |
| Formular-Steuerelemente einfügen/Schaltfläche                | IDC_INSERT_BUTTON                   | 37859 |
| Automatische Berechnung einfügen/Wert...                     | IDC_INSERT_AUTOCALC_VALUE           | 37596 |
| Automatische Berechnung einfügen/Eingabefeld...              | IDC_INSERT_AUTOCALC_FIELD           | 37594 |
| Automatische Berechnung einfügen/Mehrzeiliges Eingabefeld... | IDC_INSERT_AUTOCALC_MULTILINE_FIELD | 37595 |
| Datumswahl einfügen                                          | IDC_INSERT_DATEPICKER               | 37602 |
| Absatz einfügen                                              | IDC_INSERT_PARAGRAPH                | 37847 |
| Speziellen Absatz einfügen/Adresse                           | IDC_INSERT_ADDRESS                  | 37843 |
| Speziellen Absatz einfügen/Block (div)                       | IDC_INSERT_BLOCK                    | 37528 |
| Speziellen Absatz einfügen/Blockquote                        | IDC_INSERT_BLOCKQUOTE               | 37527 |
| Speziellen Absatz einfügen/Zentriert                         | IDC_INSERT_CENTER                   | 37534 |
| Speziellen Absatz einfügen/Fieldset                          | IDC_INSERT_FIELDSET                 | 37574 |
| Speziellen Absatz einfügen/Vorformatiert                     | IDC_INSERT_FORMATTED                | 37580 |
| Speziellen Absatz einfügen/Vorformatiert, Zeilenumbruch      | IDC_INSERT_FORMATTED_WRAP           | 37876 |

Forts....

|                                               |                     |       |
|-----------------------------------------------|---------------------|-------|
| Speziellen Absatz einfügen/Überschrift 1 (U1) | IDC_INSERT_HEADING1 | 37585 |
| Speziellen Absatz einfügen/Überschrift 2 (U2) | IDC_INSERT_HEADING2 | 37586 |
| Speziellen Absatz einfügen/Überschrift 3 (U3) | IDC_INSERT_HEADING3 | 37587 |



|                                                 |                                      |       |
|-------------------------------------------------|--------------------------------------|-------|
| Speziellen Absatz einfügen/Überschrift 4 (U4)   | IDC_INSERT_HEADING4                  | 37588 |
| Speziellen Absatz einfügen/Überschrift 5 (U5)   | IDC_INSERT_HEADING5                  | 37589 |
| Speziellen Absatz einfügen/Überschrift 6 (U6)   | IDC_INSERT_HEADING6                  | 37590 |
| Bild einfügen...                                | IDC_INSERT_IMAGE                     | 37593 |
| Horizontale Linie einfügen                      | IDC_INSERT_HORIZONTAL_LINE           | 37606 |
| Tabelle einfügen...                             | IDC_INSERT_TABLE                     | 40212 |
| Nummerierung und Aufzählungszeichen einfügen... | IDC_INSERT_FORMAT_BULLETS            | 37848 |
| Lesezeichen einfügen...                         | IDC_INSERT_BOOKMARK                  | 37530 |
| Hyperlink einfügen...                           | IDC_INSERT_HYPERLINK                 | 37849 |
| Bedingung einfügen...                           | IDC_INSERT_CONDITION                 | 37850 |
| Ausgabe-basierte Bedingung einfügen             | IDC_INSERT_CONDITION_PER_OUTPUT      | 37851 |
| Vorlage einfügen                                | IDC_INSERT_TEMPLATE                  | 40216 |
| Benutzerdefinierte Vorlage einfügen...          | IDC_INSERT_USER_DEFINED_TEMPLAT<br>E | 40184 |
| Variablenvorlage einfügen...                    | IDC_INSERT_VARIABLE_TEMPLATE         | 37531 |
| Layout Container einfügen                       | IDC_INSERT_LAYOUT_CONTAINER          | 40213 |
| Layout-Feld einfügen                            | IDC_INSERT_LAYOUT_BOX                | 40214 |
| Linie einfügen                                  | IDC_INSERT_SHAPE_LINE                | 40215 |

Forts....

|                                                                    |                               |       |
|--------------------------------------------------------------------|-------------------------------|-------|
| Inhaltsverzeichnis einfügen/Inhaltsverzeichnis...                  | IDC_INSERT_TOC_WIZARD         | 37618 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Markierung                | IDC_INSERT_TOC_WIZARD         | 37608 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.Lesezeichen (Assistent)... | IDC_INSERT_MARKER_WIZARD      | 37609 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Referenz                  | IDC_INSERT_REF                | 37613 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Referenz/Eingabetext      | IDC_INSERT_REF_ENTRY_TEXT     | 37614 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Referenz/Punktierte Linie | IDC_INSERT_REF_LEADER         | 37615 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Referenz/Seitenreferenz   | IDC_INSERT_REF_PAGE_REFERENCE | 37616 |

|                                                            |                              |       |
|------------------------------------------------------------|------------------------------|-------|
| Inhaltsverzeichnis einfügen/Hierarchische Nummerierung     | IDC_INSERT_AUTO_NUMBER_LEVEL | 37598 |
| Inhaltsverzeichnis einfügen/Sequenzielle Nummerierung      | IDC_INSERT_AUTO_NUMBER_FLAT  | 37597 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Ebene             | IDC_INSERT_LEVEL             | 37607 |
| Inhaltsverzeichnis einfügen/Inhaltsverz.-Ebenenreferenz    | IDC_INSERT_REFLEVEL          | 37860 |
| Inhaltsverzeichnis einfügen/Vorlage dient als Ebene        | IDC_ASSIGN_LEVEL             | 37524 |
| Design-Fragment einfügen/Dummy-Eintrag                     | IDC_INSERT_NAMED_TEMPLATE    | 39300 |
| Seite/Spalte/Dokumentabschnitt einfügen/Neue Seite         | IDC_INSERT_PAGE_BREAK        | 37610 |
| Seite/Spalte/Dokumentabschnitt einfügen/Seitennummer       | IDC_INSERT_PAGE_NUMBER       | 37611 |
| Seite/Spalte/Dokumentabschnitt einfügen/Gesamtseitenanzahl | IDC_INSERT_PAGE_TOTAL        | 37612 |

Forts...

|                                                                  |                                       |       |
|------------------------------------------------------------------|---------------------------------------|-------|
| Seite/Spalte/Dokumentabschnitt einfügen/Neue Spalte              | IDC_INSERT_COLUMN_BREAK               | 37864 |
| Seite/Spalte/Dokumentabschnitt einfügen/Neuer Dokumentabschnitt  | IDC_INSERT_DOCUMENT_SECTION           | 37951 |
| DB-Steuerelement einfügen/Navigation                             | IDC_INSERT_DBNAVIGATION               | 37603 |
| DB-Steuerelement einfügen/Navigation + Gehe zu                   | IDC_INSERT_DBNAVIGATIONGOTO           | 37604 |
| DB-Steuerelement einfügen/Abfrageschaltfläche                    | IDC_INSERT_DBQUERY                    | 37605 |
| XBRL-Element einfügen/Kontext                                    | IDC_INSERT_XBRL_CONTEXT               | 37942 |
| XBRL-Element einfügen/Zeitraum                                   | IDC_INSERT_XBRL_PERIOD                | 37909 |
| XBRL-Element einfügen/Zeitraumbeginn                             | IDC_INSERT_XBRL_PERIOD_START          | 37947 |
| XBRL-Element einfügen/Zeitraumende                               | IDC_INSERT_XBRL_PERIOD_END            | 37948 |
| XBRL-Element einfügen/Identifizier                               | IDC_INSERT_XBRL_IDENTIFIER            | 37910 |
| XBRL-Element einfügen/Unit                                       | IDC_INSERT_XBRL_UNIT                  | 37946 |
| XBRL-Element einfügen/Fußnote                                    | IDC_INSERT_XBRL_FOOTNOTE              | 37911 |
| Benutzerdefiniertes Element einfügen/Benutzerdefiniertes Element | IDC_STYLEVISIONGUI_INSERT_USERXMLELEM | 37908 |
| Benutzerdefiniertes Element einfügen/Benutzerdefinierter Block   | IDC_STYLEVISIONGUI_INSERT_USERXMLTEXT | 37920 |

**Menü "Einschließen in"**

Befehle aus dem Menü **Einschließen in** (einige davon sind in der Pro- und der Std-Edition eventuell nicht verfügbar):

|                                                |                                         |       |
|------------------------------------------------|-----------------------------------------|-------|
| Vorlage...                                     | IDC_ENCLOSE_WITH_TEMPLATES              | 40180 |
| Benutzerdefinierte Vorlage...                  | IDC_ENCLOSE_WITH_USER_DEFINED_TEMPLATES | 40217 |
| Variablenvorlage...                            | IDC_ENCLOSE_WITH_VARIABLE_TEMPLATES     | 37520 |
| Absatz                                         | IDC_ENCLOSE_PARAGRAPH                   | 37627 |
| Spezieller Absatz/Adresse                      | IDC_ENCLOSE_ADDRESS                     | 40194 |
| Spezieller Absatz/Block (div)                  | IDC_ENCLOSE_BLOCK                       | 40195 |
| Spezieller Absatz/Blockquote                   | IDC_ENCLOSE_BLOCKQUOTE                  | 40196 |
| Spezieller Absatz/Zentriert                    | IDC_ENCLOSE_CENTER                      | 40197 |
| Spezieller Absatz/Fieldset                     | IDC_ENCLOSE_FIELDSET                    | 40198 |
| Spezieller Absatz/Vorformatiert                | IDC_ENCLOSE_FORMATTED                   | 40199 |
| Spezieller Absatz/Vorformatiert, Zeilenumbruch | IDC_ENCLOSE_FORMATTED_WRAP              | 40200 |
| Spezieller Absatz/Überschrift (Ü1)             | IDC_ENCLOSE_HEADING1                    | 40201 |
| Spezieller Absatz/Überschrift 2 (Ü2)           | IDC_ENCLOSE_HEADING2                    | 40202 |
| Spezieller Absatz/Überschrift 3 (Ü3)           | IDC_ENCLOSE_HEADING3                    | 40203 |
| Spezieller Absatz/Überschrift 4 (Ü4)           | IDC_ENCLOSE_HEADING4                    | 40204 |
| Spezieller Absatz/Überschrift 5 (Ü5)           | IDC_ENCLOSE_HEADING5                    | 40205 |
| Spezieller Absatz/Überschrift 6 (Ü6)           | IDC_ENCLOSE_HEADING6                    | 40206 |
| Nummerierung und Aufzählungszeichen...         | IDC_ENCLOSE_FORMAT_BULLETS              | 37581 |
| Lesezeichen...                                 | IDC_ENCLOSE_BOOKMARK                    | 40207 |
| Hyperlink...                                   | IDC_ENCLOSE_HYPERLINK                   | 37620 |
| Bedingung...                                   | IDC_ENCLOSE_CONDITION                   | 37599 |

Forts....

|                                        |                                  |       |
|----------------------------------------|----------------------------------|-------|
| Ausgabe-basierte Bedingung             | IDC_ENCLOSE_CONDITION_PER_OUTPUT | 37600 |
| Inhaltsverz.-Markierung                | IDC_ENCLOSE_MARKER               | 40210 |
| Inhaltsverz.-Markierung (Assistent)... | IDC_ENCLOSE_MARKER_WIZARD        | 40211 |

|                                |                                           |       |
|--------------------------------|-------------------------------------------|-------|
| Inhaltsverz.-Ebene             | IDC_ENCLOSE_LEVEL                         | 40208 |
| Inhaltsverz.-Ebenenreferenz    | IDC_ENCLOSE_REFLEVEL                      | 40209 |
| Benutzerdefiniertes Element... | IDC_ENCLOSE_WITH_USER_DEFINED_XML_ELEMENT | 40222 |

**Menü "Tabelle"**

Befehle aus dem Menü "Tabelle":

|                                           |                                |       |
|-------------------------------------------|--------------------------------|-------|
| Tabelle einfügen                          | IDC_TABLE_INSERT_FIX           | 40212 |
| Tabelle löschen                           | IDC_TABLE_DELETE               | 37658 |
| Spalte für Tabellenüberschrift hinzufügen | IDC_TABLE_ADD_HEAD<br>ERCOL    | 37888 |
| Spalte für Tabellenfußzeile hinzufügen    | IDC_TABLE_ADD_FOOT<br>ERCOL    | 37889 |
| Tabellenkopfzeile hinzufügen              | IDC_TABLE_ADD_HEAD<br>ERROW    | 37900 |
| Tabellenfußzeile hinzufügen               | IDC_TABLE_ADD_FOOT<br>ERROW    | 37901 |
| Zeile anhängen                            | IDC_TABLE_APPEND_ROW           | 37657 |
| Spalte anhängen                           | IDC_TABLE_APPEND_COL           | 37656 |
| Zeile einfügen                            | IDC_TABLE_INSERT_ROW           | 37664 |
| Spalte einfügen                           | IDC_TABLE_INSERT_COL           | 37662 |
| Zeile löschen                             | IDC_TABLE_DELETE_ROW           | 37660 |
| Spalte löschen                            | IDC_TABLE_DELETE_COL           | 37659 |
| Zelle links verbinden                     | IDC_TABLE_JOIN_LEFT            | 37666 |
| Zelle rechts verbinden                    | IDC_TABLE_JOIN_RIGHT           | 37667 |
| Zelle unten verbinden                     | IDC_TABLE_JOIN_DOWN            | 37665 |
| Zelle oben verbinden                      | IDC_TABLE_JOIN_UP              | 37668 |
| Zelle horizontal teilen                   | IDC_TABLE_SPLIT_HORZ           | 37670 |
| Zelle vertikal teilen                     | IDC_TABLE_SPLIT_VERT           | 37671 |
| Zellumrandung anzeigen                    | IDC_TABLE_SHOW_ZERO_<br>BORDER | 37669 |
| Tabellen-Markup-Code anzeigen             | IDC_TABLE_SHOW_UIC<br>ELLS     | 37887 |
| Tabelleneigenschaften...                  | IDC_TABLE_EDIT_PROPER<br>TIES  | 37661 |

**Menü "Authentic"**

Befehle aus dem Menü "Authentic" (Einige davon stehen in der Professional und der Basic Edition nicht zur Verfügung):

|                                                       |                                  |       |
|-------------------------------------------------------|----------------------------------|-------|
| Textstatussymbole...                                  | IDC_TEXT_STATE_ICONS             | 37672 |
| CALS / HTML-Tabellen...                               | IDC_CALS_HTML_TABLES             | 37533 |
| Datumswahl automatisch hinzufügen                     | IDC_TOGGLE_DATEPICKER_AUTOINSERT | 37674 |
| DB-Steuerelemente automatisch hinzufügen              | IDC_TOGGLE_DBCONTROL_AUTOINSERT  | 37675 |
| Authentic-Ansicht neu laden                           | IDC_AUTHENTICGUI_RELOAD          | 32800 |
| XML validieren                                        | IDC_VALIDATE                     | 32954 |
| Neue Zeile mit XML-Daten für Bearbeitung auswählen... | IDC_CHANGE_WORKING_DB_XML_CELL   | 32861 |
| XML-Entities definieren...                            | IDC_DEFINE_ENTITIES              | 32805 |
| Markup-Symbole ausblenden                             | IDC_MARKUP_HIDE                  | 32855 |
| Kleine Markup-Symbole einblenden                      | IDC_MARKUP_SMALL                 | 32858 |
| Große Markup-Symbole einblenden                       | IDC_MARKUP_LARGE                 | 32856 |
| Gemischte Markup-Symbole einblenden                   | IDC_MARKUP_MIXED                 | 32857 |
| Zeile anhängen                                        | IDC_ROW_APPEND                   | 32806 |
| Zeile einfügen                                        | IDC_ROW_INSERT                   | 32809 |
| Zeile kopieren                                        | IDC_ROW_DUPLICATE                | 32808 |
| Zeile nach oben                                       | IDC_ROW_MOVE_UP                  | 32811 |
| Zeile nach unten                                      | IDC_ROW_MOVE_DOWN                | 32810 |
| Zeile löschen                                         | IDC_ROW_DELETE                   | 32807 |

**Menü "Datenbank"**

Befehle aus dem Menü **Datenbank**:

|                            |                     |       |
|----------------------------|---------------------|-------|
| Datenbank<br>abfragen      | ID_VIEW_DBQUERY     | 37818 |
| DB-Filter<br>bearbeiten... | IDC_EDIT_DBFILTERS  | 37571 |
| DB-Filter löschen          | IDC_CLEAR_DBFILTERS | 37547 |

**Menü "Eigenschaften"**

Befehle aus dem Menü "Eigenschaften":

|                                                      |                                |       |
|------------------------------------------------------|--------------------------------|-------|
| Nummerierung und<br>Aufzählungszeichen bearbeiten... | IDC_EDIT_BULLETS_AND_NUMBERING | 37839 |
| Vordefinierte<br>Eingabe-Formatierungsstrings...     | IDC_FORMAT_PREDEFINES          | 37582 |



**Menü "Extras"**

Befehle aus dem Menü "Extras":

|                                                  |                                    |       |
|--------------------------------------------------|------------------------------------|-------|
| Rechtschreibung...                               | IDC_SPSGUI_SPELL_CHECK             | 36800 |
| Rechtschreibung -<br>Optionen...                 | IDC_SPSGUI_SPELL_OPTIONS           | 36801 |
| Globale Ressourcen                               | IDC_GLOBALRESOURCES                | 37401 |
| Aktive Konfiguration/<plugin<br>not loaded>      | IDC_GLOBALRESOURCES_SUBMENUENTRY1  | 37408 |
| Anpassen...                                      | IDC_CUSTOMIZE_VIEW                 | 37560 |
| Symbolleisten und Fenster<br>wiederherstellen... | IDC_APP_RESET_TOOLBARS_AND_WINDOWS | 32956 |
| Optionen...                                      | IDC_TOOLS_OPTIONS                  | 37676 |

**Menü "Fenster"**

Befehle aus dem Menü "Fenster":

|                        |                     |       |
|------------------------|---------------------|-------|
| Überlappend            | ID_WINDOW_CASCADE   | 57650 |
| Horizontal<br>anordnen | ID_WINDOW_TILE_HORZ | 57651 |
| Vertikal<br>anordnen   | ID_WINDOW_TILE_VERT | 59405 |
| Symbole<br>anordnen    | ID_WINDOW_ARRANGE   | 57649 |

**Menü "Hilfe"**

Befehle aus dem Menü "Hilfe":

|                                 |                        |       |
|---------------------------------|------------------------|-------|
| Inhaltsverzeichnis              | ID_HELP_FINDER         | 57667 |
| Index...                        | ID_HELP_INDEX          | 57666 |
| Suchen...                       | ID_HELP_SEARCH         | 36036 |
| Software-Aktivierung...         | IDC_ACTIVATION         | 36025 |
| Bestellformular...              | ID_HELP_ORDERFORM      | 36034 |
| Registrieren...                 | ID_HELP_REGISTRATION   | 36035 |
| Auf Updates überprüfen...       | IDC_CHECK_FOR_UPDATES  | 36026 |
| StyleVision-Produktvergleich... | IDC_PRODUCT_COMPARISON | 32955 |
| Support Center...               | ID_HELP_SUPPORTCENTER  | 36037 |
| Fragen und Antworten im Web...  | ID_HELP_FAQ            | 36032 |
| Komponenten downloaden...       | ID_HELP_COMPONENTS     | 36031 |
| StyleVision im Internet...      | ID_HELP_INTERNET       | 36033 |
| StyleVision Training...         | ID_HELP_TRAINING       | 36038 |
| Über StyleVision...             | ID_APP_ABOUT           | 57664 |

**Diverse andere Menübefehle**

Diverse andere Menü- und Kontextmenübefehle (Einige davon stehen in der Professional und der Basic Edition nicht zur Verfügung):

|                                 |                               |       |
|---------------------------------|-------------------------------|-------|
| Aktiv                           | IDC_ACTIVATE_PART_TOGGLE      | 37532 |
| Datenbank                       | IDC_ADDRESOURCE_DATABASE      | 37405 |
| Datei                           | IDC_ADDRESOURCE_FILE          | 37403 |
| Ordner                          | IDC_ADDRESOURCE_FOLDER        | 37404 |
| Verzweigung kopieren            | IDC_ADD_COPY_OPTION           | 37515 |
| Neue Verzweigung hinzufügen     | IDC_ADD_NEW_OPTION            | 37516 |
| Element anhängen                | IDC_ADD_NEXT_NODE             | 37517 |
| Element einfügen                | IDC_ADD_PREV_NODE             | 37518 |
| Child-Element hinzufügen        | IDC_ADD_SUB_NODE              | 37519 |
|                                 | IDC_ALIGN_CENTER              | 37826 |
|                                 | IDC_ALIGN_JUSTIFY             | 37828 |
|                                 | IDC_ALIGN_LEFT                | 37825 |
|                                 | IDC_ALIGN_RIGHT               | 37827 |
| XML-Schema/DTD-Datei zuweisen   | IDC_ASSIGN_SCHEMA_FILE        | 37886 |
| XBRL-Vorlagendatei zuweisen     | IDC_ASSIGN_TEMPLATE_XBRL_FILE | 37938 |
| XBRL-Arbeitsdatei zuweisen      | IDC_ASSIGN_WORKING_XBRL_FILE  | 37939 |
| XBRL-Taxonomiedatei zuweisen... | IDC_ASSIGN_XBRL_TAXONOMY_FILE | 37935 |
| Authentic                       | IDC_AUTHENTIC                 | 37692 |
|                                 | IDC_BOLD                      | 37822 |
|                                 | IDC_BULLET_ORDERED            | 37829 |
|                                 | IDC_BULLET_UNORDERED          | 37830 |
| Seitenränder ändern...          | IDC_CHANGE_PAGE_MARGINS       | 37535 |
| Dummy entry                     | IDC_CHANGE_TEMPLATE_MATCH     | 39320 |
| Inhalt                          | IDC_CHANGE_TO_ALL_CONTENTS    | 37536 |
| Schaltfläche                    | IDC_CHANGE_TO_BUTTON          | 37537 |
| Kontrollkästchen...             | IDC_CHANGE_TO_CHECKBOX        | 37538 |
| Eingabefeld                     | IDC_CHANGE_TO_FIELD           | 37539 |
| Bild...                         | IDC_CHANGE_TO_IMAGE           | 37540 |
| Mehrzeiliges Eingabefeld        | IDC_CHANGE_TO_MULTILINE_FIELD | 37542 |
| Absatz                          | IDC_CHANGE_TO_PARAGRAPH       | 37543 |

Forts...

|                                          |                                                 |       |
|------------------------------------------|-------------------------------------------------|-------|
| XPath-Filter löschen                     | IDC_CLEAR_FILTER                                | 39014 |
| Eingabetext mittels XPath erstellen      | IDC_CONSTRUCT_ENTRY_TEXT_USING_XPATH            | 37831 |
| Alles in globale Ressourcen konvertieren | IDC_CONVERT_ALL_DATASOURCES_TO_GLOBAL_RESOURCES | 36687 |
| In globale Ressource konvertieren        | IDC_CONVERT_DATASOURCE_TO_GLOBAL_RESOURCES      | 36684 |
| Globale Vorlage lokal kopieren           | IDC_CONVERT_FROM_GLOBAL_TEMPLATE                | 37548 |
| In Attribut konvertieren                 | IDC_CONVERT_TO_ATTRIBUTE                        | 37549 |
| In Element konvertieren                  | IDC_CONVERT_TO_ELEMENT                          | 37550 |
| Als globale Vorlage definieren           | IDC_CONVERT_TO_GLOBAL_TEMPLATE                  | 37551 |
| Globale Vorlage verwenden                | IDC_CONVERT_TO_GLOBAL_TEMPLATE_WITH_EXISTS      | 37552 |
| Globale Ressource in Projekt kopieren    | IDC_COPY_GLOBAL_RESOURCES_TO_PROJECT            | 36685 |
| Schaltfläche erstellen                   | IDC_CREATE_BUTTON                               | 37553 |
| Kontrollkästchen erstellen...            | IDC_CREATE_CHECKBOX                             | 37554 |
| Bild erstellen...                        | IDC_CREATE_DYNAMIC_IMAGE                        | 37555 |
| Auswahlliste erstellen...                | IDC_CREATE_DYNAMIC_SELECTION                    | 37556 |
| Eingabefeld erstellen                    | IDC_CREATE_FIELD                                | 37557 |
| Hyperlink erstellen                      | IDC_CREATE_HYPERLINK                            | 37842 |
| Als Hyperlink einfügen                   | IDC_CREATE_HYPERLINK_FROM_FILE                  | 37881 |
| Als Bild einfügen                        | IDC_CREATE_IMAGE_FROM_FILE                      | 37882 |
| Mehrzeiliges Eingabefeld erstellen       | IDC_CREATE_MULTILINE_FIELD                      | 37558 |
| Neues Design erstellen...                | IDC_CREATE_NEW_DESIGN_FROM_FILE                 | 37878 |
| Optionsfeld erstellen...                 | IDC_CREATE_RADIOBUTTON                          | 37559 |
| Löschen                                  | IDC_DELETE_NODE                                 | 37561 |
| Verzweigung erstellen                    | IDC_DELETE_OPTION                               | 37562 |
| Umschlagseite hinzufügen                 | IDC_DESIGNTREE_ADD_COVERPAGE                    | 37563 |
| Fußzeile hinzufügen                      | IDC_DESIGNTREE_ADD_FOOTER_ALL                   | 37564 |
| Fußzeile hinzufügen (gerade)             | IDC_DESIGNTREE_ADD_FOOTER_EVEN                  | 37565 |
| Fußzeile hinzufügen (ungerade)           | IDC_DESIGNTREE_ADD_FOOTER_ODD                   | 37566 |
| Kopfzeile hinzufügen                     | IDC_DESIGNTREE_ADD_HEADER_ALL                   | 37567 |

|                                 |                                |       |
|---------------------------------|--------------------------------|-------|
| Kopfzeile hinzufügen (gerade)   | IDC_DESIGNTREE_ADD_HEADER_EVEN | 37568 |
| Kopfzeile hinzufügen (ungerade) | IDC_DESIGNTREE_ADD_HEADER_ODD  | 37569 |

Forts....

|                                            |                                                 |       |
|--------------------------------------------|-------------------------------------------------|-------|
| Optionsfeld...                             | IDC_CHANGE_TO_RADIO_BUTTON                      | 37544 |
| Restlicher Inhalt                          | IDC_CHANGE_TO_REST_OF_CONTENTS                  | 40220 |
| Auswahlliste...                            | IDC_CHANGE_TO_SELECTION                         | 37545 |
| Tabelle...                                 | IDC_CHANGE_TO_TABLE                             | 37546 |
| XPath-Filter löschen                       | IDC_CLEAR_FILTER                                | 37925 |
|                                            | IDC_COMMAND_REFRESH_DATASOURCE                  | 36674 |
| Zu Datenvergleichsdokument hinzufügen      | IDC_COMMAND_SHOW_IN_EXISTING_COMPARISON         | 36542 |
| Zu Schemavergleichsdokument hinzufügen     | IDC_COMMAND_SHOW_IN_EXISTING_SCHEMA_COMPARISON  | 36690 |
| In neuem Datenvergleichsdokument anzeigen  | IDC_COMMAND_SHOW_IN_NEW_COMPARISON              | 36541 |
| In neuem Schemavergleichsdokument anzeigen | IDC_COMMAND_SHOW_IN_NEW_SCHEMA_COMPARISON       | 36691 |
| Eingabetext mittels XPath erstellen        | IDC_CONSTRUCT_ENTRY_TEXT_USING_XPATH            | 37831 |
| Alles in globale Ressourcen konvertieren   | IDC_CONVERT_ALL_DATASOURCES_TO_GLOBAL_RESOURCES | 36687 |
| In globale Ressource konvertieren          | IDC_CONVERT_DATASOURCE_TO_GLOBAL_RESOURCES      | 36684 |
| Globale Vorlage lokal kopieren             | IDC_CONVERT_FROM_GLOBAL_TEMPLATE                | 37548 |
| In Attribut konvertieren                   | IDC_CONVERT_TO_ATTRIBUTE                        | 37549 |
| In Element konvertieren                    | IDC_CONVERT_TO_ELEMENT                          | 37550 |
| Als globale Vorlage definieren             | IDC_CONVERT_TO_GLOBAL_TEMPLATE                  | 37551 |
| Globale Vorlage verwenden                  | IDC_CONVERT_TO_GLOBAL_TEMPLATE_WITH_EXISTS      | 37552 |
| Globale Ressource in Projekt kopieren      | IDC_COPY_GLOBAL_RESOURCES_TO_PROJECT            | 36685 |
| Schaltfläche erstellen                     | IDC_CREATE_BUTTON                               | 37553 |
| Kontrollkästchen erstellen...              | IDC_CREATE_CHECKBOX                             | 37554 |
| Bild erstellen...                          | IDC_CREATE_DYNAMIC_IMAGE                        | 37555 |
| Auswahlliste erstellen...                  | IDC_CREATE_DYNAMIC_SELECTION                    | 37556 |
| Eingabefeld erstellen                      | IDC_CREATE_FIELD                                | 37557 |
| Hyperlink erstellen                        | IDC_CREATE_HYPERLINK                            | 37842 |

|                                    |                                 |       |
|------------------------------------|---------------------------------|-------|
| Als Hyperlink einfügen             | IDC_CREATE_HYPERLINK_FROM_FILE  | 37881 |
| Als Bild einfügen                  | IDC_CREATE_IMAGE_FROM_FILE      | 37882 |
| Mehrzeiliges Eingabefeld erstellen | IDC_CREATE_MULTILINE_FIELD      | 37558 |
| Neues Design erstellen...          | IDC_CREATE_NEW_DESIGN_FROM_FILE | 37878 |
| Optionsfeld erstellen...           | IDC_CREATE_RADIOBUTTON          | 37559 |

Forts....

|                                                                        |                                             |       |
|------------------------------------------------------------------------|---------------------------------------------|-------|
| Alle Änderungen beim Speichern von XML-Datenbank-Dokumenten übernehmen | IDC_DBQUERY_AUTOCOMMIT                      | 38012 |
| Datenbankabfragefenster ausblenden                                     | IDC_DBQUERY_AUTOHIDE                        | 38011 |
|                                                                        | IDC_DBQUERY_OPTIONS                         | 38005 |
|                                                                        | IDC_DBQUERY_SQLVIEW_EXECUTE                 | 38006 |
|                                                                        | IDC_DBQUERY_SQLVIEW_EXECUTE_EDIT            | 38007 |
|                                                                        | IDC_DBQUERY_SQLVIEW_OPEN                    | 38004 |
| In DatabaseSpy öffnen                                                  | IDC_DBQUERY_SQLVIEW_OPENINDBSPY             | 38002 |
|                                                                        | IDC_DBQUERY_SQLVIEW_SAVE                    | 38003 |
|                                                                        | IDC_DBQUERY_SQLVIEW_STOPEXECUTION           | 38009 |
|                                                                        | IDC_DBQUERY_TOGGLEBROWSER                   | 38001 |
|                                                                        | IDC_DBQUERY_TOGGLERESULT                    | 38000 |
| Parameter bearbeiten...                                                | IDC_DEFINE_NAMED_TEMPLATE_ACTUAL_PARAMETERS | 39715 |
| Parameter definieren...                                                | IDC_DEFINE_NAMED_TEMPLATE_FORMAL_PARAMETERS | 39716 |
| Variablen definieren...                                                | IDC_DEFINE_VARIABLES                        | 40185 |
| Löschen                                                                | IDC_DELETE_NODE                             | 37561 |
| Verzweigung löschen                                                    | IDC_DELETE_OPTION                           | 37562 |
| Fußzeile hinzufügen                                                    | IDC_DESIGNTREE_ADD_FOOTER_ALL               | 37564 |
| Add Footer Even                                                        | IDC_DESIGNTREE_ADD_FOOTER_EVEN              | 37565 |
| Add Footer First                                                       | IDC_DESIGNTREE_ADD_FOOTER_FIRST             | 40119 |
| Add Footer Last                                                        | IDC_DESIGNTREE_ADD_FOOTER_LAST              | 40123 |
| Add Footer Odd                                                         | IDC_DESIGNTREE_ADD_FOOTER_ODD               | 37566 |
| Kopfzeile hinzufügen                                                   | IDC_DESIGNTREE_ADD_HEADER_ALL               | 37567 |
| Add Header Even                                                        | IDC_DESIGNTREE_ADD_HEADER_EVEN              | 37568 |
| Add Header First                                                       | IDC_DESIGNTREE_ADD_HEADER_FIRST             | 39995 |
| Add Header Last                                                        | IDC_DESIGNTREE_ADD_HEADER_LAST              | 40075 |

|                            |                                      |       |
|----------------------------|--------------------------------------|-------|
| Add Header Odd             | IDC_DESIGNTREE_ADD_HEADER_ODD        | 37569 |
| Hauptvorlage hinzufügen    | IDC_DESIGNTREE_ADD_MAIN_TEMPLATE     | 37866 |
| Modul hinzufügen...        | IDC_DESIGNTREE_ADD_MODULE            | 37865 |
| Design Fragment hinzufügen | IDC_DESIGNTREE_ADD_NAMED_TEMPLATE    | 37570 |
| Design Fragment erstellen  | IDC_DESIGNTREE_CREATE_NAMED_TEMPLATE | 40181 |
| Gehe zu Definition         | IDC_DESIGNTREE_GOTO_DEF              | 37869 |
| Nach unten                 | IDC_DESIGNTREE_MOVE_DOWN             | 37868 |

Forts...

|                                                      |                                                  |       |
|------------------------------------------------------|--------------------------------------------------|-------|
| Nach oben                                            | IDC_DESIGNTREE_MOVE_UP                           | 37867 |
| Eine Ebene nach vorne                                | IDC_DESIGN_BRING_FORWARD                         | 37961 |
| In den Vordergrund                                   | IDC_DESIGN_BRING_TO_FRONT                        | 37959 |
| Eigenschaften der Grafikvorlage bearbeiten...        | IDC_DESIGN_EDIT_BLUEPRINT_IMAGE_PROPERTIES       | 40187 |
| Linieigenschaften bearbeiten...                      | IDC_DESIGN_EDIT_LINE_PROPERTIES                  | 40188 |
| Am Raster ausrichten aktivieren                      | IDC_DESIGN_ENABLE_SNAP_TO_GRID                   | 37963 |
| Design-Filter                                        | IDC_DESIGN_FILTER                                | 37699 |
|                                                      | IDC_DESIGN_FILTER_TEMPLATE_COMBO                 | 37706 |
| Automatisch an Papiergröße anpassen                  | IDC_DESIGN_LAYOUTCONTAINER_AUTOFIT_TO_PAPER_SIZE | 40190 |
| CSS-Datei hinzufügen...                              | IDC_DESIGN_OVERVIEW_ADD_CSS                      | 37890 |
| Parameter hinzufügen...                              | IDC_DESIGN_OVERVIEW_ADD_PARAMETER                | 37793 |
| XSLT-Datei hinzufügen...                             | IDC_DESIGN_OVERVIEW_ADD_XSLT                     | 37871 |
| DB-Verbindung ändern                                 | IDC_DESIGN_OVERVIEW_CHANGE_DB_CONNECTION         | 37891 |
| Modul öffnen                                         | IDC_DESIGN_OVERVIEW_OPEN_MODULE                  | 37931 |
| Aktiv                                                | IDC_DESIGN_OVERVIEW_PAGE_LAYOUT_ACTIVE           | 37922 |
| Seiteneigenschaften bearbeiten...                    | IDC_DESIGN_OVERVIEW_PAGE_LAYOUT_PROPERTIES       | 37921 |
| DB-Tabellen und Ansichten auswählen...               | IDC_DESIGN_OVERVIEW_SELECT_DB_TABLES_AND_VIEWS   | 37892 |
| XML-Schema von DB auswählen ...                      | IDC_DESIGN_OVERVIEW_SELECT_SCHEMA_FROM_DB        | 37621 |
| XML-Arbeitsdatei-DB-Feld auswählen...                | IDC_DESIGN_OVERVIEW_SELECT_WORKING_XML_DB_FIELD  | 37622 |
| Eigenschaften der XBRL-Taxonomiequelle bearbeiten... | IDC_DESIGN_OVERVIEW_XBRL_TAXONOMY_PROPERTIES     | 37945 |
| Eine Ebene nach hinten                               | IDC_DESIGN_SEND_BACKWARD                         | 37960 |



|                                             |                                 |       |
|---------------------------------------------|---------------------------------|-------|
| In den Hintergrund                          | IDC_DESIGN_SEND_TO_BACK         | 37958 |
| Raster anzeigen                             | IDC_DESIGN_SHOW_GRID            | 37962 |
| Große Design Markup-Symbole einblenden      | IDC_DESIGN_SHOW_LARGE_MARKUPS   | 37954 |
| Kleine Design Markup-Symbole einblenden     | IDC_DESIGN_SHOW_SMALL_MARKUPS   | 37953 |
| Größe des Layout-Felds automatisch anpassen | IDC_DESIGN_TEXTBOX_AUTORESIZE   | 40189 |
|                                             | IDC_DEVHELPER_ADDERRORTTESTCASE | 36014 |
|                                             | IDC_DEVHELPER_ADDTESTCASE       | 36005 |
|                                             | IDC_DEVHELPER_HTMLVIEW_SRC      | 36803 |
| Edit Alt...                                 | IDC_EDIT_ALT                    | 37841 |

Forts....

|                                             |                                |       |
|---------------------------------------------|--------------------------------|-------|
| Edit Alt...                                 | IDC_EDIT_ALT                   | 37841 |
| Format bearbeiten...                        | IDC_EDIT_AUTO_NUMBER_FORMAT    | 37834 |
| Lesezeichennamen bearbeiten...              | IDC_EDIT_BOOKMARK_NAME         | 37838 |
| Geprüfte Werte bearbeiten...                | IDC_EDIT_CHECKED_VALUES        | 37835 |
| Eingabewerte der Auswahlliste bearbeiten... | IDC_EDIT_COMBOBOX_ENTRY_VALUES | 37836 |
| XPath-Filter bearbeiten...                  | IDC_EDIT_FILTER                | 37924 |
| Globale Ressource bearbeiten...             | IDC_EDIT_GLOBAL_RESOURCES      | 36686 |
| Eingabeformatierung bearbeiten...           | IDC_EDIT_INPUT_FORMATTING      | 37833 |
| Datei in XMLSpy bearbeiten                  | IDC_EDIT_IN_XMLSPY             | 37575 |
| Namen bearbeiten....                        | IDC_EDIT_NAME                  | 37840 |
| Bedingung bearbeiten                        | IDC_EDIT_OPTION_CONDITION      | 37572 |
| Absatztyp bearbeiten...                     | IDC_EDIT_PARAGRAPH_TYPE        | 37845 |
| Geltungsbereich bearbeiten...               | IDC_EDIT_SCOPE                 | 37844 |
| Vorlagenauswahl ändern...                   | IDC_EDIT_TEMPLATE_MATCH        | 40182 |
| XML-Node bearbeiten aktualisieren...        | IDC_EDIT_UPDATE_XML_NODE       | 37853 |
| URL bearbeiten...                           | IDC_EDIT_URL                   | 37837 |
| Benutzerdefiniertes Element bearbeiten...   | IDC_EDIT_USERXMLELEM           | 37949 |
| Benutzerdefinierten Block bearbeiten..      | IDC_EDIT_USERXMLTEXT           | 37950 |

|                                              |                                             |       |
|----------------------------------------------|---------------------------------------------|-------|
| XBRL-Label bearbeiten...                     | IDC_EDIT_XBRL_LABEL                         | 37933 |
| XPath bearbeiten...                          | IDC_EDIT_XPATH                              | 37832 |
| Einchecken...                                | IDC_FILE_CHECK_IN                           | 32951 |
| Auschecken...                                | IDC_FILE_CHECK_OUT                          | 32952 |
| Auschecken rückgängig...                     | IDC_FILE_UNDO_CHECK_OUT                     | 32953 |
| Formatierung                                 | IDC_FORMAT                                  | 37707 |
| Durchsuchen...                               | IDC_GLOBALRESOURCESUI_CHOOSEFILEASFILE      | 37420 |
| Andere globale Ressource auswählen...        | IDC_GLOBALRESOURCESUI_CHOOSEFILEASGR        | 37419 |
| Konfiguration hinzufügen                     | IDC_GLOBALRESOURCESUI_DETAILS_ADDCONFIG     | 37423 |
| Konfigurationskopie hinzufügen               | IDC_GLOBALRESOURCESUI_DETAILS_ADDCONFIGCOPY | 37424 |
| Konfiguration löschen                        | IDC_GLOBALRESOURCESUI_DETAILS_DELETECONFIG  | 37425 |
| Konfiguration der aktiven globalen Ressource | IDC_GLOBALRESOURCES_ACTIVECONFIG            | 37400 |
| Löschen                                      | IDC_GLOBALRESOURCES_MAINDLG_DELETERESOURCE  | 37422 |

Forts...

|                                       |                                       |       |
|---------------------------------------|---------------------------------------|-------|
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY2     | 37409 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY3     | 37410 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY4     | 37411 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY5     | 37412 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY6     | 37413 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY7     | 37414 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY8     | 37415 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY9     | 37416 |
| <placeholder>                         | IDC_GLOBALRESOURCES_SUBMENUENTRY_LAST | 37417 |
| Gehe zu nächstem Lesezeichen          | IDC_GOTONEXTBOOKMARK                  | 37583 |
| Gehe zu vorherigem Lesezeichen        | IDC_GOTOPREVBOOKMARK                  | 37584 |
| Authentic-Eigenschaften bearbeiten... | IDC_GOTO_AUTHENTIC_PROPERTIES         | 37852 |
| Gruppieren nach...                    | IDC_GROUP                             | 37874 |
|                                       | IDC_HYPERLINK                         | 37591 |
|                                       | IDC_HYPERLINK_PROPERTIES              | 37592 |
|                                       | IDC_ICDBWNS_ADD_SELECT_STATEMENT      | 36663 |

|                                          |                                       |       |
|------------------------------------------|---------------------------------------|-------|
|                                          | IDC_ICDBWNS_DELETE_DSN                | 36666 |
| SELECT-Anweisung bearbeiten...           | IDC_ICDBWNS_EDIT_LOCALVIEW            | 36676 |
|                                          | IDC_ICDBWNS_MANAGE_LOCAL_RELATIONS    | 36688 |
|                                          | IDC_ICDBWNS_NEW_DSN                   | 36661 |
| Datei suchen...                          | IDC_ICDBWNS_OPENINEXPLORER            | 36540 |
|                                          | IDC_ICDBWNS_REFRESH_DSN               | 36667 |
| SELECT-Anweisung entfernen               | IDC_ICDBWNS_REMOVE_LOCALVIEW          | 36680 |
|                                          | IDC_ICDBWND_BACK_COLOR                | 36543 |
|                                          | IDC_ICDBWND_BOLD                      | 36544 |
| SELECT-Anweisung hinzufügen...           | IDC_ICDBWND_CREATE_LOCALVIEW          | 36677 |
|                                          | IDC_ICDBWND_EDIT_DSN                  | 36662 |
| SELECT-Anweisung erzeugen und hinzufügen | IDC_ICDBWND_GENEATE_SQL_FOR_LOCALVIEW | 36678 |
|                                          | IDC_ICDBWND_ITALIC                    | 36545 |
| Aus Online Browser entfernen             | IDC_ICDBWND_REMOVE_FROM_BROWSERVIEW   | 36683 |

Forts...

|                                         |                                               |       |
|-----------------------------------------|-----------------------------------------------|-------|
|                                         | IDC_ICDBWND_TEXT_COLOR                        | 36546 |
|                                         | IDC_ICDBWND_TOOLBAR_COLOR                     | 36547 |
|                                         | IDC_ICDBWND_UNDERLINE                         | 36548 |
| Aktive und verwandte Dateien hinzufügen | IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES | 37217 |
| Aktive Datei hinzufügen                 | IDC_ICPROJECTGUI_ADD_ACTIVE_FILE              | 37216 |
| Externes Verzeichnis hinzufügen...      | IDC_ICPROJECTGUI_ADD_EXT_FOLDER               | 37219 |
| Externes Webverzeichnis hinzufügen...   | IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER           | 37220 |
| Dateien hinzufügen...                   | IDC_ICPROJECTGUI_ADD_FILES                    | 37213 |
| Projektdateien hinzufügen...            | IDC_ICPROJECTGUI_ADD_FOLDER                   | 37218 |
| Globale Ressourcendatei hinzufügen...   | IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE          | 37238 |
| URL hinzufügen...                       | IDC_ICPROJECTGUI_ADD_URL                      | 37214 |
| Übergeordneten Ordner anzeigen          | IDC_ICPROJECTGUI_OPEN_CONTAINING_FOLDER       | 37237 |
| Öffnen                                  | IDC_ICPROJECTGUI_OPEN_IN_EXTERNAL_APP         | 37236 |

|                                       |                                                           |       |
|---------------------------------------|-----------------------------------------------------------|-------|
| Datei in MapForce bearbeiten          | IDC_ICPROJECTGUI_OPEN_IN_MAPFORCE                         | 37234 |
| Datei in XMLSpy bearbeiten            | IDC_ICPROJECTGUI_OPEN_IN_XMLSPY                           | 37233 |
| Eigenschaften...                      | IDC_ICPROJECTGUI_PROPERTIES                               | 37222 |
| Aktualisieren                         | IDC_ICPROJECTGUI_REFRESH_EXT_FOLDER                       | 37221 |
| Als Modul importieren                 | IDC_IMPORT_MODULE_FROM_FILE                               | 37879 |
| In globale Stile importieren          | IDC_IMPORT_STYLESHEET_FROM_FILE                           | 37880 |
| Design-Element einfügen               | IDC_INSERT_DESIGN_ELEMENT                                 | 40192 |
| XPath einfügen...                     | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_AUTO_CALC              | 40164 |
| Beim Einfügen nach Datenquelle fragen | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_AUTO_CREATE_TEMP_LATES | 40193 |
| Kontrollkästchen einfügen...          | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_CHECKBOX               | 40177 |
| Auswahlliste einfügen...              | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_COMBOBOX               | 40176 |
| Inhalt einfügen                       | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_CONTENTS               | 40165 |
| Eingabefeld einfügen                  | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_EDITFIELD              | 40174 |
| Mehrzeiliges Eingabefeld einfügen     | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_EDITFIELD_MULTILINE    | 40175 |
| Bild einfügen...                      | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_IMAGE                  | 40173 |
| Layout-Feld einfügen...               | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_LAYOUT_BOX             | 40157 |
| Layout Container einfügen             | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_LAYOUT_CONTAINER       | 40156 |

Forts...

|                                                |                                                          |       |
|------------------------------------------------|----------------------------------------------------------|-------|
| Numerierung und Aufzählungszeichen einfügen... | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_LIST                  | 40171 |
| Absatz einfügen                                | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_PARAGRAPH             | 40167 |
| Optionsfeld einfügen...                        | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_RADIOBUTTON           | 40178 |
| Linie einfügen                                 | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_SHAPE_LINE            | 40186 |
| Tabelle einfügen...                            | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_TABLE                 | 40169 |
| Vorlage einfügen                               | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_TEMPLATE              | 40166 |
| Benutzerdefinierte Vorlage einfügen            | IDC_INSERT_DESIGN_ELEM_INTERACTIVE_USER_DEFINED_TEMPLATE | 40183 |
|                                                | IDC_ITALIC                                               | 37823 |
| Liste einfügen                                 | IDC_LIST_INSERT_FIX                                      | 36028 |
| Verzweigung nach unten                         | IDC_MOVE_OPTION_DOWN                                     | 37625 |
| Verzweigung nach oben                          | IDC_MOVE_OPTION_UP                                       | 37626 |
| Design öffnen                                  | IDC_OPEN_DESIGN_FROM_FILE                                | 37877 |
| Absatz anwenden/ändern                         | IDC_PARAGRAPH_PLACEHOLDER                                | 37821 |

|                                     |                                      |       |
|-------------------------------------|--------------------------------------|-------|
| Als neue Schemaquelle hinzufügen... | IDC_PROJECT_ADD_AS_XBRL_SCHEMASOURCE | 36133 |
| Als neue Schemaquelle hinzufügen... | IDC_PROJECT_ADD_AS_XSD_SCHEMASOURCE  | 36045 |
| Als Schema-Datei zuweisen           | IDC_PROJECT_ASSIGN_AS_SCHEMA         | 36048 |
| Als XBRL-Vorlagendatei zuweisen     | IDC_PROJECT_ASSIGN_AS_TEMPLATE_XBRL  | 36135 |
| Als XML-Vorlagendatei zuweisen      | IDC_PROJECT_ASSIGN_AS_TEMPLATE_XML   | 36047 |
| Als XBRL-Arbeitsdatei zuweisen      | IDC_PROJECT_ASSIGN_AS_WORKING_XBRL   | 36134 |
| Als XML-Arbeitsdatei zuweisen       | IDC_PROJECT_ASSIGN_AS_WORKING_XML    | 36046 |
| Neues Design erstellen              | IDC_PROJECT_CREATE_NEW_FROM_HTML     | 36044 |
| Neues Design erstellen...           | IDC_PROJECT_CREATE_NEW_FROM_XBRL     | 36132 |
| Neues Design erstellen...           | IDC_PROJECT_CREATE_NEW_FROM_XSD      | 36043 |
| Als Modul importieren               | IDC_PROJECT_IMPORT_AS_MODULE         | 36042 |
| In globale Stile importieren        | IDC_PROJECT_IMPORT_AS_STYLESHEET     | 36049 |
| Bild in Design einfügen             | IDC_PROJECT_INSERT_IMAGE             | 36051 |
| Design öffnen                       | IDC_PROJECT_OPEN_IN_STYLEVISION      | 36041 |
| Nur Tag entfernen                   | IDC_REMOVE                           | 37521 |
| Alle Lesezeichen löschen            | IDC_REMOVEALLBOOKMARKS               | 37629 |

Forts...

|                                                     |                                               |       |
|-----------------------------------------------------|-----------------------------------------------|-------|
| Entfernen                                           | IDC_REMOVE_PAGE_LAYOUT                        | 37630 |
| Globale Vorlage löschen                             | IDC_REMOVE_TEMPLATE                           | 37631 |
| Umbenennen                                          | IDC_RENAME_NODE                               | 37632 |
| Automatisch reduzieren beim Synchronisieren         | IDC_SCHEMASOURCES_AUTOCOLLAPSE                | 37873 |
| Selektion in Attribute konvertieren                 | IDC_SCHEMASOURCES_CONVERT_TO_ATTRIBUTES       | 37644 |
| Ausgewählte Tabelle/Liste in Attribute konvertieren | IDC_SCHEMASOURCES_CONVERT_TO_ATTRIBUTES_TABLE | 37645 |
| Selektion in Elemente konvertieren                  | IDC_SCHEMASOURCES_CONVERT_TO_ELEMENTS         | 37646 |
| Ausgewählte Tabelle/Liste in Elemente konvertieren  | IDC_SCHEMASOURCES_CONVERT_TO_ELEMENTS_TABLE   | 37647 |
| Selektion in Element einschließen                   | IDC_SCHEMASOURCES_SURROUND_TEMPLATE           | 37648 |
| DB-Schema hinzufügen...                             | IDC_SCHEMASOURCE_ADD_FROM_DB                  | 37649 |
| Schema für XML-Spalte in DB-Tabelle hinzufügen...   | IDC_SCHEMASOURCE_ADD_FROM_DBCELL              | 37862 |
| XML-Schema/DTD/XML hinzufügen...                    | IDC_SCHEMASOURCE_ADD_FROM_FILE                | 37650 |

|                                           |                                                |       |
|-------------------------------------------|------------------------------------------------|-------|
| XBRL-Taxonomie hinzufügen...              | IDC_SCHEMASOURCE_ADD_FROM_XBRL_TAXONOMY        | 37917 |
| Benutzerdefiniertes Schema hinzufügen     | IDC_SCHEMASOURCE_ADD_USER                      | 37651 |
| Anzeigen: %s                              | IDC_SHOW_OPTION_FIRST                          | 37652 |
| Sortieren nach...                         | IDC_SORT                                       | 37653 |
| Alle reduzieren                           | IDC_STYLES_COLLAPSE                            | 37654 |
| Alle erweitern                            | IDC_STYLES_EXPANDALL                           | 37655 |
| Alle Verzweigungen anzeigen               | IDC_STYLEVISIONGUI_CONDITION_SHOW_ALL_BRANCHES | 37794 |
| Tabelle                                   | IDC_TABLE                                      | 37734 |
| Tabelle einfügen...                       | IDC_TABLE_INSERT_FIX                           | 37663 |
| Lesezeichen einfügen / löschen            | IDC_TOGGLE_BOOKMARK                            | 37673 |
| Globale Ressourcen                        | IDC_TOOLBAR_ALTOVA_GLOBAL_RESOURCES            | 36029 |
| Standard-Symbolleiste                     | IDC_TOOLBAR_STANDARD                           | 36030 |
| Alle reduzieren                           | IDC_TREE_COLLAPSE                              | 37677 |
| Ab hier reduzieren                        | IDC_TREE_COLLAPSE_KNOT                         | 37678 |
| Alle erweitern                            | IDC_TREE_EXPANDALL                             | 37679 |
| Ab hier erweitern                         | IDC_TREE_EXPAND_KNOT                           | 37680 |
| Alle auf diese Ebene erweitern/reduzieren | IDC_TREE_SAME_LEVEL                            | 37681 |
| XBRL-Vorlagendateizuweisung aufheben      | IDC_UNASSIGN_TEMPLATE_XBRL_FILE                | 37940 |

Forts...

|                                          |                                |       |
|------------------------------------------|--------------------------------|-------|
| Zuweisung der XBRL-Arbeitsdatei aufheben | IDC_UNASSIGN_WORKING_XBRL_FILE | 37941 |
|                                          | IDC_UNDERLINE                  | 37824 |
| Globale Vorlage verwenden                | IDC_USE_GLOBAL_TEMPLATE        | 37685 |
|                                          | IDC_VALIGN_BOTTOM              | 37687 |
|                                          | IDC_VALIGN_CENTER              | 37688 |
|                                          | IDC_VALIGN_TOP                 | 37689 |
| Schema hinzufügen...                     | IDC_XML_ADD_SCHEMA             | 36649 |
| Änderungen übernehmen                    | IDC_XML_COMMIT_CHANGES         | 36653 |
| Dekomposition                            | IDC_XML_DECOMPOSITION          | 36654 |
| Schema löschen                           | IDC_XML_DROP_SCHEMA            | 36650 |
| Drop Flag entfernen                      | IDC_XML_UNDROP_SCHEMA          | 36651 |
| Schema anzeigen                          | IDC_XML_VIEW_SCHEMA            | 36652 |
|                                          | IDC_XSLT_VERSION_1             | 37690 |
|                                          | IDC_XSLT_VERSION_2             | 37691 |

|                                                 |                                             |       |
|-------------------------------------------------|---------------------------------------------|-------|
|                                                 | IDC_ZOOM_PLACEHOLDER                        | 40191 |
| Flag 'Wichtig' setzen/entfernen                 | IDR_CONTEXT_IMPORTANT                       | 37693 |
| Zurücksetzen                                    | IDR_CONTEXT_REMOVE_ATTRIBUTE                | 37694 |
| Alle auflisten                                  | IDR_CONTEXT_VIEWMODE_ALPHA                  | 37695 |
| Nicht leere auflisten                           | IDR_CONTEXT_VIEWMODE_ALPHA_SET              | 37696 |
| Gruppiert                                       | IDR_CONTEXT_VIEWMODE_GROUPED                | 37697 |
| XPath                                           | IDR_CONTEXT_XPATH                           | 37698 |
| Attribut anhängen                               | IDR_SCHEMASOURCES_ADD_ATTRIBUTE             | 37708 |
| Element anhängen                                | IDR_SCHEMASOURCES_ADD_ELEMENT               | 37709 |
| Child-Attribut anhängen                         | IDR_SCHEMASOURCES_CHILD_ATTRIBUTE           | 37711 |
| Child-Element anhängen                          | IDR_SCHEMASOURCES_CHILD_ELEMENT             | 37712 |
| In Attribut/Element konvertieren                | IDR_SCHEMASOURCES_CONVERT_ATTRIBUTE_ELEMENT | 37713 |
| Als global definieren / Globale Vorlage löschen | IDR_SCHEMASOURCES_GLOBAL                    | 37714 |
| Attribut einfügen                               | IDR_SCHEMASOURCES_INSERT_ATTRIBUTE          | 37715 |
| Element einfügen                                | IDR_SCHEMASOURCES_INSERT_ELEMENT            | 37716 |
| Alle Vorlagen dienen als Ebene                  | IDR_SCHEMASOURCES_MARK_ALL_AS_LEVEL         | 37717 |

Forts...

|                                     |                                       |       |
|-------------------------------------|---------------------------------------|-------|
| Objekt löschen                      | IDR_SCHEMASOURCES_REMOVE              | 37718 |
| Umbenennen                          | IDR_SCHEMASOURCES_RENAME              | 37719 |
| Als Hauptschemaquelle definieren    | IDR_SCHEMASOURCES_SET_MAIN            | 37720 |
| Baumstruktur synchronisieren        | IDR_SCHEMASOURCES_SYNC                | 37721 |
| Keine Vorlage dient als Ebene       | IDR_SCHEMASOURCES_UNMARK_ALL_AS_LEVEL | 37722 |
| Hinzufügen                          | IDR_STYLES_ADD                        | 37723 |
| Flag 'Wichtig' setzen/entfernen     | IDR_STYLES_IMPORTANT                  | 37724 |
| Einfügen                            | IDR_STYLES_INSERT                     | 37725 |
| Nach unten                          | IDR_STYLES_MOVE_DOWN                  | 37726 |
| Nach oben                           | IDR_STYLES_MOVE_UP                    | 37727 |
| Alle externen CSS-Dateien neu laden | IDR_STYLES_RELOAD                     | 37728 |
| Entfernen/zurücksetzen              | IDR_STYLES_REMOVE                     | 37729 |
| Zurücksetzen                        | IDR_STYLES_RESET                      | 37730 |

|                                     |                               |       |
|-------------------------------------|-------------------------------|-------|
| Alle auflisten                      | IDR_STYLES_VIEWMODE_ALPHA     | 37731 |
| Nicht leere auflisten               | IDR_STYLES_VIEWMODE_ALPHA_SET | 37732 |
| Gruppiert                           | IDR_STYLES_VIEWMODE_GROUPED   | 37733 |
| Aktive Datei zum Projekt hinzufügen | ID_ADDACTIVEFILETOPROJECT     | 36549 |
| Dateien zu Projekt hinzufügen ...   | ID_ADDFILESTOPROJECT          | 36550 |
|                                     | ID_BUTTON_OPTIONS             | 36551 |
|                                     | ID_BUTTON_SOURCE              | 36552 |
| Verbindung herstellen               | ID_CONNECT                    | 36553 |
| Mit allen Datenquellen verbinden    | ID_CONNECTTOALLDATASOURCES    | 36554 |
| -14:00                              | ID_CONTENT_TIMEZONES_M14_0000 | 18356 |
| -13:00                              | ID_CONTENT_TIMEZONES_M14_0100 | 18357 |
| -12:00                              | ID_CONTENT_TIMEZONES_M14_0200 | 18358 |
| -11:00                              | ID_CONTENT_TIMEZONES_M14_0300 | 18359 |
| -10:00                              | ID_CONTENT_TIMEZONES_M14_0400 | 18360 |
| -09:00                              | ID_CONTENT_TIMEZONES_M14_0500 | 18361 |
| -08:00                              | ID_CONTENT_TIMEZONES_M14_0600 | 18362 |
| -07:00                              | ID_CONTENT_TIMEZONES_M14_0700 | 18363 |

Forts...

|          |                               |       |
|----------|-------------------------------|-------|
| -06:00   | ID_CONTENT_TIMEZONES_M14_0800 | 18364 |
| -05:00   | ID_CONTENT_TIMEZONES_M14_0900 | 18365 |
| -04:00   | ID_CONTENT_TIMEZONES_M14_1000 | 18366 |
| -03:00   | ID_CONTENT_TIMEZONES_M14_1100 | 18367 |
| -02:00   | ID_CONTENT_TIMEZONES_M14_1200 | 18368 |
| -01:00   | ID_CONTENT_TIMEZONES_M14_1300 | 18369 |
| 00:00    | ID_CONTENT_TIMEZONES_M14_1400 | 18370 |
| +01:00   | ID_CONTENT_TIMEZONES_M14_1500 | 18371 |
| +02 : 00 | ID_CONTENT_TIMEZONES_M14_1600 | 18372 |
| +03 : 00 | ID_CONTENT_TIMEZONES_M14_1700 | 18373 |
| +04 : 00 | ID_CONTENT_TIMEZONES_M14_1800 | 18374 |
| +05 : 00 | ID_CONTENT_TIMEZONES_M14_1900 | 18375 |
| +06 : 00 | ID_CONTENT_TIMEZONES_M14_2000 | 18376 |
| +07 : 00 | ID_CONTENT_TIMEZONES_M14_2100 | 18377 |
| +08 : 00 | ID_CONTENT_TIMEZONES_M14_2200 | 18378 |
| +09 : 00 | ID_CONTENT_TIMEZONES_M14_2300 | 18379 |
| +10 : 00 | ID_CONTENT_TIMEZONES_M14_2400 | 18380 |



|                  |                                  |       |
|------------------|----------------------------------|-------|
| +11 : 00         | ID_CONTENT_TIMEZONES_M14_2500    | 18381 |
| +12 : 00         | ID_CONTENT_TIMEZONES_M14_2600    | 18382 |
| +13 : 00         | ID_CONTENT_TIMEZONES_M14_2700    | 18383 |
| +14 : 00         | ID_CONTENT_TIMEZONES_M14_2800    | 18384 |
| #####            | ID_CONTENT_TIMEZONES_MIN00       | 18341 |
| 14:59            | ID_CONTENT_TIMEZONES_MIN15       | 18342 |
| 29:59:00         | ID_CONTENT_TIMEZONES_MIN30       | 18343 |
| 44:59:00         | ID_CONTENT_TIMEZONES_MIN45       | 18344 |
| No TZ            | ID_CONTENT_TIMEZONES_NOTIMEZONE  | 18354 |
| UTC              | ID_CONTENT_TIMEZONES_UTCTIMEZONE | 18355 |
|                  | ID_CONTEXT_HELP                  | 57669 |
| Ordner erstellen | ID_CREATEFOLDER                  | 36555 |
| Inhalt erstellen | ID_CREATE_ALL_CONTENTS           | 37735 |

Forts...

|                                               |                                          |       |
|-----------------------------------------------|------------------------------------------|-------|
| Nummerierung und Aufzählungszeichen erstellen | ID_CREATE_LIST                           | 37736 |
| Absatz erstellen                              | ID_CREATE_PARAGRAPH                      | 37737 |
| Tabelle erstellen...                          | ID_CREATE_TABLE                          | 37738 |
| Vorlagen erstellen                            | ID_CREATE_TEMPLATES                      | 37739 |
| XBRL-Label erstellen                          | ID_CREATE_XBRL_LABEL                     | 37915 |
| XBRL-Label als Text erstellen                 | ID_CREATE_XBRL_LABEL_AS_TEXT             | 37916 |
| XBRL-Tabelle erstellen...                     | ID_CREATE_XBRL_TABLE                     | 37918 |
| XBRL-Vorlage erstellen                        | ID_CREATE_XBRL_TEMPLATE                  | 37914 |
|                                               | ID_DATABASEQUERY_CONNECTION_DATABASES    | 36557 |
|                                               | ID_DATABASEQUERY_CONNECTION_DATASOURCE   | 38008 |
|                                               | ID_DATABASEQUERY_CONNECTION_QUICKCONNECT | 38013 |
| Automatisch reduzieren beim Synchronisieren   | ID_DESIGNTREE_AUTOCOLLAPSE               | 37740 |
| Entfernen                                     | ID_DESIGNTREE_REMOVE                     | 37741 |
| Umbenennen                                    | ID_DESIGNTREE_RENAME                     | 37872 |
| Baumstruktur synchronisieren                  | ID_DESIGNTREE_SYNC                       | 37742 |
| Verbindung trennen                            | ID_DISCONNECT                            | 38014 |
| Verbindung zu allen Datenquellen trennen      | ID_DISCONNECTFROMALLDATASOURCES          | 36560 |
| Daten bearbeiten                              | ID_EDITRESULTDATA                        | 38022 |
| Löschen                                       | ID_EDIT_DROP_DBOBJECT                    | 38019 |
| Löschen                                       | ID_EDIT_DROP_FAV_OBJECT                  | 36562 |

|                              |                               |       |
|------------------------------|-------------------------------|-------|
| Umbenennen                   | ID_EDIT_RENAME                | 38020 |
| Alle SQL-Dateien ausführen   | ID_EXECUTEALLSQLFILES         | 36564 |
| 'v. Chr.' / 'n. Chr.'        | ID_FIELDTYPES_AD              | 37744 |
| 'AM' / 'PM'                  | ID_FIELDTYPES_AM              | 37745 |
| 'n. u. Z.' / 'v. u. Z.'      | ID_FIELDTYPES_CE              | 37746 |
| Jahrhundert ('2004' -> '20') | ID_FIELDTYPES_CENTURY         | 37747 |
| Jahrhundert ('2004' -> '21') | ID_FIELDTYPES_CENTURY1        | 37748 |
| Komma                        | ID_FIELDTYPES_COMMA           | 37749 |
| Tag als Zahl                 | ID_FIELDTYPES_DAYASNUMBER     | 37750 |
| Tag-im-Jahr als Zahl         | ID_FIELDTYPES_DAYINYEASNUMBER | 37751 |

Forts...

|                                    |                                       |       |
|------------------------------------|---------------------------------------|-------|
| Stelle oder vorangestellte Null    | ID_FIELDTYPES_DIGITORLEADINGZERO      | 37752 |
| Stelle oder Leerzeichen            | ID_FIELDTYPES_DIGITORSPEACE           | 37753 |
| Dezimaltrennzeichen                | ID_FIELDTYPES_DIGITSEPARATOR          | 37754 |
| Kommastellen (2 Stellen)           | ID_FIELDTYPES_FRACTION2               | 37755 |
| Kommastellen (2 oder mehr Stellen) | ID_FIELDTYPES_FRACTION2PLUS           | 37756 |
| Stunde als Zahl (1 - 12)           | ID_FIELDTYPES_HOURASNUMBER12          | 37757 |
| Stunde als Zahl                    | ID_FIELDTYPES_HOURASNUMBER24          | 37758 |
| Minuten als Zahl                   | ID_FIELDTYPES_MINUTESASNUMBER         | 37759 |
| Monat als Zahl                     | ID_FIELDTYPES_MONTHASNUMBER           | 37760 |
| Monat als Text, Abk.               | ID_FIELDTYPES_MONTHASTEXTABBREV       | 37761 |
| Monat als Text, langes Format      | ID_FIELDTYPES_MONTHASTEXTLONGFORMAT   | 37762 |
| Zahl (3 Stellen)                   | ID_FIELDTYPES_NUMBER3                 | 37763 |
| Zahl (4 Stellen mit 0en am Anfang) | ID_FIELDTYPES_NUMBER4_0               | 37764 |
| Zahl (6 Stellen mit Trennzeichen)  | ID_FIELDTYPES_NUMBER6                 | 37765 |
| Text in Anführungszeichen          | ID_FIELDTYPES_QUOTEDTEXT              | 37766 |
| Sekunden als Dezimalzahl           | ID_FIELDTYPES_SECONDSASDECIMALNUMBER  | 37767 |
| Sekunden als Zahl                  | ID_FIELDTYPES_SECONDSASNUMBER         | 37768 |
| Zeitzone Stunden                   | ID_FIELDTYPES_TIMEZONEHOURS           | 37769 |
| Zeitzone Minuten                   | ID_FIELDTYPES_TIMEZONEMINUTES         | 37770 |
| Woche als Zahl                     | ID_FIELDTYPES_WEEKASNUMBER            | 37771 |
| Wochentag als Zahl                 | ID_FIELDTYPES_WEEKDAYASNUMBER         | 37772 |
| Wochentag als Text, Abk.           | ID_FIELDTYPES_WEEKDAYASTEXTABBREV     | 37773 |
| Wochentag als Text, langes Format  | ID_FIELDTYPES_WEEKDAYASTEXTLONGFORMAT | 37774 |
| Jahr als Zahl                      | ID_FIELDTYPES_YEARASNUMBER            | 37775 |

|                                                               |                                 |       |
|---------------------------------------------------------------|---------------------------------|-------|
| Jahr als Kurzzahl                                             | ID_FIELDTYPES_YEARASSHORTNUMBER | 37776 |
| Projekt öffnen                                                | ID_FILE_LOAD_PROJECT            | 36565 |
| Neu...                                                        | ID_FILE_NEW                     | 57600 |
|                                                               | ID_FILTER                       | 36566 |
| Basisjahr definieren (Eingabe erforderlich)                   | ID_FORMATOPTIONS_BASE           | 37779 |
| Zeichen für Dezimaltrennzeichen setzen (Eingabe erforderlich) | ID_FORMATOPTIONS_DEC            | 37780 |

Forts...

|                                                                        |                                      |       |
|------------------------------------------------------------------------|--------------------------------------|-------|
| (Leere Option)                                                         | ID_FORMATOPTIONS_EMPTY               | 37781 |
| Zeichen für Füllzeichen setzen (Eingabe erforderlich)                  | ID_FORMATOPTIONS_FILL                | 37782 |
| Stunden von 1-12                                                       | ID_FORMATOPTIONS_H12                 | 37783 |
| Stunden von 0-23                                                       | ID_FORMATOPTIONS_H24                 | 37784 |
| Nur erstes Zeichen                                                     | ID_FORMATOPTIONS_INIT                | 37785 |
| Langes Format                                                          | ID_FORMATOPTIONS_LONG                | 37786 |
| Kleinschreibung                                                        | ID_FORMATOPTIONS_LOWERCASE           | 37787 |
| Als positive Zahl anzeigen                                             | ID_FORMATOPTIONS_POS                 | 37788 |
| Schreibgeschützt                                                       | ID_FORMATOPTIONS_READONLY            | 37789 |
| Zeichen für Tausendertrennzeichen setzen (Eingabe erforderlich)        | ID_FORMATOPTIONS_SEP                 | 37790 |
| Kurzes Format                                                          | ID_FORMATOPTIONS_SHORT               | 37791 |
| GROSSBUCHSTABEN                                                        | ID_FORMATOPTIONS_UPPERCASE           | 37792 |
| Fügt eine Konfiguration hinzu                                          | ID_GLOBALRESOURCES_ADDCONFIG         | 37429 |
| Fügt eine Konfiguration als Kopie der ausgewählten Konfiguration hinzu | ID_GLOBALRESOURCES_ADDCONFIGCOPY     | 37430 |
| Löscht eine Konfiguration                                              | ID_GLOBALRESOURCES_DELCONFIG         | 37431 |
| XML-Schema zuweisen...                                                 | ID_ICDBWND_ASSIGN_XML_SCHEMA         | 36656 |
| Referenzierte Tabelle anzeigen                                         | ID_ICDBWND_FGNKEY_GOTO_REFERENCE     | 36567 |
| In XMLSpy anzeigen                                                     | ID_ICDBWND_VIEW_XMLSCHEMA_IN_XMLSPY  | 36500 |
| Neue Datenquelle hinzufügen...                                         | ID_ICDBWND_ADDANEWDATASOURCE         | 36568 |
| Zum Design Editor hinzufügen                                           | ID_ICDBWND_BROWSER_ADD_TO_DESIGNVIEW | 38021 |
| Zu Favoriten hinzufügen/aus Favoriten löschen                          | ID_ICDBWND_BROWSER_ADD_TO_FAVOURITES | 38018 |
| Aktualisieren                                                          | ID_ICDBWND_BROWSER_REFRESH_ROOT      | 36571 |
|                                                                        | ID_ICDBWND_BROWSER_SEARCH            | 36572 |
|                                                                        | ID_ICDBWND_BROWSER_SEARCH_COMBO      | 36573 |

|                                 |                                             |       |
|---------------------------------|---------------------------------------------|-------|
|                                 | ID_ICDBWND_BROWSER_SEARCH_MODE              | 36574 |
| Alle                            | ID_ICDBWND_BROWSER_SEARCH_MODE_ALL          | 36575 |
| der aktuellen Datenquelle       | ID_ICDBWND_BROWSER_SEARCH_MODE_DATA_SOURCE  | 36576 |
| Aus dem aktiven Objekt          | ID_ICDBWND_BROWSER_SEARCH_MODE_FOCUSED_ITEM | 36577 |
| Im neuen Design Editor anzeigen | ID_ICDBWND_BROWSER_SHOW_IN_DESIGNVIEW       | 38016 |
| Aktivieren                      | ID_ICDBWND_CHECK                            | 36579 |

Forts...

|                               |                                       |       |
|-------------------------------|---------------------------------------|-------|
| Children aktivieren           | ID_ICDBWND_CHECK_ALL                  | 36580 |
| Löschen                       | ID_ICDBWND_CLEAR_ROWCOUNT             | 36538 |
| Children                      | ID_ICDBWND_COLLAPSE_CHILDREN          | 36581 |
| Gleichrangige                 | ID_ICDBWND_COLLAPSE_SIBLING           | 36582 |
| SQL ausführen                 | ID_ICDBWND_EXECUTE                    | 36583 |
| Children                      | ID_ICDBWND_EXPAND_CHILDREN            | 36584 |
| Gleichrangige                 | ID_ICDBWND_EXPAND_SIBLINGS            | 36585 |
| Datenbankdaten exportieren... | ID_ICDBWND_EXPORT                     | 38015 |
|                               | ID_ICDBWND_EXPORT_PREVIEW             | 36587 |
| Hinzufügen                    | ID_ICDBWND_FILEDSN_ADD                | 36588 |
| Löschen                       | ID_ICDBWND_FILEDSN_DELETE             | 36589 |
|                               | ID_ICDBWND_FILTER_CHECKED             | 36590 |
| Enthält                       | ID_ICDBWND_FILTER_CONTAINS            | 36591 |
| Enthält nicht                 | ID_ICDBWND_FILTER_DOES_NOT_CONTAIN    | 36592 |
| Endet mit                     | ID_ICDBWND_FILTER_ENDS_WITH           | 36593 |
| Ist gleich                    | ID_ICDBWND_FILTER_EQUALS              | 36594 |
|                               | ID_ICDBWND_FILTER_FAVORITES           | 36595 |
| Kein Filter                   | ID_ICDBWND_FILTER_INACTIVE            | 36596 |
| Beginnt mit                   | ID_ICDBWND_FILTER_STARTS_WITH         | 36597 |
|                               | ID_ICDBWND_FLAG_BLUE                  | 36598 |
|                               | ID_ICDBWND_FLAG_GREEN                 | 36599 |
|                               | ID_ICDBWND_FLAG_MARK                  | 36600 |
|                               | ID_ICDBWND_FLAG_ORANGE                | 36601 |
|                               | ID_ICDBWND_FLAG_PURPLE                | 36602 |
|                               | ID_ICDBWND_FLAG_RED                   | 36603 |
|                               | ID_ICDBWND_FLAG_YELLOW                | 36604 |
| Verbindung herstellen         | ID_ICDBWND_MENU_DATASOURCE_CONNECT    | 36605 |
| Verbindung trennen            | ID_ICDBWND_MENU_DATASOURCE_DISCONNECT | 36606 |

|                  |                                      |       |
|------------------|--------------------------------------|-------|
| Tabellen abrufen | ID_ICDBWND_MENU_DATASOURCE_GETTABLES | 36607 |
| Vorschau         | ID_ICDBWND_PREVIEWITEM               | 36608 |

Forts...

|                             |                                 |       |
|-----------------------------|---------------------------------|-------|
| Alle Favoriten entfernen    | ID_ICDBWND_REMOVE_ALL_FAVORITES | 36609 |
| Ein-/Ausblenden             | ID_ICDBWND_TOGGLE               | 36610 |
| Deaktivieren                | ID_ICDBWND_UNCHECK              | 36611 |
| Children deaktivieren       | ID_ICDBWND_UNCHECK_ALL          | 36612 |
| Anzeigen/Aktualisieren      | ID_ICDBWND_UPDATE_ROWCOUNT      | 36537 |
|                             | ID_LAYOUTS                      | 36613 |
| Tabellenabhängigkeiten      | ID_LAYOUT_DEPENDENCIES          | 36614 |
| Flache Hierarchie           | ID_LAYOUT_FLAT                  | 36615 |
| Ordner                      | ID_LAYOUT_FOLDERS               | 36616 |
| Keine Schemas               | ID_LAYOUT_FOLDERS_NOSCHEMAS     | 36617 |
| Keine Ordner                | ID_LAYOUT_NOFOLDERS             | 36618 |
|                             | ID_NEXT_PANE                    | 57680 |
| Öffnen                      | ID_OPEN                         | 36619 |
| XML-Schemas verwalten...    | ID_OPEN_MANAGE_XMLSCHEMA_DIALOG | 36658 |
|                             | ID_PREV_PANE                    | 57681 |
| Neues Projekt               | ID_PROJECT_NEWPROJECT           | 36620 |
| Alle Datenquellen entfernen | ID_REMOVEALLDATASOURCES         | 36621 |
| Entfernen                   | ID_REMOVE_DATASOURCE            | 36622 |
| Aus Favoriten löschen       | ID_REMOVE_FAVORITE_ITEM         | 36623 |
| Entfernen                   | ID_REMOVE_FROM_PROJECT          | 36624 |
|                             | ID_RESULTGRID_FINDNEXT          | 36671 |
|                             | ID_RESULTGRID_FINDPREV          | 36672 |
| Alle Zeilen                 | ID_RETRIEVE_ALLROWS             | 36625 |
| Erste n Zeilen              | ID_RETRIEVE_FIRSTROWS           | 36626 |
| Projekt speichern           | ID_SAVEPROJECT                  | 36627 |
| Projekt speichern unter ... | ID_SAVEPROJECTAS                | 36628 |
| Hinzufügen                  | ID_SHOW_SQL_ADD                 | 36629 |
| Ändern                      | ID_SHOW_SQL_ALTER               | 36630 |
| Erstellen                   | ID_SHOW_SQL_CREATE              | 36631 |
| Daten löschen               | ID_SHOW_SQL_DELETE              | 36632 |

Forts...

|                                             |                                          |       |
|---------------------------------------------|------------------------------------------|-------|
| Löschen                                     | ID_SHOW_SQL_DROP                         | 36633 |
| Ausführen                                   | ID_SHOW_SQL_EXECUTE                      | 36673 |
| Einfügen                                    | ID_SHOW_SQL_INSERT                       | 36634 |
| Name                                        | ID_SHOW_SQL_NAME                         | 36635 |
| Pfad                                        | ID_SHOW_SQL_PATH                         | 36636 |
| Umbenennen                                  | ID_SHOW_SQL_RENAME                       | 36637 |
| Auswählen                                   | ID_SHOW_SQL_SELECT                       | 36638 |
| Aktualisieren                               | ID_SHOW_SQL_UPDATE                       | 36639 |
| Nach Benutzer- und Systemtabellen sortieren | ID_SORT_BY_TABLE_TYPE                    | 36640 |
| Dummy entry                                 | ID_STYLES_ADD_SELECTOR_CLASS_RANGE_FIRST | 19661 |
| Dummy entry                                 | ID_STYLES_ADD_SELECTOR_HTML_RANGE_FIRST  | 21031 |
| Dummy entry                                 | ID_STYLES_ADD_SELECTOR_ID_RANGE_FIRST    | 19681 |
|                                             | ID_STYLEVISION                           | 36039 |
| Office 2000                                 | ID_VIEW_APPLOOK_2000                     | 37797 |
| Office 2003                                 | ID_VIEW_APPLOOK_2003                     | 37798 |
| Visual Studio.NET 2005                      | ID_VIEW_APPLOOK_VS2005                   | 37799 |
| Windows XP                                  | ID_VIEW_APPLOOK_WIN_XP                   | 37800 |
| Office XP                                   | ID_VIEW_APPLOOK_XP                       | 37801 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR1                    | 37808 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR10                   | 37817 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR2                    | 37809 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR3                    | 37810 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR4                    | 37811 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR5                    | 37812 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR6                    | 37813 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR7                    | 37814 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR8                    | 37815 |
| <User Toolbar>                              | ID_VIEW_USER_TOOLBAR9                    | 37816 |

## Aufrufen von StyleVisionAPI

Diese Dokumentation beschäftigt sich hauptsächlich mit den ActiveX Controls und Schnittstellen, die für die Integration der StyleVision Benutzeroberfläche in Ihre Applikation benötigt werden. Über die folgenden Eigenschaften haben Sie Zugriff auf das StyleVision Automation Interface (StyleVisionAPI), um die Funktionalitäten der integrierten Komponenten zu automatisieren oder zu definieren.

[StyleVisionControl.Application](#)

[StyleVisionControlDocument.Document](#)

[StyleVisionControlPlaceHolder.Project](#)

Für die Verwendung des StyleVision Automation Interface gelten beim Integrieren von StyleVisionControl auf Dokumentebene einige Einschränkungen. Nähere Informationen dazu finden Sie unter [Integration auf Dokumentebene](#).

## Objektreferenz

### Objekte:

[StyleVisionCommand](#)

[StyleVisionCommands](#)

[StyleVisionControl](#)

[StyleVisionControlDocument](#)

[StyleVisionControlPlaceHolder](#)

Um Zugriff auf die StyleVision Standardfunktionalitäten zu erhalten, können auch Objekte des **StyleVision Automation Interface** aufgerufen werden. Nähere Informationen dazu finden Sie unter [StyleVisionControl.Application](#), [StyleVisionControlDocument.Document](#) und [StyleVisionControlPlaceHolder.Project](#).



## StyleVisionCommand

### Eigenschaften:

[ID](#)  
[Label](#)  
[IsSeparator](#)  
[ToolTip](#)  
[StatusText](#)  
[Accelerator](#)  
[SubCommands](#)

### Beschreibung:

Jedes `MapForceCommand` Objekt kann einer von drei möglichen Typen sein:

- **Befehl:** `ID` ist auf einen Wert größer als 0 gesetzt und `Label` ist auf "Command Name" gesetzt. `IsSeparator` ist `false` und die `SubCommands` Collection ist leer.
- **Trennzeichen:** `IsSeparator` ist `true`. `ID` ist 0 und `Label` ist nicht definiert. Die `SubCommands` Collection ist leer.
- **(Unter)-Menü:** Die `SubCommands` Collection enthält [Command](#) Objekte und `Label` ist der Name des Menüs. `ID` ist auf 0 gesetzt und `IsSeparator` ist `false`.

### Accelerator

**Eigenschaft:** `Label` als `string`

### Beschreibung:

Dies ist bei Befehlsobjekten, die Children der `ALL_COMMANDS` Collection sind, die für den Befehl definierte Zugriffstaste. Wenn dem Befehl keine Zugriffstaste zugewiesen wurde, gibt diese Eigenschaft den leeren String zurück.

Die String-Darstellung der Zugriffstaste hat das folgenden Format:

[ ALT+ ][ CTRL+ ][ SHIFT+ ] key

key wird mittels der Windows Plattform SDK-Funktion `GetKeyNameText` **konvertiert**.

### ID

**Eigenschaft:** `ID` als `long`

### Beschreibung:

`ID` ist 0 für Trennzeichen und Menüs.

Bei Befehlen ist dies die ID, die mit [Exec](#) und [QueryStatus](#) verwendet werden kann.

### IsSeparator

**Eigenschaft:** `IsSeparator` als `boolean`

### Beschreibung:

True, wenn der Befehl ein Trennzeichen ist.

### Label

### Beschreibung:

`Label` ist bei Trennzeichen leer.

Bei Befehlsobjekten, die Children der `ALL_COMMANDS` Collection sind, ist dies ein eindeutiger Name. Die Symbole für die Befehlsschaltflächen sind in Dateien mit diesem Namen gespeichert. Nähere Informationen dazu siehe [Abfragen von Befehlen](#).

Bei Befehlsobjekten, die Children von Menüs sind, enthält die Eigenschaft "Label" den Menütext des Befehls. Bei Untermenüs enthält diese Eigenschaft den Menütext.

StatusText

**Eigenschaft:** Label als [string](#)

**Beschreibung:**

Bei Befehlsobjekten, die Children der ALL\_COMMANDS Collection sind, ist dies der Text, der in der Statusleiste angezeigt wird, wenn der Befehl ausgewählt ist.

SubCommands

**Eigenschaft:** SubCommands als [Commands](#)

**Beschreibung:**

Die SubCommands Collection enthält alle untergeordneten Befehle, wenn dieser Befehl tatsächlich ein Menü oder Untermenü ist.

ToolTip

**Eigenschaft:** ToolTip als [string](#)

**Beschreibung:**

Bei Befehlsobjekten, die Children der ALL\_COMMANDS Collection sind, ist dies der Text, der als Tooltipp angezeigt wird.

## StyleVisionCommands

### Eigenschaften:

[Count](#)

[Item](#)

### Beschreibung:

Sammlung von [Command](#)-Objekten für den Zugriff auf Befehlsbezeichnungen und IDs des StyleVisionControl. Diese Befehle können mit der [Exec](#) Methode ausgeführt werden und ihr Status kann mittels [QueryStatus](#) abgefragt werden.

Count

**Eigenschaft:** Count als [long](#)

### Beschreibung:

Anzahl der [Command](#) Objekte auf dieser Ebene der Collection

Item

**Eigenschaft:** Item (n als [long](#)) als [Command](#)

### Beschreibung:

Ruft den Befehl mit den Index n in dieser Sammlung auf. Der Index basiert auf 1.

## StyleVisionControl

### Eigenschaften:

[IntegrationLevel](#)

[Appearance](#)

[Application](#)

[BorderStyle](#)

[CommandsList](#)

CommandsStructure ( deprecated)

[EnableUserPrompts](#)

[MainMenu](#)

[Toolbars](#)

### Methoden:

[Open](#)

[Exec](#)

[QueryStatus](#)

### Events:

[OnUpdateCmdUI](#)

[OnOpenedOrFocused](#)

[OnCloseEditingWindow](#)

[OnFileChangedAlert](#)

[OnContextChanged](#)

[OnDocumentOpened](#)

[OnValidationWindowUpdated](#)

Dieses Objekt ist ein vollständiges ActiveX Control und sollte nur sichtbar sein, wenn die StyleVision Bibliothek im Applikationsebenenmodus verwendet wird.

### Eigenschaften

Es sind die folgenden Eigenschaften definiert:

[IntegrationLevel](#)

[EnableUserPrompts](#)

[Appearance](#)

[BorderStyle](#)

### Befehlsbezogene Eigenschaften:

[CommandsList](#)

[MainMenu](#)

[Toolbars](#)

CommandsStructure ( nicht mehr verwendet)

### Zugriff auf die StyleVisionAPI:

[Application](#)

### Appearance

**Eigenschaft:** Appearance als [short](#)

**Dispatch Id:** -520

### Beschreibung:

Bei einem Wert, der nicht gleich 0 ist, wird ein Client-Rand rund um das Control angezeigt. Der Standardwert ist 0.

Application

**Eigenschaft:** Application als [Application](#)

**Dispatch Id:** 1

**Beschreibung:**

Mit der Eigenschaft `Application` erhalten Sie Zugriff auf das `Application` Objekt der vollständigen StyleVision Automation Server API. Die Eigenschaft ist schreibgeschützt.

BorderStyle

**Eigenschaft:** BorderStyle als [short](#)

**Dispatch Id:** -504

**Beschreibung:**

Bei einem Wert 1 wird das Control mit einer dünnen Umrandung angezeigt. Der Standardwert ist 0.

CommandsList

**Eigenschaft:** CommandList als [Commands](#) (schreibgeschützt)

**Dispatch Id:** 1004

**Beschreibung:**

Diese Eigenschaft gibt eine flache Liste aller Befehle zurück, die mit StyleVisionControl verfügbar sind.

EnableUserPrompts

**Eigenschaft:** EnableUserPrompts als [boolean](#)

**Dispatch Id:** 1006

**Beschreibung:**

Wenn Sie diese Eigenschaft auf *false* setzen, wird die Eingabeaufforderung im Control deaktiviert. Der Standardwert ist *true*.

IntegrationLevel

**Eigenschaft:** IntegrationLevel als [ICActiveXIntegrationLevel](#)

**Dispatch Id:** 1000

**Beschreibung:**

Die Eigenschaft `IntegrationLevel` bestimmt den Operationsmodus des Control. Nähere Informationen dazu siehe auch [Integration auf Applikationsebene](#) und [Integration auf Dokumentenebene](#).

**Anmerkung:** Diese Eigenschaft muss unbedingt sofort nach Erstellung des `StyleVisionControl` Objekts definiert werden.

## MainMenu

**Eigenschaft:** MainMenu als [Command](#) (schreibgeschützt)

**Dispatch Id:** 1003

### Beschreibung:

Mit dieser Eigenschaft erhalten Sie Zugriff auf die Beschreibung des StyleVisionControl Hauptmenüs.

## Toolbars

**Eigenschaft:** Toolbars als [Commands](#) (schreibgeschützt)

**Dispatch Id:** 1005

### Beschreibung:

Diese Eigenschaft gibt eine Liste aller Symbolleistenbeschreibungen zurück, die alle mit StyleVisionControl verfügbaren Symbolleisten beschreiben.

## Methoden

Es sind die folgenden Methoden definiert:

[Open](#)

[Exec](#)

[QueryStatus](#)

## Exec

**Methode:** Exec (nCmdID als [long](#)) als [boolean](#)

**Dispatch Id:** 6

### Beschreibung:

Exec ruft den StyleVision Befehl mit der ID nCmdID auf. Wenn der Befehl ausgeführt werden kann, gibt die Methode true zurück. Eine Liste aller verfügbaren Befehle finden Sie unter [CommandsStructure](#) und eine Beschreibung, wie man den Status eines Befehls abruft, finden Sie unter [QueryStatus](#).

## Open

**Methode:** Open (strFilePath als [string](#)) als [boolean](#)

**Dispatch Id:** 5

### Beschreibung:

Das Ergebnis der Methode ist von der Erweiterung abhängig, die im Argument strFilePath übergeben wird. Ist die Dateierweiterung .sps, wird ein neues Dokument geöffnet. Ist die Dateierweiterung .svp, wird das entsprechende Projekt geöffnet. Wird eine andere Dateierweiterung in die Methode übergeben, versucht das Control, die Datei als neue Komponente in das aktive Dokument zu laden.

Verwenden Sie diese Methode nicht, um Dokumente oder Projekte bei Verwendung des Control auf Dokumentenebene zu laden. Verwenden Sie statt dessen

[StyleVisionControlDocument.Open](#) und

[StyleVisionControlPlaceholder.OpenProject](#).

**QueryStatus**

**Methode:** `QueryStatus` (`nCmdID` als `long`) als `long`

**Dispatch Id:** 7

**Beschreibung:**

`QueryStatus` gibt den Status "enabled/disabled" und "checked/unchecked" des von `nCmdID` definierten Befehls zurück. Der Status wird als Bitmaske zurückgegeben.

| Bit | Wert | Name      | Bedeutung                                                       |
|-----|------|-----------|-----------------------------------------------------------------|
| 0   | 1    | Supported | Setzen, wenn der Befehl unterstützt wird.                       |
| 1   | 2    | Enabled   | Setzen, wenn der Befehl aktiviert ist (ausgeführt werden kann). |
| 2   | 4    | Checked   | Setzen, wenn der Befehl angehakt ist.                           |

Das bedeutet, dass die Befehls-ID bei Rückgabe von 0 durch `QueryStatus` nicht als gültiger `StyleVision` Befehl erkannt wird. Wenn `QueryStatus` einen Wert 1 oder 5 zurückgibt, wird der Befehl deaktiviert.

**Events**

Das `StyleVisionControl` ActiveX Control stellt die folgenden Verbindungspunkt-Events bereit:

[OnUpdateCmdUI](#)

[OnOpenedOrFocused](#)

[OnCloseEditingWindow](#)

[OnFileChangedAlert](#)

[OnContextChanged](#)

[OnDocumentOpened](#)

[OnValidationWindowUpdated](#)

**OnCloseEditingWindow**

**Event:** `OnCloseEditingWindow` (`i_strFilePath` als `String`) als `boolean`

**Dispatch Id:** 1002

**Beschreibung:**

Dieses Event wird ausgelöst, wenn `StyleVision` ein bereits geöffnetes Dokument schließen muss. Als Antwort auf dieses Event müssen Clients das mit `i_strFilePath` verknüpfte Bearbeitungsfenster schließen. Bei Rückgabe von `true` von diesem Event, wird angezeigt, dass der Client das Dokument geschlossen hat. Clients können `false` zurückgeben, wenn keine bestimmte Behandlung erforderlich ist und `StyleVisionControl` versuchen soll, das Bearbeitungsfenster zu schließen und das damit verknüpfte Dokument-Control zu zerstören.

**OnContextChanged**

**Event:** `OnContextChanged` (`i_strContextName` als `String`, `i_bActive` als `bool`) als `bool`

**Dispatch Id:** 1004

**Beschreibung:**

**Dieses Event wird in `StyleVision` nicht verwendet**

**OnDocumentOpened**

**Event:** `OnDocumentOpened` (`objDocument` als `Document`)

**Dispatch Id:** 1

**Beschreibung:**

Dieses Event wird immer, wenn ein Dokument geöffnet wird, ausgelöst. Das Argument `objDocument` ist ein `Document` Objekt aus dem `StyleVision Automation Interface` und kann dazu verwendet werden, weitere Details zum Dokument abzurufen oder weitere Operationen durchzuführen. Bei Integration auf Dokumentebene ist es oft besser, stattdessen das Event [`StyleVisionControlDocument.OnDocumentOpened`](#) zu verwenden.

`OnFileChangedAlert`

**Event:** `OnFileChangedAlert (i_strFilePath als String) als bool`

**Dispatch Id:** 1001

**Beschreibung:**

Dieses Event wird ausgelöst, wenn eine mit `StyleVisionControl` geladene Datei auf der Festplatte von einer anderen Applikation geändert wurde. Clients sollten `true` zurückgeben, wenn Sie das Event behandelt haben oder `false`, wenn `StyleVision` es auf die übliche Art behandeln soll, also fragen soll, ob das Dokument neu geladen werden soll.

`OnLicenseProblem`

**Event:** `OnLicenseProblem (i_strLicenseProblemText als String)`

**Dispatch Id:** 1005

**Beschreibung:**

Dieses Event wird ausgelöst, wenn `StyleVisionControl` feststellt, dass für dieses Control keine gültige Lizenz vorhanden ist. Wenn die Lizenz auf eine bestimmte Anzahl an Benutzern beschränkt ist, kann dies auch einige Zeit nach Initialisierung des Control geschehen. Dieses Event sollte dazu verwendet werden, um den Zugriff auf die Funktionalität dieses Control zu deaktivieren. Nach diesem Event blockiert dieses Control den Zugriff auf seine Funktionalitäten (und zeigt z.B. leere Fenster in seinen Controls an und gibt bei Anfragen Fehlermeldungen zurück).

`OnOpenedOrFocused`

**Event:** `OnOpenedOrFocused (i_strFilePath als String, i_bOpenWithThisControl als bool)`

**Dispatch Id:** 1000

**Beschreibung:**

Bei Integration auf Applikationsebene informiert dieses Event Clients, dass ein Dokument von `StyleVision` geöffnet oder aktiv gemacht wurde.

Bei Integration auf Dokumentebene gibt dieses Event dem Client die Anweisung, die Datei `i_strFilePath` in einem Dokumentfenster zu öffnen. Wenn die Datei bereits offen ist, sollte das entsprechende Dokumentfenster zum aktiven Fenster gemacht werden.

Wenn `i_bOpenWithThisControl` `true` ist, muss das Dokument mit `StyleVisionControl` geöffnet werden, da ein interner Zugriff erforderlich ist. Andernfalls kann die Datei mit anderen Editoren geöffnet werden.

`OnToolWindowUpdated`

**Event:** `OnToolWindowUpdated( pToolWnd als long )`

**Dispatch Id:** 1006



**Beschreibung:**

Dieses Event wird ausgelöst, wenn das Fenster "Extras" aktualisiert wird.

OnUpdateCmdUI

**Event:** OnUpdateCmdUI ( )

**Dispatch Id:** 1003

**Beschreibung:**

Wird häufig aufgerufen, um dem Programmierer die Möglichkeit zu geben, den Status von StyleVision Befehls mittels [StyleVisionControl.QueryStatus](#) zu überprüfen. Führen Sie an diesem Callback keine langen Operationen durch.

OnValidationWindowUpdated

**Event:** OnValidationWindowUpdated ( )

**Dispatch Id:** 3

**Beschreibung:**

Dieses Event wird ausgelöst, wenn das Fenster "Validierungsausgabe" mit neuen Informationen aktualisiert wird.

## StyleVisionControlDocument

### Eigenschaften:

[Appearance](#)  
[BorderStyle](#)  
[Document](#)  
[IsModified](#)  
[Path](#)  
[ReadOnly](#)

### Methoden:

[Exec](#)  
[New](#)  
[Open](#)  
[QueryStatus](#)  
[Reload](#)  
[Save](#)  
[SaveAs](#)

### Events:

[OnDocumentOpened](#)  
[OnDocumentClosed](#)  
[OnModifiedFlagChanged](#)  
[OnContextChanged](#)  
[OnFileChangedAlert](#)  
[OnActivate](#)

Wenn das StyleVisionControl im Modus auf Dokumentebene integriert ist, wird jedes Dokument in einem eigenen Objekt vom Typ `StyleVisionControlDocument` angezeigt. Das `StyleVisionControlDocument` enthält immer nur ein Dokument, kann aber wiederverwendet werden, um hintereinander mehrere Dateien anzuzeigen.

Dieses Objekt ist ein komplettes ActiveX Control.

Eigenschaften

Es sind die folgenden Eigenschaften definiert:

[ReadOnly](#)  
[IsModified](#)  
[Path](#)  
[Appearance](#)  
[BorderStyle](#)

Zugriff auf die StyleVisionAPI:

[Document](#)

Appearance

**Eigenschaft:** Appearance als [short](#)

**Dispatch Id:** -520

### Beschreibung:

Bei einem Wert, der nicht gleich 0 ist, wird ein Client-Rand rund um das Control angezeigt. Der Standardwert ist 0.

BorderStyle

**Eigenschaft:** BorderStyle als [short](#)

**Dispatch Id:** -504

**Beschreibung:**

Bei einem Wert 1 wird das Control mit einer dünnen Umrandung angezeigt. Der Standardwert ist 0.

Document

**Eigenschaft:** Document als Document

**Dispatch Id:** 3

**Beschreibung:**

Mit der Eigenschaft Document erhalten Sie Zugriff auf das Document Objekt der StyleVision Automation Server API. Diese Schnittstelle bietet zusätzliche Funktionalitäten, die mit dem im Control geladenen Dokument verwendet werden können. Die Eigenschaft ist schreibgeschützt.

IsModified

**Eigenschaft:** IsModified als [boolean](#) (schreibgeschützt)

**Dispatch Id:** 1006

**Beschreibung:**

IsModified ist *true*, wenn der Dokumentinhalt seit dem letzten Öffnen, Neuladen oder Speichern geändert wurde. Andernfalls ist es *false*.

Path

**Eigenschaft:** Path als [String](#)

**Dispatch Id:** 1005

**Beschreibung:**

Definiert den vollständigen Pfadnamen des im Control geladenen Dokuments bzw. ruft diesen ab.

ReadOnly

**Eigenschaft:** ReadOnly als [boolean](#)

**Dispatch Id:** 1007

**Beschreibung:**

Mit Hilfe dieser Eigenschaft können Sie den schreibgeschützten Status des Dokuments aktivieren und deaktivieren. Wenn ReadOnly *true* ist, können keine Änderungen vorgenommen werden.

Methoden

Es sind die folgenden Methoden definiert:

Behandlung von Dokumenten:

[New](#)

[Open](#)

[Reload](#)  
[Save](#)  
[SaveAs](#)

Behandlung von Befehlen:

[Exec](#)  
[QueryStatus](#)

Exec

**Methode:** `Exec (nCmdID als long) als boolean`

**Dispatch Id:** 8

**Beschreibung:**

`Exec` ruft den StyleVision Befehl mit der ID `nCmdID` auf. Wenn der Befehl ausgeführt werden kann, gibt die Methode `true` zurück. Der Client sollte die `Exec` Methode des Dokuments aufrufen, wenn in der Applikation gerade ein aktives Dokument verfügbar ist.

Eine Liste aller verfügbaren Befehle finden Sie unter `CommandsStructure` und eine Beschreibung, wie man den Status eines Befehls abrufen, finden Sie unter [QueryStatus](#).

New

**Methode:** `New () als boolean`

**Dispatch Id:** 1000

**Beschreibung:**

Diese Methode initialisiert ein neues Mapping innerhalb des Control.

Open

**Methode:** `Open (strFilePath als string) als boolean`

**Dispatch Id:** 1001

**Beschreibung:**

`Open` lädt die Datei `strFileName` als das neue Dokument in das Control.

QueryStatus

**Methode:** `QueryStatus (nCmdID als long) als long`

**Dispatch Id:** 9

**Beschreibung:**

`QueryStatus` gibt den Status "enabled/disabled" und "checked/unchecked" des von `nCmdID` definierten Befehls zurück. Der Status wird als Bitmaske zurückgegeben.

| Bit | Wert | Name      | Bedeutung                                                       |
|-----|------|-----------|-----------------------------------------------------------------|
| 0   | 1    | Supported | Setzen, wenn der Befehl unterstützt wird.                       |
| 1   | 2    | Enabled   | Setzen, wenn der Befehl aktiviert ist (ausgeführt werden kann). |
| 2   | 4    | Checked   | Setzen, wenn der Befehl angehakt ist.                           |

Das bedeutet, dass die Befehls-ID bei Rückgabe von 0 durch `QueryStatus` nicht als gültiger StyleVision Befehl erkannt wird. Wenn `QueryStatus` einen Wert 1 oder 5 zurückgibt, ist der

Befehl deaktiviert. Der Client sollte die `QueryStatus` Methode des Dokument-Control aufrufen, wenn in der Applikation gerade ein aktives Dokument vorhanden ist.

Reload

**Methode:** `Reload ()` als `boolean`

**Dispatch Id:** 1002

**Beschreibung:**

`Reload` aktualisiert den Dokumentinhalt aus dem Dateisystem.

Save

**Methode:** `Save ()` als `boolean`

**Dispatch Id:** 1003

**Beschreibung:**

`Save` speichert das aktuelle Dokument unter dem Pfad [Path](#).

SaveAs

**Methode:** `SaveAs (strFileName als string)` als `boolean`

**Dispatch Id:** 1004

**Beschreibung:**

`SaveAs` setzt [Path](#) auf `strFileName` und speichert das Dokument anschließend unter diesem Pfad.

Events

Das `StyleVisionControlDocument` ActiveX Control stellt die folgenden Verbindungspunkt-Events zur Verfügung:

[OnDocumentOpened](#)  
[OnDocumentClosed](#)  
[OnModifiedFlagChanged](#)  
[OnContextChanged](#)  
[OnFileChangedAlert](#)  
[OnActivate](#)  
[OnSetEditorTitle](#)

OnActivate

**Event:** `OnActivate ()`

**Dispatch Id:** 1005

**Beschreibung:**

Dieses Event wird ausgelöst, wenn das Dokument-Control aktiviert ist, den Fokus hat und bereit für die Benutzereingabe ist.

OnContextChanged

**Event:** `OnContextChanged (i_strContextName als String, i_bActive als bool)` als `bool`

**Dispatch Id:** 1004

**Beschreibung:** Keine

OnDocumentClosed

**Event:** OnClosed ()

**Dispatch Id:** 1001

**Beschreibung:**

Dieses Event wird immer dann ausgelöst, wenn das in dieses Control geladene Event geschlossen wird. Das Argument `objDocument` ist ein `Document` Objekt aus dem StyleVision Automation Interface und sollte mit Vorsicht verwendet werden.

OnDocumentOpened

**Event:** OnDocumentOpened (`objDocument` als `Document`)

**Dispatch Id:** 1000

**Beschreibung:**

Dieses Event wird immer dann ausgelöst, wenn ein Dokument in diesem Control geöffnet wird. Das Argument `objDocument` ist ein `Document` Objekt aus dem StyleVision Automation Interface und dient dazu, nähere Informationen über das Dokument abzurufen oder weitere Operationen auszuführen.

OnDocumentSaveAs

**Event:** OnContextDocumentSaveAs (`i_strFileName` als `String`)

**Dispatch Id:** 1007

**Beschreibung:**

Dieses Event wird ausgelöst, wenn dieses Dokument intern unter einem neuen Namen gespeichert wird.

OnFileChangedAlert

**Event:** OnFileChangedAlert () als `bool`

**Dispatch Id:** 1003

**Beschreibung:**

Dieses Event wird ausgelöst, wenn eine in dieses Control geladene Datei auf der Festplatte von einer anderen Applikation geändert wurde. Clients sollten `true` zurückgeben, wenn Sie das Event behandelt haben oder `false`, wenn StyleVision es auf die übliche Art behandeln soll, also fragen soll, ob das Dokument neu geladen werden soll.

OnModifiedFlagChanged

**Event:** OnModifiedFlagChanged (`i_bIsModified` als `boolean`)

**Dispatch Id:** 1002

**Beschreibung:**

Dieses Event wird immer dann ausgelöst, wenn das Dokument zwischen dem Status "geändert" und "nicht geändert" wechselt. Der Parameter `i_bIsModified` ist `true`, wenn sich der Inhalt des Dokuments vom ursprünglichen Inhalt unterscheidet und ist andernfalls `false`.

OnSetEditorTitle

**Event:** OnSetEditorTitle ()

**Dispatch Id:** 1006

**Beschreibung:**

Dieses Event wird ausgelöst, wenn das enthaltene Dokument intern umbenannt wird.

## StyleVisionControlPlaceHolder

### Eigenschaften für alle Arten von Platzhalterfenstern:

[PlaceholderWindowID](#)

### Eigenschaften für das Projekt-Platzhalterfenster:

[Project](#)

### Methoden für das Projekt-Platzhalterfenster:

[OpenProject](#)

[CloseProject](#)

Das `StyleVisionControlPlaceHolder` Control dient dazu, die zusätzlichen `StyleVision` Fenster, wie Übersicht, Bibliothek oder das Projektfenster anzuzeigen. Es wird wie jedes andere ActiveX Control verwendet und kann überall in die Client-Applikation platziert werden.

## Eigenschaften

Es sind die folgenden Eigenschaften definiert:

[PlaceholderWindowID](#)

Zugriff auf die `StyleVisionAPI`:

[Project](#)

## Label

**Eigenschaft:** `Label` als `String` (schreibgeschützt)

**Dispatch Id:** 1001

### Beschreibung:

Mit dieser Eigenschaft erhalten Sie Zugriff auf den Titel des Platzhalters. Diese Eigenschaft ist schreibgeschützt.

## PlaceholderWindowID

**Eigenschaft:** `PlaceholderWindowID` als

**Dispatch Id:** 1

### Beschreibung:

Mit Hilfe dieser Eigenschaft weiß das Objekt, welches `StyleVision` Fenster im Client-Bereich des Control angezeigt werden soll. Die `PlaceholderWindowID` kann jederzeit auf jeden gültigen Wert der gesetzt werden. Das Control ändert seinen Status sofort und zeigt das neue `StyleVision` Fenster an.

## Project

**Eigenschaft:** `Project` als `Project` (schreibgeschützt)

**Dispatch Id:** 2

### Beschreibung:

Mit der Eigenschaft `Project` erhalten Sie Zugriff auf das `Project` Objekt der `StyleVision` Automation Server API. Diese Schnittstelle bietet zusätzliche Funktionalitäten, die mit dem in



das Control geladenen Projekt verwendet werden können. Die Eigenschaft gibt nur dann eine gültige Projektschnittstelle zurück, wenn das Platzhalterfenster [PlaceholderWindowID](#) mit einem Wert von `StyleVisionXProjectWindow (=3)` hat. Die Eigenschaft ist schreibgeschützt.

#### Methoden

Es sind die folgenden Methoden definiert:

[OpenProject](#)  
[CloseProject](#)

#### OpenProject

**Methode:** `OpenProject (strFileName als string) als boolean`

**Dispatch Id:** 3

##### Beschreibung:

`OpenProject` lädt die Datei `strFileName` als das neue Projekt in das Control. Wenn das Platzhalterfenster eine [PlaceholderWindowID](#) hat, die nicht mit `XMLSpyXProjectWindow (=3)` übereinstimmt, schlägt die Methode fehl.

#### CloseProject

**Methode:** `CloseProject ()`

**Dispatch Id:** 4

##### Beschreibung:

`CloseProject` schließt das in das Control geladene Projekt. Wenn das Platzhalterfenster eine [PlaceholderWindowID](#) hat, die nicht mit `StyleVisionXProjectWindow (=3)` übereinstimmt, schlägt die Methode fehl.

#### Events

Das `StyleVisionControlPlaceholder` ActiveX Control stellt die folgenden Verbindungspunkt-Events zur Verfügung:

[OnModifiedFlagChanged](#)

#### OnModifiedFlagChanged

**Event:** `OnModifiedFlagChanged (i_bIsModified als boolean)`

**Dispatch Id:** 1

##### Beschreibung:

Dieses Event wird nur bei Platzhalter-Controls mit einer [PlaceholderWindowID](#) von `StyleVisionXProjectWindow (=3)` ausgelöst. Das Event wird immer dann ausgelöst, wenn der Projektinhalt sich zwischen dem Status "geändert" und "nicht geändert" ändert. Der Parameter `i_bIsModified` ist `true`, wenn sich der Projektinhalt vom ursprünglichen Inhalt unterscheidet und `false`, wenn dies nicht der Fall ist.

#### OnSetLabel

**Event:** `OnSetLabel (i_strLabel als String)`

**Dispatch Id:** 1000

**Beschreibung:**

Dieses Event wird immer dann ausgelöst, wenn die Bezeichnung eines Platzhalter-Control-Fensters geändert wird.

## Enumerationen

Es sind die folgenden Enumerationen definiert:

[IActiveXIntegrationLevel](#)  
[StyleVisionControlPlaceholderWindow](#)

### IActiveXIntegrationLevel

Mögliche Werte für die Eigenschaft [IntegrationLevel](#) für das StyleVisionControl.

```
IActiveXIntegrationOnApplicationLevel = 0
IActiveXIntegrationOnDocumentLevel = 1
```

### StyleVisionControlPlaceholderWindow

Diese Enumeration enthält die Liste der unterstützten zusätzlichen StyleVision Fenster.

```
StyleVisionControlNoToolWnd = -1
StyleVisionControlProjectWnd = 0
StyleVisionControlDesignOverviewWnd = 1
StyleVisionControlSchemaSourcesWnd = 2
StyleVisionControlDesignTreeWnd = 3
StyleVisionControlStyleRepositoryWnd = 4
StyleVisionControlContextPropertiesWnd = 5
StyleVisionControlContextStylesWnd = 6
StyleVisionControlMessageWnd = 7
```



## **Kapitel 21**

---

### **Anhänge**

## 21 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](#)

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
- XPath 2.0 und XQuery 1.0 Funktionen
- Erweiterungsfunktionen (Java, .NET und MSXSL)

### [DatentypenDB2XSD](#)

Wenn Datenbankfelder in XML-Nodes konvertiert werden, werden die Datenbank-Datentypen in XML-Schema-Datentypen konvertiert. Im Anhang sind die Zuordnungen für die folgenden Quelldatenbanken aufgelistet.

- MS Access
- MS SQL Server
- MySQL
- Oracle
- ODBC
- ADO
- Sybase

### [Technische Daten](#)

Enthält technische Informationen zu StyleVision.

- Betriebssystem und Arbeitsspeicheranforderungen
- Altova XML Parser
- Altova XSLT- und XQuery-Prozessoren
- Unicode-Unterstützung
- Internet-Verwendung

### [Lizenzinformationen](#)

Enthält Informationen über Vertrieb und Lizenzierung von StyleVision.

- Electronic Software Distribution
- Lizenzüberwachung
- Copyright
- Endbenutzerlizenzvereinbarung

## 21.1 XSLT-Prozessorinformationen

In diesem Abschnitt finden Sie Informationen über implementierungsspezifische Funktionen des [Altova XSLT 1.0-Prozessors](#) und des [Altova XSLT 2.0-Prozessors](#).

## XSLT 1.0-Prozessor: Implementierungsinformationen

Der Altova XSLT 1.0-Prozessor ist in die Altova XML-Produkte XMLSpy, StyleVision, Authentic und MapForce integriert. Außerdem steht er im kostenlosen AltovaXML-Paket zur Verfügung. Im Altova XSLT 1.0-Prozessor sind die W3C-Spezifikationen implementiert und entsprechen der [XSLT 1.0 Recommendation vom 16. November 1999](#) und der [XPath 1.0 Recommendation vom 16. November 1999](#). Im Folgenden sind Einschränkungen und implementierungsspezifisches Verhalten aufgelistet.

### Einschränkungen

- Die Elemente **xsl: preserve-space** und **xsl: strip-space** werden nicht unterstützt.
- Wenn das Attribut `method` von **xsl: output** auf HTML gesetzt ist, oder wenn standardmäßig HTML output ausgewählt ist, werden Sonderzeichen in der XML- oder XSLT-Datei direkt als Sonderzeichen in die HTML-Dokument eingefügt und nicht als HTML-Zeichenreferenzen in der Ausgabe. So wird z.B. das Zeichen `&#160;` (die Dezimalzeichenreferenz für ein geschütztes Leerzeichen) nicht als `&nbsp;` in den HTML-Code eingefügt sondern direkt als geschütztes Leerzeichen.

### Behandlung der Implementierung von Nur-Whitespace Nodes in XML-Quelldokumenten

Aus den XML-Daten (und somit dem XML-Infoset), die an den Altova XSLT 1.0-Prozessor übergeben werden, werden boundary-whitespace-only Text Nodes entfernt. (Ein boundary-whitespace-only Text Node ist ein Textnode, der nur Whitespaces enthält und der zwischen zwei Elementen innerhalb eines mixed-content-Elements vorkommt). Dies kann sich auf den Wert auswirken, der von den Funktionen `fn: position()`, `fn: last()` und `fn: count()` zurückgegeben wird.

In jeder Node-Auswahl, bei der auch Text Nodes ausgewählt werden, sind normalerweise auch Boundary-whitespace-only Text Nodes enthalten. Da jedoch beim XML Infoset, das von den Altova-Prozessoren verwendet wird, boundary-whitespace-only Text Nodes entfernt werden, sind diese Nodes im XML Infoset nicht vorhanden. Folglich unterscheidet sich die Größe der Auswahl und die Nummerierung der Nodes von einer, in der diese Text Nodes enthalten sind. Aus diesem Grund können sich die Ergebnisse der Funktionen `fn: position()`, `fn: last()` und `fn: count()` von denen anderer Prozessoren unterscheiden.

Am häufigsten tritt die Situation, dass boundary-whitespace-only Text Nodes als gleichrangige Elemente anderer Elemente überprüft werden, dann auf, wenn `xsl: apply-templates` verwendet wird, um Templates anzuwenden. Wenn die Funktionen `fn: position()`, `fn: last()` und `fn: count()` in Patterns mit einer Namensüberprüfung verwendet werden (z.B. `para[3]`, kurz für `para[position()=3]`), sind boundary-whitespace-only Nodes irrelevant, da nur die benannten Elemente (`para` im obigen Beispiel) ausgewählt werden. (Beachten Sie jedoch, dass boundary-whitespace-only Nodes in Patterns, in denen ein Platzhalterzeichen wie z.B. `*[10]` verwendet wird, sehr wohl relevant sind).

**Hinweis:** Wenn der boundary-whitespace-only Text Node in der Ausgabe benötigt wird, fügen Sie das erforderliche Whitespace-Zeichen in eines der benachbarten Child-Elemente ein. So erzeugt z.B. das XML-Fragment:

```
<para>This is bold <i>italic</i>.</para>
```

bei Verarbeitung mit dem XSLT Template

```
<xsl:template match="para">
 <xsl:apply-templates/>
</xsl:template>
```



folgendes Resultat:

This is ***bolditalic***.

Um in der Ausgabe ein Leerzeichen zwischen `bold` und `italic` zu erhalten, fügen Sie entweder in das Element `<b>` oder `<i>` in der XML-Quelle ein Leerzeichen ein. z.B.:

```
<para>This is bold <i> italic</i>. </para> or
<para>This is bold<#x20; <i>italic</i>. </para> or
<para>This is bold<i><#x20;italic</i>. </para>
```

Wenn eines der oben genannten `para`-Elemente mit demselben XSLT Template verarbeitet wird, erhalten Sie folgendes Ergebnis:

This is bold italic.

**XSLT 2.0-Prozessor: Implementierungsinformationen**

Der Altova XSLT 2.0-Prozessor ist in die Altova XML-Produkte XMLSpy, StyleVision, Authentic und MapForce integriert. Außerdem steht er im kostenlosen AltovaXML-Paket zur Verfügung. In diesem Abschnitt werden die implementierungsspezifischen Aspekte des Prozessors beschrieben. Im ersten Teil des Abschnitts finden Sie allgemeine Informationen über den Prozessor. Im Anschluss daran finden Sie eine Liste der implementierungsspezifischen Aspekte der XSLT 2.0-Funktionen.

Informationen über das implementierungsspezifische Verhalten von XPath 2.0-Funktionen finden Sie im Abschnitt [XPath 2.0- und XQuery 1.0-Funktionen](#).

## Allgemeine Informationen

Der Altova XSLT 2.0-Prozessor entspricht der [XSLT 2.0 Recommendation](#) vom 23 January 2007. Beachten Sie bitte die folgenden allgemeinen Informationen über den Prozessor.

### Rückwärtskompatibilität

Der Altova XSLT 2.0-Prozessor ist rückwärtskompatibel. Die Rückwärtskompatibilität des XSLT 2.0.-Prozessors kommt nur dann zum Einsatz, wenn Sie den XSLT 2.0-Prozessor von Altova XML zur Verarbeitung eines XSLT 1.0 Stylesheets verwenden. Beachten Sie, dass sich das Ergebnis des XSLT 1.0-Prozessors und des rückwärtskompatiblen XSLT 2.0-Prozessors unter Umständen unterscheiden kann.

Bei allen anderen Altova-Produkten wird Rückwärtskompatibilität nie benötigt, da automatisch der entsprechende Prozessor für die Transformation ausgewählt wird. Angenommen, Sie legen fest, dass ein bestimmtes XML-Dokument in XMLSpy mit einem XSLT 1.0 Stylesheet verarbeitet werden soll. Bei Aufruf des Transformationsbefehls wählt XMLSpy zur Transformation automatisch den XSLT 1.0-Prozessor aus.

### Anmerkung:

Diese Auswahl basiert auf dem Wert im Attribut `version` des Stylesheet-Elements `stylesheet` oder `transform`.

### 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:    | <a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>             |
| XPath 2.0-Funktionen | fn:    | <a href="http://www.w3.org/2005/xpath-functions">http://www.w3.org/2005/xpath-functions</a> |

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"
 ...
/>
```

Beachten Sie bitte die folgenden Punkte:

- Der Altova XSLT 2.0-Prozessor verwendet als Default Functions 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.

- In den Candidate Recommendations vom 23 January 2007 wurden die Datentypen `untypedAtomic` und `duration` (`dayTimeDuration` und `yearMonthDuration`), die sich zuvor im XPath Datatypes Namespace befanden (normalerweise mit dem Präfix `xdtd`) in den XML-Schema-Namespace verschoben.
- 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')`.

### Schemafähigkeit

Der Altova XSLT 2.0-Prozessor ist schemafähig.

### Whitespaces im XML-Dokument

Standardmäßig entfernt der Altova XSLT 2.0-Prozessor alle boundary whitespaces aus boundary-whitespace-only Nodes im XML-Quelldokument. Dies wirkt sich auf die Werte aus, die die Funktionen `fn:position()`, `fn:last()`, `fn:count()` und `fn:deep-equal()` zurückgeben. Nähere Informationen dazu finden Sie in den Abschnitten über XPath 2.0- und XQuery 1.0-Funktionen unter [Nur-Whitespace Nodes im XML-Quelldokument](#).

**Hinweis:** Wenn in der Ausgabe ein boundary-whitespace-only-Text Node erforderlich ist, fügen Sie das gewünschte Whitespace-Zeichen in eines der beiden benachbarten Child-Elemente ein. So erzeugt z.B. das XML-Fragment:

```
<para>This is bold <i>italic</i>.</para>
```

bei Verarbeitung mit dem XSLT Template

```
<xsl:template match="para">
 <xsl:apply-templates/>
</xsl:template>
```

folgendes Resultat:

```
This is bolditalic.
```

Um in der Ausgabe ein Leerzeichen zwischen `bold` und `italic` zu erhalten, fügen Sie entweder in das Element `<b>` oder `<i>` in der XML-Quelle ein Leerzeichen ein. z.B.:

```
<para>This is bold <i> italic</i>.</para> oder
<para>This is bold<#x20; <i>italic</i>.</para> oder
<para>This is bold<i><#x20;italic</i>.</para>
```

Wenn nun ein solches XML-Fragment mit demselben XSLT Template verarbeitet wird, erhalten Sie folgendes Ergebnis:

```
This is bold italic.
```

### XSLT 2.0-Elemente und -Funktionen

Einschränkungen und implementierungsspezifisches Verhalten von XSLT 2.0-Elementen und -Funktionen werden im Abschnitt [XSLT 2.0-Elemente und -Funktionen](#) aufgelistet.

### XPath 2.0-Funktionen

Implementierungsspezifisches Verhalten von XPath 2.0-Funktionen wird im Abschnitt [XPath 2.0- und XQuery 1.0-Funktionen](#) aufgelistet.

## XSLT 2.0-Elemente und -Funktionen

### Einschränkungen

Die Elemente `xsl: preserve-space` und `xsl: strip-space` werden nicht unterstützt.

### Implementierungsspezifisches Verhalten

Im Folgenden finden Sie eine Beschreibung, wie der Altova XSLT 2.0-Prozessor implementierungsspezifische Aspekte des Verhaltens bestimmter 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 2.0, XPath 2.0 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 Kodierungen unterstützt: `binarytobase16` und `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 Kodierungen `x-binarytobase16` und `x-binarytobase64` unterstützt.

**Anmerkung:** Die folgenden Kodierungswerte, die in früheren Versionen von AltovaXML implementiert waren, werden nun nicht mehr verwendet: `base16tobinary`, `base64tobinary`, `binarytobase16` und `binarytobase64`.

## XPath 2.0- und XQuery 1.0-Funktionen

Die XPath 2.0- und XQuery 1.0-Funktionen werden von folgenden Prozessoren überprüft:

- dem **Altova XPath 2.0-Prozessor**, der (i) eine Komponente des Altova XSLT 2.0-Prozessors ist und (ii) im XPath Evaluator von Altova XMLSpy verwendet wird, um XPath-Ausdrücke hinsichtlich des in XMLSpy aktiven XML-Dokuments auszuwerten.
- dem **Altova XQuery 1.0-Prozessor**.

In diesem Abschnitt wird beschrieben, wie XPath 1.0- und XQuery 1.0-Funktionen vom Altova XPath 2.0-Prozessor und dem Altova XQuery 1.0-Prozessor behandelt werden. Es sind nur die Funktionen aufgelistet, die implementierungsspezifisch behandelt werden oder bei denen sich eine einzelne Funktion in einer der drei Umgebungen, in denen sie verwendet wird (also in XSLT 2.0, in XQuery 1.0 und im XPath Evaluator von XMLSpy), anders verhält. Beachten Sie, dass in diesem Abschnitt nicht beschrieben wird, wie diese Funktionen verwendet werden. Nähere Informationen über die Verwendung dieser Funktionen finden Sie im [XQuery 1.0 and XPath 2.0 Functions and Operators PR des W3C](#) vom 23 January 2007.

## Allgemeine Informationen

### Standardkonformität

- Der Altova XPath 2.0-Prozessor implementiert die [XPath 2.0 Recommendation](#) vom 23 January 2007. Der Altova XQuery 1.0-Prozessor implementiert die [XQuery 1.0 Recommendation](#) vom 23 January 2007. Die Unterstützung der XPath 2.0- und XQuery 1.0-Funktionen in diesen beiden Prozessoren entspricht den Vorgaben in den [XQuery 1.0 and XPath 2.0 Functions and Operators Recommendation](#) vom 23 January 2007.
- Der Altova XPath 2.0-Prozessor entspricht den Vorgaben für [XML 1.0 \(Fourth Edition\)](#) und [XML Namespaces \(1.0\)](#).

### Default Functions Namespace

Der Default Functions Namespace wurde dem Standard entsprechend eingestellt. Funktionen können daher ohne Präfix aufgerufen werden.

### Boundary-whitespace-only-Nodes in XML-Quelldokumenten

Aus den XML-Daten (und somit dem XML-Infoset), die an den Altova XPath 2.0- und den Altova XQuery 1.0-Prozessor übergeben werden, werden boundary-whitespace-only Text Nodes entfernt. (Ein boundary-whitespace-only Text Node ist ein Child-Textnode, der nur Whitespaces enthält und der zwischen zwei Elementen innerhalb eines mixed-content-Elements vorkommt). Dies kann sich auf den Wert auswirken, der von den Funktionen `fn:position()`, `fn:last()`, `fn:count()` und `fn:deep-equal()` zurückgegeben wird.

In jeder Node-Auswahl, bei der auch Text Nodes ausgewählt werden, sind normalerweise auch Boundary-whitespace-only Text Nodes enthalten. Da jedoch beim XML Infoset, das von den Altova-Prozessoren verwendet wird, boundary-whitespace-only Text Nodes entfernt werden, sind diese Nodes im XML Infoset nicht vorhanden. Folglich unterscheidet sich die Größe der Auswahl und die Nummerierung der Nodes von einer, in der diese Text Nodes enthalten sind. Aus diesem Grund können sich die Ergebnisse der Funktionen `fn:position()`, `fn:last()`, `fn:count()` und `fn:deep-equal()` von denen anderer Prozessoren unterscheiden.

Am häufigsten tritt die Situation, dass boundary-whitespace-only Text Nodes als gleichrangige Elemente anderer Elemente überprüft werden, dann auf, wenn `xsl:apply-templates` verwendet wird, um Templates anzuwenden. Wenn die Funktionen `fn:position()`, `fn:last()` und `fn:count()` in Patterns mit einer Namensüberprüfung verwendet werden (z.B. `para[3]`, kurz für `para[position()=3]`), sind boundary-whitespace-only Nodes irrelevant, da nur die benannten Elemente (`para` im obigen Beispiel) ausgewählt werden. (Beachten Sie jedoch, dass boundary-whitespace-only Nodes in Patterns, in denen ein Platzhalterzeichen wie z.B. `*[10]` verwendet wird, sehr wohl relevant sind).

### Numerische Notation

Wenn bei der Ausgabe ein `xs:double` in einen String konvertiert wird, wird die logarithmische Darstellung (z.B. `1.0E12`) verwendet, wenn der absolute Wert kleiner als 0,000001 oder größer als 1.000.000 ist. Andernfalls wird der Wert als Dezimalwert oder Integer angegeben.

### 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 `fn:implicit-timezone()` überprüft werden.

**Collations**

Es wird nur die Unicode Codepoint Collation unterstützt. Es können keine anderen Collations verwendet werden. Der Vergleich von Strings, u.a. auch für die Funktionen `fn:max` und `fn:min` basiert auf dieser Collation.

**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 `fn:in-scope-prefixes()`, `fn:namespace-uri()` und `fn:namespace-uri-for-prefix()`.

**Static Type Extensions**

Die optionale Funktion zum Überprüfen von statischen Typen wird nicht unterstützt.



### Unterstützung von Funktionen

In der nachfolgenden Tabelle wird das implementierungsspezifische Verhalten bestimmter Funktionen (in alphabetischer Reihenfolge) aufgelistet. Beachten Sie bitte die folgenden allgemeinen Punkte:

- Wenn bei einer Funktion eine Sequenz von einem Datenelement als Argument erwartet wird und eine Sequenz von mehr als einem Datenelement gesendet wird, wird im allgemeinen 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.

| Funktionsname         | Anmerkungen                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>base-uri</code> | <ul style="list-style-type: none"><li>• Wenn im XML-Quelldokument externe Entities verwendet werden und ein Node in der externen Entity als Input-Node-Argument der Funktion <code>base-uri()</code> definiert ist, wird trotzdem die Basis-URI des inkludierenden XML-Dokuments verwendet und nicht die Basis-URI der externen Entity.</li><li>• Die Basis-URI eines Node im XML-Dokument kann mit Hilfe des Attributs <code>xml:base</code> geändert werden.</li></ul> |

|                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| collection                                         | <ul style="list-style-type: none"> <li>• Das Argument ist eine relative URI, die gegen die aktuelle Basis-URI aufgelöst wird.</li> <li>• Wenn mit der aufgelösten URI eine XML-Datei identifiziert wird, so wird diese XML-Datei als Katalog behandelt, der eine Sammlung von Dateien referenziert. Diese Datei muss die folgende Form haben: <pre>&lt;collection&gt;   &lt;doc href="uri-1" /&gt;   &lt;doc href="uri-2" /&gt;   &lt;doc href="uri-3" /&gt; &lt;/collection&gt;</pre> Die von den href Attributen referenzierten Dateien werden geladen und ihre Dokument-Nodes werden als Sequenz zurückgegeben. </li> <li>• Wenn mit der aufgelösten URI keine XML-Datei mit der oben beschriebenen Katalogstruktur identifiziert wird, dann wird der Argument-String (in dem die Platzhalterzeichen ? und * zulässig sind) als Suchstring verwendet. XML-Dateien, deren Namen dem Suchausdruck entsprechen, werden geladen und ihre Dokument-Nodes werden als Sequenz zurückgegeben. Siehe Beispiel unten.</li> <li>• XSLT-Beispiel: Der Ausdruck <code>collection("c:\MyDocs\*.xml")//Title</code> gibt eine Sequenz aller DocTitle Elemente in den .xml Dateien im Ordner MyDocs zurück.</li> <li>• XQuery-Beispiel: Der Ausdruck <code>{for \$i in collection(c:\MyDocs\*.xml) return element doc{base-uri(\$i)}}</code> gibt die Basis-URLs aller .xml Dateien im Ordner MyDocs zurück, wobei jede URI sich innerhalb eines doc Elemente befindet.</li> <li>• Die Standard-Collection ist leer.</li> </ul> |
| count                                              | <ul style="list-style-type: none"> <li>• Siehe Anmerkungen zu Whitespaces im Abschnitt <a href="#">Allgemeine Informationen</a>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| current-date,<br>current-dateTime,<br>current-time | <ul style="list-style-type: none"> <li>• The argument is a relative URI that is resolved against the current base URI.</li> <li>• If the resolved URI identifies an XML file, then this XML file is treated as a catalog which references a collection of files. This file must have the form: <pre>&lt;collection&gt;   &lt;doc href="uri-1" /&gt;   &lt;doc href="uri-2" /&gt;   &lt;doc href="uri-3" /&gt; &lt;/collection&gt;</pre> The files referenced by the href attributes are loaded, and their document nodes are returned as a sequence. </li> <li>• If the resolved URI does not identify an XML file with the catalog structure described above, then the argument string is used as a file-search expression (in which wildcards such as ? and * ) are allowed. The specified directory, as identified by the base URI and search string, is searched. XML files with names that matches the search expression are loaded, and their document nodes are returned as a sequence.</li> <li>• The default collection is empty.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| deep-equal        | <ul style="list-style-type: none"> <li>• Siehe Anmerkung zu Whitespace im Abschnitt <a href="#">Allgemeine Informationen</a>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| doc               | <ul style="list-style-type: none"> <li>• Es wird nur dann ein Fehler ausgegeben, wenn unter dem angegebenen Pfad keine XML-Datei vorhanden ist oder wenn die Datei nicht wohlgeformt ist. Die Datei wird validiert, wenn ein Schema vorhanden ist. Wenn die Datei nicht gültig ist, so wird die ungültige Datei ohne Schema-Informationen geladen.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                     |
| id                | <ul style="list-style-type: none"> <li>• In einem wohlgeformten aber ungültigen Dokument, das zwei oder mehr Elemente mit demselben ID-Wert enthält, wird das erste Element in der Dokumentenreihenfolge zurückgegeben.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| in-scope-prefixes | <ul style="list-style-type: none"> <li>• Im XML-Dokument dürfen nur Default Namespaces entdeclariert werden. Doch selbst wenn ein Default Namespace an einem Element-Node entdeclariert wird, wird das Präfix für den Default Namespace, welches der Nulllängenstring ist, für diesen Node zurückgegeben.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                              |
| last              | <ul style="list-style-type: none"> <li>• Siehe Anmerkung zu Whitespace im Abschnitt <a href="#">Allgemeine Informationen</a>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| lower-case        | <ul style="list-style-type: none"> <li>• Es wird der Unicode-Zeichensatz unterstützt.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| normalize-unicode | <ul style="list-style-type: none"> <li>• Es werden die Normalisierungsformulare NFC, NFD, NFKC und NFKD unterstützt.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| position          | <ul style="list-style-type: none"> <li>• Siehe Anmerkung zu Whitespace im Abschnitt <a href="#">Allgemeine Informationen</a>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| resolve-uri       | <ul style="list-style-type: none"> <li>• Wenn das zweite, optionale Argument weggelassen wird, wird die aufzulösende URI (das erste Argument) gegen die Basis-URI aus dem statischen Kontext aufgelöst. Dies ist die URI des XSLT Stylesheets oder die im Prolog des XQuery-Dokuments angegebene Basis-URI.</li> <li>• Die relative URI (das erste Argument) wird nach dem letzten "/" in der Pfadnotation der Basis-URI-Notation angehängt.</li> <li>• Wenn der Wert des ersten Arguments der Nulllängenstring ist, wird die Basis-URI aus dem statischen Kontext zurückgegeben und diese URI inkludiert den Dateinamen des Dokuments, von dem die Basis-URI des statischen Kontexts abgeleitet wird (z.B. die XSLT- oder XML-Datei).</li> </ul> |
| static-base-uri   | <ul style="list-style-type: none"> <li>• Die Basis-URI aus dem statischen Kontext ist die Basis-URI des XSLT Stylesheets oder die im Prolog des XQuery-Dokuments angegebene Basis-URI.</li> <li>• Bei Verwendung des XPath Evaluators in der XMLSpy IDE ist die Basis-URI aus dem statischen Kontext die URI des aktiven XML-Dokuments.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                |

|            |                                                                                             |
|------------|---------------------------------------------------------------------------------------------|
| upper-case | <ul style="list-style-type: none"><li>• Der Unicode-Zeichensatz wird unterstützt.</li></ul> |
|------------|---------------------------------------------------------------------------------------------|

## Erweiterungen

Es gibt in Programmiersprachen wie Java und C# eine Reihe von fertigen Funktionen, die nicht als XPath 2.0 / XQuery 1.0-Funktionen oder XSLT 2.0-Funktionen zur Verfügung stehen. Ein gutes Beispiel für solche Funktionen 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.

Die Altova-Prozessoren (XSLT 1.0, XSLT 2.0 und XQuery 1.0), die in einer Reihe von Altova-Produkten verwendet werden, unterstützen die Verwendung von Erweiterungsfunktionen in Java und .NET. Zusätzlich unterstützen die Altova XSLT-Prozessoren MSXSL Scripts für XSLT 1.0 und 2.0.

Beachten Sie, dass Erweiterungsfunktionen immer von XPath-Ausdrücken aus aufgerufen werden. In diesem Abschnitt wird beschrieben, wie Sie Erweiterungsfunktionen und MSXSL Scripts in Ihren XSLT Stylesheets verwenden können. Diese Beschreibungen finden Sie in den folgenden Abschnitten:

- [Java-Erweiterungsfunktionen](#)
- [.NET-Erweiterungsfunktionen](#)
- [MSXSL Scripts für XSLT](#)
- [Altova Erweiterungsfunktionen](#)

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) installiert sein oder es muss Zugriff auf eine solche bestehen.

## 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](#)
- [Java: Statische Methoden und statische Felder](#)
- [Java: Instanzmethoden und Instanzfelder](#)
- [Datentypen: XSLT/XQuery in Java](#)
- [Datentypen: Java in XSLT/XQuery](#)

## 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 `präfix:` 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 `präfix:` 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.

## 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](#) und [Benutzerdefinierte Jar-Dateien](#) 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.

### Benutzerdefinierte Klassendateien

Wenn der Zugriff über eine Klassendatei erfolgt, gibt es zwei Möglichkeiten:

- Die Klassendatei befindet sich in einem Paket. Die XSLT- oder XQuery-Datei befindet sich im selben Ordner wie das Java-Paket.
- Die Klassendatei befindet sich nicht in einem Paket. Die XSLT- oder XQuery-Datei befindet sich im selben Ordner wie die Klassendatei.
- Die Klassendatei befindet sich in einem Paket. Die XSLT- oder XQuery-Datei befindet sich in irgendeinem beliebig gewählten Ordner.
- Die Klassendatei befindet sich nicht in einem Paket. Die XSLT- oder XQuery-Datei befindet sich in irgendeinem beliebig gewählten Ordner.

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()" />

</xsl:template>

</xsl:stylesheet>

```

**Die Klassendatei befindet sich nicht in einem Paket, die XSLT/XQuery-Datei befindet sich im selben Ordner wie die Klassendatei**

Im Beispiel unten wird die Methode `getVehicleType()` der Klasse `Car` class des Pakets `com.altova.extfunc` aufgerufen. Die Klassendatei `Car` class befindet sich im folgenden Ordner: `JavaProject/com/altova/extfunc`. 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()" />

</xsl:template>

</xsl:stylesheet>

```

**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` class 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)"/>
 </xsl:template>

</xsl:stylesheet>

```

### Die Klassendatei befindet sich nicht in einem Paket, die XSLT/XQuery-Datei befindet sich in einem beliebigen Ordner

Im Beispiel unten wird die Methode `getCarColor()` der Klasse `Car` class 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. Der Pfad der Klassendatei wird in der Namespace-URI als Abfragestring definiert. 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)"/>
 </xsl:template>

</xsl:stylesheet>

```

**Anmerkung:** Wenn ein Pfad über eine Erweiterungsfunktion angegeben wird, wird er zum `ClassLoader` hinzugefügt.

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()
```

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/Carl. jar! /" >
<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
 <xsl:variable name="myCar" select="car: Carl. new(' red')" />
 <a><xsl: value-of select="car: Carl. getCarColor($myCar) "/>
</xsl:template>

<xsl:template match="car"/>

</xsl:stylesheet>
```

**Anmerkung:** Wenn ein Pfad über eine Erweiterungsfunktion angegeben wird, wird er zum ClassLoader hinzugefügt.

## 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](#), dann gibt die Java-Erweiterungsfunktion eine Sequenz zurück, bei der es sich um einen 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](#)):  

```
<xsl: value-of select="date: toString(date: new()) " xmlns: date="
java: java. util. Date" />
```

### 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>
```

### 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:

```
<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.

### 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.

|                         |                                                                                                                                                                                                |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>xs:string</code>  | <code>java.lang.String</code>                                                                                                                                                                  |
| <code>xs:boolean</code> | <code>boolean (primitive)</code> , <code>java.lang.Boolean</code>                                                                                                                              |
| <code>xs:integer</code> | <code>int</code> , <code>long</code> , <code>short</code> , <code>byte</code> , <code>float</code> , <code>double</code> und die Wrapper-Klassen davon wie z.B. <code>java.lang.Integer</code> |
| <code>xs:float</code>   | <code>float (primitive)</code> , <code>java.lang.Float</code> , <code>double (primitive)</code>                                                                                                |
| <code>xs:double</code>  | <code>double (primitive)</code> , <code>java.lang.Double</code>                                                                                                                                |
| <code>xs:decimal</code> | <code>float (primitive)</code> , <code>java.lang.Float</code> , <code>double (primitive)</code> , <code>java.lang.Double</code>                                                                |

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 einige 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 Datentypes 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.

#### 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 `java.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 Darstellung des benötigten Typs entspricht, und anschließend z.B. mit Hilfe des Ausdrucks `cast as` in den benötigten Typ konvertiert werden.



## .NET-Erweiterungsfunktionen

Wenn Sie 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](#)
- [.NET: Statische Methoden und statische Felder](#)
- [.NET: Instanzmethoden und Instanzfelder](#)
- [Datentypen: XSLT/XQuery in .NET](#)
- [Datentypen: .NET in XSLT/XQuery](#)

### 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 `präfix: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:

|                          |                                                                                                                                                                                                                                               |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>asm</code>         | Der Name der zu ladenden Assembly                                                                                                                                                                                                             |
| <code>ver</code>         | Die Versionsnummer: eine Maximalzahl von vier Ganzzahlen, die durch Punkte getrennt sind                                                                                                                                                      |
| <code>sn</code>          | Das Key Token des Strong Name der Assembly (16 Hex-Stellen).                                                                                                                                                                                  |
| <code>from</code>        | 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.                             |
| <code>partialname</code> | Der partielle Name der Assembly. Er wird für <code>Assembly.LoadWith.PartialName()</code> bereitgestellt, welches versucht wird, die Assembly zu laden. Wenn <code>partialname</code> vorhanden ist, werden alle anderen Parameter ignoriert. |
| <code>loc</code>         | Die Locale, z.B. <code>en-US</code> . Die Standardeinstellung ist <code>neutral</code>                                                                                                                                                        |

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:mysns="clitype:System.Environment"
```

Ein Beispiel für eine Namespace Deklaration in XSLT, die die zu ladende Klasse als `Trade.Forward.Scrip` identifiziert.

```
xmlns:mysns="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">
 <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.

### .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](#), 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) auf den erforderlichen XPath-Datentyp gecastet 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](#)):  

```
<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( data: get_Month( date: new( 2008, 4, 29) ) )
" xmlns:date="clitype: System. DateTime" />
```

.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"/>
```

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>
```

.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. Dieses 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.

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.

<code>xs: string</code>	<code>StringValue</code> , <code>string</code>
<code>xs: boolean</code>	<code>BooleanValue</code> , <code>bool</code>
<code>xs: integer</code>	<code>IntegerValue</code> , <code>decimal</code> , <code>long</code> , <code>integer</code> , <code>short</code> , <code>byte</code> , <code>double</code> , <code>float</code>
<code>xs: float</code>	<code>FloatValue</code> , <code>float</code> , <code>double</code>
<code>xs: double</code>	<code>DoubleValue</code> , <code>double</code>

<code>xs:decimal</code>	<code>DecimalValue, decimal, double, float</code>

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 einige 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 Datentypes 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.

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.

MSXSL Scripts 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>

```

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.

Altova Erweiterungsfunktionen

Altova Erweiterungsfunktionen befinden sich im Namespace `http://www.altova.com/xslt-extension` und werden in diesem Abschnitt mit dem Präfix `altova:` gekennzeichnet, das an den oben angegebenen Namespace gebunden ist.

In der aktuellen Version Ihres Altova-Produkts werden die folgenden Erweiterungsfunktionen auf die unten beschriebene Art unterstützt. Beachten Sie jedoch, dass eine oder mehrere dieser Funktionen in zukünftigen Produktversionen eventuell nicht mehr unterstützt werden, bzw. dass sich das Verhalten einzelner Funktionen ändern kann. Um Informationen über die Unterstützung für Altova Erweiterungsfunktionen in der jeweiligen Release zu erhalten, schlagen Sie bitte in der Dokumentation der jeweils aktuellen Release nach.

Allgemeine Funktionen

- [altova: evaluate\(\)](#)
- [altova: distinct-nodes\(\)](#)
- [altova: encode-for-rtf\(\)](#)
- [altova: xbrl-labels\(\)](#)
- [altova: xbrl-footnotes\(\)](#)
- [altova: generate-auto-number\(\)](#)
- [altova: reset-auto-number\(\)](#)
- [altova: get-temp-folder\(\)](#)

Diagrammfunktionen (Nur Enterprise und Reporting Edition)

Die Altova-Erweiterungsfunktionen für Diagramme werden nur in der Enterprise und der Reporting Edition von Altova Produkten unterstützt und gestatten die Erstellung von Diagrammen anhand von XML-Daten. Die Diagrammfunktionen wurden in zwei Gruppen unterteilt:

- Funktionen zum Generieren und Speichern von Diagrammen
- Funktionen zum Erstellen von Diagrammen

Ein dritter Abschnitt enthält eine Liste der Diagrammdaten-XML-Struktur, anhand der Diagramme generiert werden können und zum Schluss finden Sie im XSLT-Beispieldokument ein Beispiel, wie Diagrammfunktionen verwendet werden können, um anhand von XML-Daten Diagramme zu generieren.

Allgemeine Funktionen

In der aktuellen Version Ihres Altova-Produkts werden die folgenden Erweiterungsfunktionen auf die unten beschriebene Art unterstützt. Beachten Sie jedoch, dass eine oder mehrere dieser Funktionen in zukünftigen Produktversionen eventuell nicht mehr unterstützt werden, bzw. dass sich das Verhalten einzelner Funktionen ändern kann. Um Informationen über die Unterstützung für Altova Erweiterungsfunktionen in der jeweiligen Release zu erhalten, schlagen Sie bitte in der Dokumentation der jeweils aktuellen Release nach.

- [altova: evaluate\(\)](#)
- [altova: distinct-nodes\(\)](#)
- [altova: encode-for-rtf\(\)](#)
- [altova: xbrl-labels\(\)](#)
- [altova: xbrl-footnotes\(\)](#)
- [altova: generate-auto-number\(\)](#)
- [altova: reset-auto-number\(\)](#)

- [altova: get-temp-folder\(\)](#)

altova: evaluate()

Die `altova: evaluate()` Funktion erhält einen XPath-Ausdruck als obligatorisches Argument, der als String übergeben wird, und gibt das Resultat des ausgewerteten Ausdrucks zurück.

```
altova: evaluate( XPathExp as xs: string)
```

Beispiel:

```
altova: evaluate(' //Name[ 1]')
```

Beachten Sie im obigen Beispiel, dass der Ausdruck `//Name[ 1]` durch Einschließen in einfache Anführungszeichen als String übergeben wird. Die Funktion `altova: evaluate` gibt den Inhalt des ersten `Name`-Elements im Dokument zurück.

Die Funktion `altova: evaluate` kann zusätzliche (optionale) Argumente erhalten. Diese Argumente sind die Werte der einzelnen Variablen und haben die Namen `p1`, `p2`, `p3`... `pN`, die im XPath-Ausdruck verwendet werden können.

```
altova: evaluate( XPathExp as xs: string [, p1value ... pNvalue])
```

wobei

- die Variablennamen die Form `px` haben müssen, wobei `x` eine Ganzzahl ist
- die Reihenfolge der Argumente der Funktion vom zweiten Argument an der Reihenfolge der Variablen mit den Namen `p1` bis `pN` entspricht. Daher wird das zweite Argument der Wert der Variablen `p1`, das dritte Argument der der Variablen `p2`, usw.
- die Werte der Variablen den Typ `item*` haben müssen

Beispiel:

```
<xs1:variable name="xpath" select="'$p3, $p2, $p1'" />
<xs1:value-of select="altova:evaluate( $xpath, 10, 20, 'hi' )" />
Gibt "hi 20 10" zurück
```

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..

Im Folgenden ein weiteres Beispiel für die Verwendung:

```
<xs1:variable name="xpath" select="'$p1'" />
<xs1:value-of select="altova:evaluate( $xpath, //Name[1] )" />
Gibt den Wert des ersten Name Elements aus.

<xs1:variable name="xpath" select="'$p1'" />
<xs1: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:

```
<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`)

Variablen können in der Erweiterungsfunktion `altova:evaluate()` wie im Beispiel unten gezeigt verwendet werden:

- 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

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.

altova:distinct-nodes()

Die `altova:distinct-nodes()` Funktion 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.

```
altova:distinct-nodes( $arg as node()* ) as node()*
```

altova:encode-for-rtf()

Die `altova:encode-for-rtf()` Funktion konvertiert den Input-String in Code für RTF.

```
altova:encode-for-rtf( $inputstr as xs:string?,  
$preserveallwhitespace as xs:boolean,  
$preservenewlines as xs:boolean) as xs:string
```

Whitespaces und neue Zeilen werden gemäß dem für die entsprechenden Parameter definierten Booleschen Wert beibehalten.

altova:xbrl-labels()

Die `altova:xbrl-labels()` Funktion 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.

```
altova:xbrl-labels( $name as xs:QName, $file as xs:string ) as node()*
```

altova:xbrl-footnotes()

Die `altova:footnotes()` Funktion erhält einen Node als Input-Argument und gibt die durch den Input-Node referenzierte Gruppe der XBRL-Fußnoten-Nodes zurück.

```
altova:footnotes( $arg as node() ) as node()*
```

altova:generate-auto-number(id als xs:string, start-with als xs:integer, increment als xs:integer, reset-on-change als xs:string)

Generiert eine Reihe von Zahlen mit der definierten ID. Die Anfangszahl (Ganzzahl) und die Inkrementierung werden definiert.

altova:reset-auto-number(id als xs:string)

Diese Funktion setzt die automatische Nummerierung der mit dem ID-Argument definierten automatischen Nummerierung zurück. Die Reihe wird auf den Anfangsganzzahlwert der Reihen zurückgesetzt (siehe `altova:generate-auto-number` oben).

altova:get-temp-folder als xs:string

Ruft den temp-Ordner ab.

Barcode-Funktionen

Die Altova XSLT-Prozessoren verwenden zur Erstellung von Barcodes Java-Bibliotheken von Drittanbietern. Im Folgenden finden Sie die verwendeten Klassen und öffentlichen Methoden. Die Klassen befinden sich im Paket `AltovaBarcodeExtension.jar`, das im Ordner `C:\Program Files\Altova\CommonYYYY\jar` gespeichert ist.

Die verwendeten Java-Bibliotheken befinden sich in Unterordnern des Ordners `C:\Program Files\Altova\CommonYYYY\jar`:

- `barcode4j\barcode4j.jar` (Website: <http://barcode4j.sourceforge.net/>)
- `zxing\core.jar` (Website: <http://code.google.com/p/zxing/>)

Die Lizenzdateien befinden sich ebenfalls in den entsprechenden Ordnern.

Das Paket `com.altova.extensions.barcode`

Das Paket `com.altova.extensions.barcode` wird zum Generieren der meisten der Barcode-Typen verwendet.

Die folgenden Klassen werden verwendet:

```
public class BarcodeWrapper
    static BarcodeWrapper newInstance( String name, String msg, int dpi, int
orientation, BarcodePropertyWrapper[] arrProperties )
    double getHeightPlusQuiet()
    double getWidthPlusQuiet()
    org.w3c.dom.Document generateBarcodeSVG()
    byte[] generateBarcodePNG()
    String generateBarcodePngAsHexString()

public class BarcodePropertyWrapper Dient zum Speichern der Barcode-Eigenschaften,
die zu einem späteren Zeitpunkt dynamisch definiert werden
    BarcodePropertyWrapper( String methodName, String propertyValue )
    BarcodePropertyWrapper( String methodName, Integer propertyValue
)
    BarcodePropertyWrapper( String methodName, Double propertyValue )
    BarcodePropertyWrapper( String methodName, Boolean propertyValue )
    BarcodePropertyWrapper( String methodName, Character propertyValue )
    String getMethodName()
    Object getPropertyValue()

public class AltovaBarcodeClassResolver Registriert die Klasse class
com.altova.extensions.barcode.proxy.zxing.QRCodeBean zusätzlich zu den vom
org.krysalis.barcode4j.DefaultBarcodeClassResolver registrierten Klassen für die
qrcode Bean.
```

Das Paket `com.altova.extensions.barcode.proxy.zxing`

Das Paket `com.altova.extensions.barcode.proxy.zxing` wird zum Generieren des QRCode Barcodetyps verwendet.

Die folgenden Klassen werden verwendet:

Klasse **QRCodeBean**

- *Erweitert* `org.krysalis.barcode4j.impl.AbstractBarcodeBean`
 - *Erstellt ein* `AbstractBarcodeBean` **Interface** für `com.google.zxing.qrcode.encoder`
- ```
void generateBarcode(CanvasProvider canvasImp, String msg)
void setQRErrorCorrectionLevel(QRCodeErrorCorrectionLevel level)
BarcodeDimension calcDimensions(String msg)
double getVerticalQuietZone()
double getBarWidth()
```

Klasse **QRCodeErrorCorrectionLevel** *Fehlerkorrekturebene für den QRCode*

```
static QRCodeErrorCorrectionLevel byName(String name)
"L" = ~7% correction
"M" = ~15% correction
"H" = ~25% correction
"Q" = ~30% correction
```

**XSLT-Beispiel**

Im Folgenden sehen Sie ein XSLT-Beispiel für die Verwendung von Barcode-Funktionen in einem XSLT Stylesheet.

```
<?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:altova="http://www.altova.com"
 xmlns:altovaext="http://www.altova.com/xslt-extensions"
 xmlns:altovaext-barcode="java: com. altova. extensions. barcode. BarcodeWrapper"

 xmlns:altovaext-barcode-property="
java: com. altova. extensions. barcode. BarcodePropertyWrapper">
 <xsl:output method="html" encoding="UTF-8" indent="yes"/>
 <xsl:template match="/">
 <html>
 <head><title/></head>
 <body>

 </body>
 </html>
 <xsl:result-document
 href="{ altovaext: get-temp-folder() } barcode. png"
 method="text" encoding="base64tobinary" >
 <xsl:variable name="barcodeObject"
 select="
altovaext-barcode: newInstance(' Code39' , string(' some
value') ,
 96, 0, (altovaext-barcode-property: new(' setModuleWidth' ,
25.4 div 96 * 2)))"/>
 <xsl:value-of select="
xs: base64Binary(xs: hexBinary(string(altovaext-barcode: generateBarcodePngAsHexS
tring($barcodeObject))))"/>
 </xsl:result-document>
 </xsl:template>
 </xsl:stylesheet>
```

## 21.2 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.

- [MS Access](#)
- [MS SQL Server](#)
- [MySQL](#)
- [Oracle](#)
- [ODBC](#)
- [ADO](#)
- [Sybase](#)

## MS Access

Bei Generierung eines XML-Schemas anhand einer MS Access-Datenbank (DB) werden die MS Access DB-Datentypen wie unten in der Tabelle gezeigt in XML-Schema-Datentypen konvertiert.

| MS Access-Datentyp   | XML-Schema-Datentyp |
|----------------------|---------------------|
| G U I D              | xsD                 |
| char                 | xstring             |
| varchar              | xstring             |
| m e m o              | xstring             |
| bt                   | xboolean            |
| Number(single)       | xfloat              |
| Number(double)       | xdouble             |
| Decimal              | xdecimal            |
| Currency             | xdecimal            |
| Date/Time            | xdateTime           |
| Number(Long Integer) | xinteger            |
| Number(Integer)      | xshort              |
| Number(Byte)         | xbyte               |
| O L E O b j e c t    | xbase64Binary       |

## MS SQL Server

Bei Generierung eines XML-Schemas anhand einer MS SQL-Datenbank (DB) werden die MS SQL Server DB-Datentypen wie unten in der Tabelle gezeigt in XML-Schema-Datentypen konvertiert.

| MS SQL Server-Datentyp | XML-Schema-Datentyp |
|------------------------|---------------------|
| uniqueidentifier       | xsD                 |
| char                   | xstring             |
| nchar                  | xstring             |
| varchar                | xstring             |
| nvarchar               | xstring             |
| text                   | xstring             |
| ntext                  | xstring             |
| sysname                | xstring             |
| bit                    | xsboolean           |
| real                   | xsfloat             |
| float                  | xsdouble            |
| decimal                | xsdecimal           |
| money                  | xsdecimal           |
| smallmoney             | xsdecimal           |
| datetime               | xsdateTime          |
| smalldatetime          | xsdateTime          |
| binary                 | xsbase64Binary      |
| varbinary              | xsbase64Binary      |
| image                  | xsbase64Binary      |
| integer                | xsinteger           |
| smallint               | xsshort             |
| bigint                 | xslong              |
| tinyint                | xsbyte              |



## MySQL

Bei Generierung eines XML-Schemas anhand einer MySQL-Datenbank (DB) werden die MySQL-Datentypen wie unten in der Tabelle gezeigt in XML-Schema-Datentypen konvertiert.

| My SQL-Datentyp | XML-Schema-Datentyp |
|-----------------|---------------------|
| char            | xstring             |
| varchar         | xstring             |
| text            | xstring             |
| tinytext        | xstring             |
| medium text     | xstring             |
| longtext        | xstring             |
| tinyint()       | xboolean            |
| float           | xfloat              |
| double          | xdouble             |
| decimal         | xdecimal            |
| datetime        | xdateTime           |
| blob            | xbase64Binary       |
| tinyblob        | xbase64Binary       |
| medium blob     | xbase64Binary       |
| longblob        | xbase64Binary       |
| smallint        | xshort              |
| bigint          | xlong               |
| tinyint         | xbyte               |

## 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 |
|-----------------------------------------|---------------------|
| ROW ID                                  | xsID                |
| CHAR                                    | xstring             |
| NCHAR                                   | xstring             |
| VARCHAR2                                | xstring             |
| NVARCHAR2                               | xstring             |
| CLOB                                    | xstring             |
| NCLOB                                   | xstring             |
| NUMBER (with check constraint applied)* | xs:boolean          |
| NUMBER                                  | xdecimal            |
| FLOAT                                   | xdouble             |
| DATE                                    | xs:dateTime         |
| INTERVAL YEAR TO MONTH                  | xs:gYearMonth       |
| BLOB                                    | xs:base64Binary     |

- \* 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 einem XML-Schema-Datentyp `xs:boolean` konvertiert. Dieser Mechanismus eignet sich zum Generieren eines Datentyps `xs:boolean` im generierten XML-Schema.

## ODBC

Bei Generierung eines XML-Schemas anhand einer ODBC-Datenbank (DB) werden die ODBC-Datentypen wie unten in der Tabelle gezeigt in XML-Schema-Datentypen konvertiert.

| ODBC-Datentyp     | XML-Schema-Datentyp |
|-------------------|---------------------|
| SQL_GUID          | xsd                 |
| SQL_CHAR          | xstring             |
| SQL_VARCHAR       | xstring             |
| SQL_LONGVARCHAR   | xstring             |
| SQL_BIT           | xsboolean           |
| SQL_REAL          | xsfloat             |
| SQL_DOUBLE        | xsdubbe             |
| SQL_DECIMAL       | xdecimal            |
| SQL_TIMESTAMP     | xsddateTime         |
| SQL_DATE          | xscate              |
| SQL_BINARY        | xsbase64Binary      |
| SQL_VARBINARY     | xsbase64Binary      |
| SQL_LONGVARBINARY | xsbase64Binary      |
| SQL_INTEGER       | xsiinteger          |
| SQL_SMALLINT      | xshot               |
| SQL_BIGINT        | xstring             |
| SQL_TINYINT       | xbyte               |

**ADO**

Bei Generierung eines XML-Schemas anhand einer ADO-Datenbank (DB) werden die ADO DB-Datentypen wie unten in der Tabelle gezeigt in XML-Schema-Datentypen konvertiert.

| <b>ADO-Datentyp</b>  | <b>XML-Schema-Datentyp</b> |
|----------------------|----------------------------|
| adG U I D            | xsD                        |
| adC har              | xstring                    |
| adW C har            | xstring                    |
| adVarC har           | xstring                    |
| adW VarC har         | xstring                    |
| adLongVarC har       | xstring                    |
| adW LongVarC har     | xstring                    |
| adVarW C har         | xstring                    |
| adBoolean            | xsboolean                  |
| adSingle             | xsfloat                    |
| adDouble             | xsdublex                   |
| adNum eric           | xdecimal                   |
| adC urrency          | xdecimal                   |
| adD B T in e S tam p | xsdatetime                 |
| adDate               | xdate                      |
| adBinary             | xsbase64Binary             |
| adVarBinary          | xsbase64Binary             |
| adLongVarBinary      | xsbase64Binary             |
| adInteger            | xsinteger                  |
| adUnsignedInt        | xsunsignedInt              |
| adSmallInt           | xshort                     |
| adUnsignedSmallInt   | xsunsignedShort            |
| adBigInt             | xlong                      |
| adUnsignedBigInt     | xsunsignedLong             |
| adTinyInt            | xsbyte                     |
| adUnsignedTinyInt    | xsunsignedByte             |

## 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                | xstring             |
| nchar               | xstring             |
| varchar             | xstring             |
| nvarchar            | xstring             |
| text                | xstring             |
| sysname-varchar(30) | xstring             |
| bit                 | xboolean            |
| real                | xfloat              |
| float               | xfloat              |
| double              | xdouble             |
| decimal             | xdecimal            |
| money               | xdecimal            |
| smallmoney          | xdecimal            |
| datetime            | xdateTime           |
| smalldatetime       | xdateTime           |
| timestamp           | xdateTime           |
| binary<=255         | xbase64Binary       |
| varbinary<=255      | xbase64Binary       |
| image               | xbase64Binary       |
| integer             | xinteger            |
| smallint            | xshort              |
| tinyint             | xbyte               |

## 21.3 Technische Daten

Dieses Kapitel enthält nützliche Hintergrundinformationen zu den technischen Aspekten Ihrer Software. Es ist in die folgenden Abschnitte gegliedert:

- [OS- und Arbeitsspeichieranforderungen](#)
- [Altova XML Parser](#)
- [Altova XSLT- und XQuery-Prozessor](#)
- [Unicode-Unterstützung](#)
- [Internet-Verwendung](#)

## OS- und Arbeitsspeichieranforderungen

### Betriebssystem

Die Altova-Software-Applikationen sind:

- 32-Bit Windows-Applikationen für Windows XP Windows Server 2003 und 2008, Windows Vista und Windows 7 oder
- 64-Bit-Applikationen für Windows Vista und Windows 7

### Arbeitsspeicher

Da die Software in C++ geschrieben wurde, wird dafür nicht so viel Platz wie in einer Java Runtime Umgebung benötigt und normalerweise wird dafür weniger Arbeitsspeicher als bei einer vergleichbaren Java-basierten Applikation benötigt.

Es ist notwendig, dass die einzelnen Dokumente in den Hauptarbeitsspeicher geladen werden, damit jedes Dokument zur Gänze geparkt und analysiert werden kann und die Anzeige- und Bearbeitungsgeschwindigkeit während der normalen Arbeit verbessert wird.

Auch die unbegrenzte Rückgängig-Funktion kann einiges an Arbeitsspeicher in Anspruch nehmen. Wenn Sie große Abschnitte in großen Dokumenten immer wieder ausschneiden und einfügen, kann dies enorm viel Speicherplatz verbrauchen.

**Altova XML Parser**

Wenn Sie ein XML-Dokument öffnen, verwendet die Applikation den integrierten Validierungsparser (den Altova XML Parser), um das Dokument auf Wohlgeformtheit zu prüfen, es gegen ein Schema zu validieren (falls eines angegeben wurde) und Baumstrukturen und Infosets zu erstellen. Der Altova XML Parser dient auch dazu, beim Editieren von Dokumenten intelligente Eingabehilfen zur Verfügung zu stellen und etwaige Validierungsfehler dynamisch anzuzeigen.

Im integrierten Altova XML Parser ist die Final Recommendation der W3C XML Schema Spezifikation implementiert. Neue Entwicklungen, die von der XML Schema-Arbeitsgruppe empfohlen werden, werden ständig in den Altova Parser integriert, sodass Ihnen mit Altova-Produkten immer eine Entwicklungsumgebung auf dem neuesten Stand der Technik zur Verfügung steht.



**Altova XSLT- und XQuery-Prozessor**

Die Altova Produkte arbeiten mit dem Altova XSLT 1.0-Prozessor, dem Altova XSLT 2.0-Prozessor und dem Altova XQuery-Prozessor. Nähere Informationen zu diesen Prozessoren finden Sie im Anhang 1 zu dieser Dokumentation im Abschnitt über Prozessoren.

Diese drei Prozessoren stehen auch als eigenständige Applikationen in Form des AltovaXML-Pakets zum kostenlosen Download von der [Altova Website](#) zur Verfügung. Die Dokumentation zur Verwendung der Prozessoren wird mit dem AltovaXML-Paket mitgeliefert.

## Unicode-Unterstützung

Unicode ist ein 16-Bit-Zeichensatzstandard (der auf 32-Bit erweiterbar ist), der vom [Unicode-Konsortium](#) definiert wurde und der eine eindeutige Zahl für jedes Zeichen zur Verfügung stellt, und zwar:

- plattformunabhängig
- programmunabhängig
- unabhängig von der Sprache

Im Prinzip arbeiten Computer nur mit Zahlen. Sie speichern Buchstaben und andere Zeichen unter einer Zahl, die sie diesen zuweisen. Vor der Entwicklung von Unicode gab es hunderte von verschiedenen Codiersystemen für die Zuweisung dieser Zahlen. Es gab keine einzige Codierung, die Platz für genügend Zeichen bot: So werden z.B. in der Europäischen Union mehrere verschiedene Codierungen benötigt, um alle ihre Sprachen abzudecken. Selbst für eine einzige Sprache wie z.B. Englisch reichte eine einzelne Codierung nicht aus, um alle gebräuchlichen Buchstaben, Satzzeichen und technischen Symbole abzudecken.

Diese Codierungssysteme standen früher miteinander in Konflikt, d.h. zwei Codierungen verwendeten dieselbe Zahl für zwei verschiedene Zeichen oder verschiedene Zahlen für dasselbe Zeichen. Jeder Computer (v.a. Server) muss viele verschiedene Codierungen unterstützen. Wenn aber Daten zwischen verschiedenen Codierungen oder Plattformen ausgetauscht werden, besteht immer das Risiko, dass Fehler entstehen.

### Unicode ist die Lösung für diese Probleme!

Unicode enthält eine eindeutige Zahl für jedes Zeichen, egal für welche Plattform, welches Programm oder welche Sprache. Der Unicode-Standard wurde von branchenführenden Unternehmen wie Apple, HP, IBM, JustSystems, Microsoft, Oracle, SAP, Sun, Base und vielen anderen übernommen.

Unicode ist für moderne Standards wie XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, usw. erforderlich und ist das offizielle Verfahren zum Implementieren von ISO/IEC 10646. Unicode wird von vielen Betriebssystemen, allen modernen Browsern und vielen anderen Produkten unterstützt. Die Entwicklung des Unicode-Standards und die Verfügbarkeit von Tools, die diesen unterstützen, gehören zu den wichtigsten globalen Software Technologie-Trends in letzter Zeit.

Die Integration von Unicode in Client-Server oder Multi-Tier-Applikationen und Websites bietet beträchtliche Kostenvorteile im Vergleich zur Verwendung herkömmlicher Zeichensätze. Dank Unicode kann ein einziges Software-Produkt oder eine einzige Website für mehrere Plattformen, Sprachen und Länder konzipiert werden, ohne dass es neu entwickelt werden muss. Daten können dadurch durch viele verschiedene Systeme übertragen werden, ohne dass dabei Fehler auftreten.

## Windows XP

Die XML-Produkte von Altova bieten vollständige Unicode-Unterstützung. Um ein XML-Dokument von einem System mit nicht lateinischer Zeichensatzcodierung bearbeiten zu können, benötigen Sie allerdings auch eine Schriftart, die die von diesem Dokument verwendeten Unicode-Zeichen unterstützt.

Beachten Sie bitte, dass die meisten Schriftarten nur eine bestimmte Untergruppe des gesamten Unicode-Bereichs enthalten und normalerweise für das entsprechende Schriftsystem ausgelegt sind. Aus diesem Grund kann es vorkommen, dass ein XML-Dokument "nicht druckbare" Zeichen enthält, da die gewählte Schriftart die erforderlichen Glyphen nicht enthält. Es ist daher manchmal nützlich, wenn man eine Schriftart hat, die den gesamten Unicode-Bereich abdeckt - v.a. wenn Sie XML-Dokumente aus der ganzen Welt editieren.

Die größte uns bekannte Universalschriftart ist "Arial Unicode MS", eine Schriftart, die von Agfa Monotype für Microsoft entwickelt wurde. Diese Schriftart enthält mehr als 50.000 Glyphen und deckt das gesamte im Unicode 2.1.-Standard definierte Zeichenspektrum ab. Die Schriftart ist in Microsoft Office 2000 enthalten und beansprucht 23 MB.

Wir empfehlen Ihnen dringend, diese Schriftart auf Ihrem System zu installieren und mit der Applikation zu verwenden, wenn Sie öfter Dokumente bearbeiten, die in unterschiedlichen Schriftsystemen verfasst wurden. Diese Schriftart wird bei Verwendung der Standardinstallation von Microsoft Office nicht installiert, Sie können jedoch die Option "Benutzerdefinierte Installation" wählen, um sie zu installieren.

Im Ordner "Examples" finden Sie außerdem eine neue XHTML-Datei mit dem Namen `Unicode-UTF8.html`, die den Satz "Wenn die Welt miteinander spricht, spricht sie Unicode" in vielen verschiedenen Sprachen und Schriftsystemen enthält ("When the world wants to talk, it speaks Unicode") (世界的に話すなら、Unicode です) - Diese Zeile stammt aus der 10. Unicode-Konferenz von 1997 und ist ein schönes Beispiel für die Bedeutung von Unicode für den XML-Standard. Wenn Sie diese Datei öffnen, erhalten Sie einen kurzen Eindruck davon, was mit Unicode möglich ist und welche Schriftsysteme von den auf Ihrem PC verfügbaren Schriftarten unterstützt werden.

**Rechts nach Links-Schreibrichtung**

Beachten Sie bitte, dass selbst unter Windows NT 4.0 Texte aus Schriftsystemen mit einer Schreibrichtung von rechts nach links (z.B. Hebräisch oder Arabisch) nur in den Ländern korrekt wiedergegeben werden, in denen solche Schriftsysteme tatsächlich verwendet werden. Dies liegt daran, dass nur die hebräische und arabische Version von Windows NT die Wiedergabe und Bearbeitung von Text in Rechts-links-Schreibweise auf Betriebssystemebene unterstützen.

## Internet-Verwendung

Altova-Applikationen können für Sie auch eine Verbindung mit dem Internet herstellen. Dies geschieht in den folgenden Fällen:

- Wenn Sie im Registrierungsdialogfeld (**Hilfe | Software-Aktivierung**) auf "Kostenlosen Evaluierungs-Key anfordern" klicken, werden die drei Felder im Registrierungsdialogfeld über eine normale HTTP-Verbindung (Port 80) an unseren Webserver übertragen. Sie erhalten dann per E-Mail (normales SMTP) den kostenlosen Evaluierungs-Keycode zugesandt.
- Falls Sie den URL-Modus des Dialogfelds "Öffnen" verwenden, um ein Dokument direkt über eine URL zu öffnen (**Datei | Öffnen | Zu URL wechseln**), wird dieses Dokument über eine HTTP-Verbindung (Port 80) aufgerufen. (*Diese Funktionalität steht in XMLSpy und Authentic Desktop zur Verfügung.*)
- Wenn Sie ein XML-Dokument öffnen, das sich auf ein XML-Schema oder eine DTD bezieht und das Dokument durch eine URL definiert wird, wird es ebenfalls über eine HTTP-Verbindung (Port 80) aufgerufen, sobald Sie das XML-Dokument validieren. Dies kann auch automatisch beim Öffnen eines Dokuments vorkommen, wenn Sie die Applikation so konfiguriert haben, dass Dateien beim Öffnen automatisch validiert werden (Register "Datei" des Dialogfelds **Extras | Optionen**).
- Wenn Sie den Befehl "Als Mail senden..." (**Datei | Als Mail senden**) verwenden, wird die aktuelle Auswahl bzw. Datei über ein MAPI-kompatibles Mail-Programm, das auf dem PC des Benutzers installiert ist, versendet
- Im Rahmen der in diesem Handbuch und der Altova-Software-Lizenzvereinbarung beschriebenen Software-Aktivierung und beim Live-Update..

## 21.4 Lizenzinformationen

Dieser Anhang enthält die folgenden Informationen:

- Informationen über den [Vertrieb dieses Software-Produkts](#)
- Informationen im Zusammenhang mit dem [intellektuellen Eigentum](#) dieses Software-Produkts
- Die [Endbenutzer-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.

## Electronic Software Distribution

Dieses Produkt ist über EDS (Electronic Software Distribution), also auf elektronischem Weg erhältlich, eine Methode, die die folgenden einzigartigen Vorteile bietet:

- Sie können die Software kostenlos testen, bevor Sie sich zu einem Kauf entscheiden.
- Wenn Sie sich entschieden haben, die Software zu kaufen, können Sie Ihre [Bestellung online abgeben](#) und das Produkt steht Ihnen in Minutenschnelle im vollen Lizenzumfang zur Verfügung.
- Sie erhalten immer die neueste Version unserer Software
- Die Software enthält ein umfassendes Online-Hilfesystem. Die neueste Version des Benutzerhandbuchs steht auf unserer Website [www.altova.com](http://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 weiterverwenden möchten, müssen Sie eine [Software-Produktlizenz](#) erwerben, die Sie in Form eines Keycodes erhalten, der in das Dialogfeld "Software-Aktivierung" eingegeben werden muss, um das Produkt zu aktivieren. Sie können Ihre Lizenz online in unserem Web-Shop unter [Altova Shop](#) erwerben.

### Weitergabe der Software an andere Mitarbeiter in Ihrem Unternehmen zu Testzwecken

Wenn Sie die Evaluierungsversion der Software auch anderen Personen in Ihrem Unternehmen über das Netzwerk zur Verfügung stellen möchten oder wenn Sie sie auf einem PC installieren möchten, der nicht mit dem Internet verbunden ist, sollten Sie nur das Installationsprogramm weitergeben - vorausgesetzt es wurde nicht modifiziert. Jeder, der das von Ihnen zur Verfügung gestellte Installationsprogramm aufruft, muss einen eigenen Evaluierungs-Keycode für 30 Tage anfordern. Nach Ablauf des Testzeitraums, muss eine Lizenz erworben werden, damit das Produkt weiter verwendet werden kann.

Nähere Informationen finden Sie in der [Altova Software-Lizenzvereinbarung](#) am Ende dieses Handbuchs.

## 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 Computernamen 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 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.

### Mehrfachlizenz

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 Website](http://www.iana.org/) (<http://www.iana.org/>)) 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.

Sie werden außerdem auch feststellen, dass Ihr Altova Produkt viele nützliche Funktionen enthält, die Ihre Internet-Verbindung verwenden, diese stehen jedoch nicht mit der Lizenzüberwachungsfunktion in Zusammenhang.



## Rechte am geistigen Eigentum

Die Software und alle Kopien, deren Anfertigung Ihnen von Altova gestattet ist, sind das geistige Eigentum von Altova und dessen Lieferanten und stehen in deren Besitz. Struktur, Organisation und Code der Software stellen wertvolle Betriebsgeheimnisse und vertrauliche Informationen von Altova und dessen Lieferanten dar. Die Software ist durch gesetzliche Bestimmungen urheberrechtlich geschützt. Diese gesetzlichen Bestimmungen beinhalten (ohne darauf beschränkt zu sein) das Urheberrecht der Vereinigten Staaten, internationale Verträge und das in den Ländern, in denen die Software genutzt wird, geltende Recht. Altova behält sich das Eigentum an allen Patenten, Urheberrechten, Branchengeheimnissen, Warenzeichen und sonstigen geistigen Eigentumsrechten, die hinsichtlich der Software bestehen, vorbehalten. Das Eigentumsrecht von Altova erstreckt sich auch auf alle Bilder, Fotografien, Animationen, Videos, Audioaufzeichnungen, Musikstücke, Texte und „Applets“, die Teil der Software sind, und alle dazugehörigen Unterlagen in gedruckter Form. Mitteilungen über geltend gemachte Verstöße gegen geistige Eigentumsrechte sind an den Copyright Agenten von Altova zu richten (nähere Angaben dazu finden Sie auf der Website von Altova).

Altova Software enthält Software-Komponenten Dritter, die ebenfalls urheberrechtlich geschützt sind. Unter anderem sind diese Urheberrechte ausführlich unter [http://www.altova.com/legal\\_3rdparty.html](http://www.altova.com/legal_3rdparty.html) aufgelistet.

Alle anderen Namen oder Warenzeichen sind das Eigentum ihrer jeweiligen Inhaber.

## Altova Endbenutzer-Lizenzvereinbarung

**DIES IST EIN RECHTSGÜLTIGES DOKUMENT -- BITTE BEWAHREN SIE ES SORGFÄLTIG AUF**

### ALTOVA® ENDBENUTZER-LIZENZVERTRAG

Lizenzgeber:  
Altova GmbH  
Rudolfsplatz 13a/9  
A-1010 Wien  
Österreich

#### **Wichtig – Bitte sorgfältig lesen. Benutzerinformation:**

Dieser Endbenutzer-Lizenzvertrag („Lizenzvertrag“) ist ein rechtsgültiger Vertrag, der zwischen Ihnen und Altova GmbH („Altova“) abgeschlossen wurde. Bevor Sie die von Altova bereitgestellte Software („Software“) sowie die dazugehörige Dokumentation (u.a. Unterlagen in gedruckter Form, Online-Dateien oder Dokumentation in elektronischer Form („Dokumentation“) verwenden, lesen Sie bitte dieses Dokument sorgfältig durch. Indem Sie auf die untenstehenden Schaltflächen „Ich akzeptiere“ und „Weiter“ klicken, oder indem Sie die Software installieren oder auf sonstige Weise nutzen, stimmen Sie zu, dass Sie die Bedingungen dieses Lizenzvertrags sowie die Datenschutzbestimmungen („Datenschutzbestimmungen“) von Altova, inklusive, jedoch nicht ausschließlich, die nachfolgend festgelegten Garantieausschlüsse, Haftungsbeschränkungen sowie Datennutzungs- und Kündigungsregelungen als rechtsverbindlich anerkennen, auch wenn Sie die Software nicht käuflich erwerben. Sie stimmen zu, dass diese Vereinbarung in demselben Umfang gültig und durchsetzbar ist, wie ein von Ihnen persönlich unterzeichneter Vertrag. Wenn Sie den Bedingungen dieses Lizenzvertrags nicht zustimmen, sind Sie nicht zur Verwendung der Software berechtigt und müssen alle heruntergeladenen Kopien, die sich in Ihrem Besitz oder unter Ihrer Kontrolle befinden, vernichten. Während des Installationsvorganges können Sie eine Kopie dieses Lizenzvertrages zum Zeitpunkt Ihrer Zustimmung ausdrucken. Darüber hinaus können Sie unsere Website unter <http://www.altova.com/de/eula> besuchen, um eine Kopie dieses Lizenzvertrags herunterzuladen und auszudrucken. Die Datenschutzbestimmungen können unter <http://www.altova.com/de/privacy> eingesehen werden.

## **1. SOFTWARELIZENZ**

### **(a) Lizenzgewährung.**

(i) Sofern Sie diesem Lizenzvertrag zugestimmt haben, gewährt Ihnen Altova eine nicht ausschließliche und nicht übertragbare (mit Ausnahme der nachfolgenden Bestimmungen) Lizenz, ohne das Recht zur Vergabe von Sublizenzen, zur Installation und Verwendung der Software auf einem kompatiblen Einzelplatzrechner oder einer Workstation, wobei die maximale Zahl an Computern, die im Lizenzumfang enthalten sind, nicht überschritten werden darf. Vorbehaltlich der Beschränkungen gemäß Abschnitt 1 (c), dürfen Sie eine Kopie der Software auf mehr als einem Ihrer kompatiblen Einzelplatzrechnern oder Workstations installieren und verwenden, wenn Sie eine Named-User-Lizenz (Lizenz zur benutzerdefinierten Nutzung) erworben haben. Vorbehaltlich der Beschränkungen gemäß den Abschnitten 1(d) und 1(e) darf die Software gleichzeitig im Netzwerk verwendet werden. Die maximale Zahl an Computern und/oder Benutzern sowie die Art der Lizenz, d.h. Einzelplatzlizenz, Named-User-Lizenz oder Parallellizenz, wird beim Kauf der Software festgelegt und genau angegeben. Einzelplatzlizenzen sind als fixe Lizenzen und nicht als Parallellizenzen zu verwenden, d.h. es ist nicht erlaubt, die Lizenz für einen Benutzer zu deinstallieren, um die Lizenz für einen anderen Benutzer zu installieren und diese Lizenz anschließend wieder zu deinstallieren und wieder für den ursprünglichen Benutzer zu

installieren. Einzelplatzlizenzen sollten statisch sein. Dessen ungeachtet sind permanente Uminstallationen einer Einzelplatzlizenz gestattet (wenn z.B. ein Mitarbeiter nicht mehr für das Unternehmen arbeitet und sein Rechner nicht mehr benötigt wird). Während des Evaluierungszeitraums (wie nachfolgend definiert) darf die Software nur von einem Benutzer und auf nur einem (1) Einzelplatzrechner oder einer Workstation installiert und genutzt werden. Wenn Sie die Lizenz im Rahmen eines Pakets von Altova-Softwareprodukten („Produktpaket“) erworben haben, von denen Sie nicht jedes Produkt einzeln installiert haben, gelten die Bestimmungen dieses Lizenzvertrags für die Verwendung der gesamten Software, die im Lieferumfang des Produktpakets enthalten ist.

(ii) Wenn Sie eine Lizenz für SchemaAgent erworben haben, dann gelten die Bestimmungen dieses Lizenzvertrags zur Nutzung der Software für den darin inkludierten SchemaAgent-Server („SchemaAgent-Server“). Dabei dürfen Sie den SchemaAgent-Server jedoch ausschließlich in Verbindung mit der Altova-Software und nur für die in der dazugehörigen Dokumentation beschriebenen Zwecke verwenden.

(iii) Wenn Sie eine Lizenz für Software erworben haben, die es dem Benutzer ermöglicht den Quellcode zu generieren, dann sind Sie durch die Ihnen im Rahmen dieser Lizenz erteilten Genehmigung zur Installation und Verwendung der Software auch berechtigt, den Quellcode zu generieren, und zwar entweder (i) anhand von Modulen der Altova Library, die in der Software inkludiert sind (der auf diese Weise generierte Code wird als „beschränkter Quellcode“ bezeichnet), oder (ii) anhand Ihrer eigenen Schemata oder Mappings (der anhand Ihres eigenen Quellmaterials generierte Code wird als „unbeschränkter Quellcode“ bezeichnet). Zusätzlich zu den durch diesen Lizenzvertrag eingeräumten Rechten gewährt Ihnen Altova eine nicht ausschließliche, nicht übertragbare und eingeschränkte Lizenz, den gesamten generierten Code (bestehend aus dem beschränkten und unbeschränkten Quellcode) in eine ausführbare Objektcodeform zu übersetzen und diese zu verwenden, zu kopieren, weiterzugeben oder zu lizenzieren. Sie sind nur dann befugt, den beschränkten Quellcode ab- oder weiterzugeben, weiterzulizenzieren, zu verkaufen oder an Dritte zu übertragen, wenn die betreffende Partei bereits in Form eines gesonderten Lizenzvertrags mit Altova zur Verwendung des beschränkten Quellcodes berechtigt ist. Ungeachtet anderslautender Bestimmungen in diesem Lizenzvertrag dürfen Sie die Altova-Library-Module oder den beschränkten Quellcode (oder sonstiges geistiges Eigentum von Altova enthalten in oder in Verbindung mit den Altova-Library-Modulen oder dem beschränkten Quellcode) nicht in solcher Art und Weise weitergeben, in andere Software einbetten, mit anderer Software verbinden oder auf andere Weise verwenden, die den beschränkten Quellcode den Bestimmungen des „Copylefts“, der freien Software oder einer Open Source Lizenz unterwerfen würde, welche die Offenlegung des beschränkten Quellcodes oder der Altova-Library-Module erforderlich machen würde. Altova behält sich alle übrigen Rechte an der Software vor. Bei den Bestandteilen von UModel, für die ein Reverse Engineering Ihres eigenen oder eines anderen Quellcodes, den Sie rechtmäßig erworben haben, zulässig ist, stellt eine solche Verwendung durch Sie keinen Verstoß gegen diesen Lizenzvertrag dar. Soweit dies nicht in Abschnitt 1(h) ausdrücklich gestattet wird, ist ein Reverse Engineering der Software explizit untersagt.

(iv) Falls beschränkter Quellcode in Form ausführbaren Objektcodes inkludiert wird, verpflichten Sie sich, folgende Informationen (1) in den Willkommensbildschirm, oder falls nicht vorhanden, in einen oder mehrere der durch den Endbenutzer jederzeit aufrufbaren Bildschirme und (2) - in elektronischer und/oder gedruckter Form in die Dokumentation zu inkludieren: "Teile dieses Programms wurden unter Verwendung von Altova® [Name der Altova Software, z.B. MapForce® 2011] entwickelt und enthalten Bibliotheken, die im Eigentum der Altova GmbH (Copyright © 2007-2011 Altova GmbH ([www.altova.com](http://www.altova.com))) stehen."

**(b) Verwendung auf einem Server zur Installation und Verwendung von SchemaAgent.** Sie sind berechtigt, ein (1) Exemplar der Software auf einem Dateiserver ausschließlich innerhalb Ihres internen Netzwerks zu installieren und von dort auf die maximale Zahl von betrieblich genutzten Computern innerhalb dieses Netzwerks, die in der Lizenz angegeben ist, herunterzuladen und zu installieren. Wenn Sie eine Lizenz für SchemaAgent

erworben haben, können Sie SchemaAgent auf jedem Server oder jeder Workstation installieren und in Verbindung mit Ihrer Software verwenden. Jede sonstige Netzwerknutzung ist unzulässig; dies umfasst insbesondere die direkte Verwendung oder die Nutzung der Software über Befehle, Daten oder Anweisungen von oder an einen Computer, der nicht Teil Ihres internen Netzwerks ist, die Verwendung für Internet- oder Webhosting-Dienste sowie die Verwendung durch einen Benutzer, der nicht unter einer gültigen Lizenz von Altova zur Verwendung dieses Exemplars der Software berechtigt ist.

**(c) Benutzerdefinierte Nutzung (Named-Use).** Wenn Sie eine Lizenz der „benutzerdefinierten Version“ der Software erworben haben, sind Sie berechtigt, die Software auf bis zu fünf (5) kompatiblen Einzelplatzrechnern oder Workstations zu installieren, auf denen Sie der primäre Nutzer sind. Dadurch können Sie von einem Computer zu einem anderen je nach Bedarf wechseln, vorausgesetzt, dass die Software von Ihnen als benutzerdefinierter Nutzer jeweils nur ein (1) Mal gleichzeitig verwendet wird. Wenn Sie mehrere benutzerdefinierte Lizenzen erworben haben, erhält jeder individuell definierte Nutzer einen gesonderten Lizenzschlüssel.

**(d) Parallellizenz für die Verwendung im selben physischen Netzwerk oder am selben Unternehmensstandort.** Wenn Sie eine Parallellizenz Software-Version erworben haben, können Sie die Software auf mehreren kompatiblen und betrieblich genutzten Computern installieren, jedoch höchstens zehnmal (10-mal) die maximale Zahl der Benutzer, vorausgesetzt, dass nur die maximale Zahl von Benutzern die Software wirklich gleichzeitig nutzen sowie weitere Voraussetzung, dass die Computer, auf welchen die Software installiert ist, in dasselbe physische Computer Netzwerk eingebunden sind. Die maximale Zahl der Parallelbenutzer wird beim Kauf der Software festgelegt. Sofern die Parallelbenutzer Version der Software an mehr als einem Standort oder in mehr als einem Netzwerk verwendet werden soll, sind für jedes physisch getrennte Netzwerk sowie für jeden physisch getrennten Standort separate Parallelbenutzer Lizenzen erforderlich, wobei die Einschränkungen hinsichtlich der maximalen Zahl der Benutzer und der Anzahl der Nutzer, die die Software benutzen, anwendbar sind. Für den Fall, dass ein Computer nicht in dasselbe physische Netzwerk eingebunden ist, ist für diesen eine lokal installierte Benutzerlizenz oder eine Parallellizenz für die Verwendung in einer virtuellen Umgebung erforderlich. Be- und Einschränkungen hinsichtlich der Nutzung von Parallelbenutzer Lizenzen auf Heimcomputern sind in Abschnitt 1(g) geregelt.

**(e) Parallellizenz für die Verwendung in einer virtuellen Umgebung.** Wenn Sie eine Parallellizenz erworben haben, sind Sie berechtigt, ein Exemplar der Software auf einem Terminalserver (Microsoft Terminal Server, Citrix Metaframe, usw.), einem Anwendungsvirtualisierungsserver (Microsoft App-V, Citrix XenApp, VMWare ThinApp, usw.) oder einer virtuellen Maschine in Ihrem Intranet zu installieren, damit einzelne Benutzer in Ihrer Organisation von einem anderen Computer über einen Terminalserver, einen Anwendungsvirtualisierungsserver oder eine virtuelle Maschine auf die Software zugreifen und diese nutzen können. Jedoch darf die Zahl der Benutzer, die über einen solchen Netzwerk- oder Terminalserver bzw. eine virtuelle Maschine zum selben Zeitpunkt auf die Software zugreifen, die maximale Zahl der im Lizenzumfang enthaltenen User nicht überschreiten. Außerdem darf die Gesamtanzahl der Benutzer, die die Software über den Terminal Server, den Anwendungsvirtualisierungsserver oder eine virtuelle Maschine verwenden, die sechs (6)-fache Anzahl der lizenzierten Benutzer nicht übersteigen. Folglich gelten die in Abschnitt 1(d) festgelegten Einschränkungen betreffend die Anzahl der Installationen und die Voraussetzung, dass die Verwendung über dasselbe physische Netzwerk erfolgen muss, nicht für Terminal Server-, Anwendungsvirtualisierungsserver-Umgebungen oder Umgebungen auf virtuellen Maschinen. Sie müssen in einer virtuellen Umgebung mit Hilfe geeigneter Maßnahmen sicherstellen, dass nicht mehr als die maximale Anzahl der in der Lizenz vorgesehenen Benutzer gleichzeitig auf die Software zugreifen. Altova gibt über die Performance der Altova-Software in einer Umgebung, in der Terminalserver, Anwendungsvirtualisierungserver oder virtuelle Maschinen verwendet werden, keinerlei Zusicherungen oder Gewährleistungen ab. Die Anwendung der Altova-Software auf die vorstehend beschriebene Weise ist

ausdrücklich von der eingeschränkten Garantie, die durch Abschnitt 5 dieses Lizenzvertrags eingeräumt wird, ausgeschlossen. Ebenso wird kein technischer Support bei Problemen geleistet, die aufgrund der Anwendung in einer solchen Umgebung auftreten.

**(f) Kopien für Sicherungs- und Archivierungszwecke.** (iii) Sie dürfen eine (1) Kopie der Software für Sicherungs- und eine Kopie für Archivierungszwecke herstellen, sofern diese nicht auf einem Computer installiert oder verwendet werden. Außerdem müssen bei Kopien zu Sicherungs- und Archivierungszwecken alle im Original enthaltenen urheber- und patentrechtlichen Angaben sowie alle anderen Angaben hinsichtlich geistiger Eigentumsrechte von Ihnen reproduziert werden und in der jeweiligen Kopie der Software unverändert enthalten sein. Die Befugnis zum Anlegen einer Sicherungs- und Archivierungskopie darf nur an Dritte übertragen werden, wenn gleichzeitig eine Übertragung aller Rechte an der Software gemäß Abschnitt 3 erfolgt.

**(g) Nutzung auf dem Privat-PC (privater und nicht-kommerzieller Gebrauch).** Um sich mit der Software, ihren Features und Funktion besser vertraut zu machen, sind Sie als vorrangiger Benutzer des Computers, auf dem die Software installiert wurde, berechtigt, ein weiteres Exemplar der Software zum persönlichen und nicht kommerziellen Gebrauch („pnk“) auf einem (1) einzigen Ihrer Heimcomputer (z.B. Ihrem Laptop oder PC) zu installieren. Dieses pnk-Exemplar darf zu keinerlei kommerziellen oder umsatzgenerierenden Geschäftsaktivitäten verwendet werden. Dazu gehören insbesondere die Arbeit von zu Hause aus, jedwede Form von Telearbeit oder andere Arbeit, die im Zusammenhang mit der Verwendung der Software steht. Das pnk-Exemplar der Software darf nicht gleichzeitig auf auf einem Privat-PC und dem eigentlich dafür vorgesehenen Primärcomputer verwendet werden.

**(h) Freischaltcodes, Upgrades und Updates.** Bevor Sie die Software kaufen bzw. im Rahmen Ihrer Registrierung für den Evaluierungszeitraum von dreißig (30) Tagen erhalten Sie einen Evaluierungs-Freischaltcode. Wenn Sie sich in der Folge dazu entscheiden, die Software von Altova GmbH oder einem autorisierten Vertriebs Händler zu kaufen, erhalten Sie einen uneingeschränkten Freischaltcode. Mit diesem können Sie die Software nach dem Evaluierungszeitraum aktivieren. Ohne ausdrückliche schriftliche Genehmigung von Altova dürfen Sie an Freischaltcodes keine Lizenzen einräumen und Freischaltcodes nicht reproduzieren oder vertreiben. Wenn die von Ihnen in Lizenz erworbene Software ein Upgrade oder Update ist, ersetzt das Upgrade oder Update das sie zuletzt heruntergeladen oder installiert haben, die zuvor lizenzierte Kopie der Software. Durch das betreffende Upgrade oder Update und den dazugehörigen Freischaltcode wird keine zweite Lizenz für die Software gewährt, und Sie dürfen die Kopie des Upgrades oder Updates nicht zusätzlich zu der Kopie der Software verwenden, die dadurch ersetzt wird und dessen Lizenz beendet wurde.

**(i) Eigentumsrecht.** Das Eigentumsrecht an der Software wird Ihnen nicht übertragen. Alle Exemplare der Software und alle von Ihnen angefertigte Kopien verbleiben – vorbehaltlich der Ihnen durch diesen Lizenzvertrag eingeräumten Rechte – im Eigentum von Altova. Dokumente, Dateien, Stylesheets, generierter Programmcode (dies beinhaltet auch den unbeschränkten Quellcode) und Schemata, die Sie unter Verwendung der Software anlegen oder erstellen, sind in Übereinstimmung mit der dazugehörigen Dokumentation und den Bestimmungen dieses Lizenzvertrags Ihr Eigentum außer diese wurden mit einer Testsoftware gemäß Abschnitt 4 dieser Vereinbarung erstellt. In diesem Fall wird Ihnen nur eine beschränkte Lizenz zur Nutzung sämtlicher Ergebnisse, welche einen generierten Programmcode (inklusive eines unbeschränkten Quellcodes) enthalten, wie Java, C++, C#, VB.NET oder XSLT und verbundene Projektdaten und Build Scripts sowie generierte XML, XML Schematas, Dokumentationen, UML Diagrammen und Datenbankstrukturen, für die 30 Tage dauernde Testperiode eingeräumt.

**(j) Reverse Engineering.** Sofern dies nicht ausdrücklich nach geltendem Recht der Europäischen Union gestattet ist, sind Sie nicht berechtigt, an der Software Reverse Engineering durchzuführen, die Software zu dekompile, zu disassemblieren, oder auf andere Weise zu versuchen, den Quellcode, zugrunde liegende Konzepte, Techniken für

Benutzerschnittstellen oder Algorithmen der Software direkt oder indirekt zu entschlüsseln. Davon ausgenommen ist lediglich das ausdrücklich nach geltendem Recht der Europäischen Union gestattete Maß an Dekompilierung, das erforderlich ist, um die Funktionsfähigkeit der Software mit anderen Softwareprogrammen zu erzielen. Dies ist Ihnen jedoch erst dann gestattet, wenn Sie zuerst bei Altova die dafür notwendigen Informationen angefordert haben und Altova Ihnen diese nicht zur Verfügung gestellt hat. Altova ist berechtigt, für die Bereitstellung solcher Informationen angemessene Bedingungen aufzustellen und eine angemessene Entschädigung zu verlangen. Die von Altova zur Verfügung gestellten Informationen bzw. alle von Ihnen im Sinne der hier festgelegten Bestimmungen rechtmäßig erworbenen Informationen dürfen nur zu den hierin angeführten Zwecken verwendet und keinesfalls an Dritte weitergegeben oder zur Entwicklung von Software genutzt werden, die der Software, die Gegenstand dieses Lizenzvertrags ist, in wesentlichen Aspekten ähnlich ist. Informationen im Sinne dieses Absatzes können von Benutzern aus der Europäischen Union bei der Kundendienstabteilung von Altova angefordert werden.

**(k) Sonstige Beschränkungen.** Sie sind nicht berechtigt, die Software (zur Gänze oder teilweise) zu verleihen, zu vermieten, per Leasing zur Verfügung zu stellen, weiterzulizenzieren, weiterzugeben oder auf sonstige Weise Dritten zu überlassen, es sei denn, dies ist durch die Bestimmungen in Abschnitt 3 oder an anderer Stelle in diesem Lizenzvertrag ausdrücklich gestattet. Sofern dies nicht ausdrücklich in diesem Vertrag gestattet ist, dürfen Sie die Software nicht kopieren. Alle Kopien, zu deren Anfertigung Sie berechtigt sind, müssen die im Original enthaltenen urheber- und patentrechtlichen Angaben sowie alle anderen Angaben hinsichtlich geistiger Eigentumsrechte unverändert enthalten. Die Software darf nicht verändert, adaptiert oder übersetzt werden. Sie dürfen weder direkt noch indirekt ein Pfand- oder Sicherungsrecht an der Software bestellen oder die Bestellung eines solchen zulassen. Es ist Ihnen nicht gestattet, wesentlich eine Handlung vorzunehmen, durch die die Software öffentlich zugänglich wird, oder die Software in einer Computerumgebung zu verwenden, die nicht in diesem Lizenzvertrag angegeben ist.

Sie haben die geltenden gesetzlichen Bestimmungen und die Anweisungen von Altova in Bezug auf die Benutzung der Software einzuhalten. Sie sind verpflichtet, Ihre Mitarbeiter und Vertreter, die Zugriff auf die Software haben, von den in diesem Lizenzvertrag enthaltenen Beschränkungen in Kenntnis zu setzen und diese auf sie zu überbinden.

**(l) ES WIRD KEINE GARANTIE ODER GEWÄHRLEISTUNG FÜR DIE FEHLERFREIHEIT DER SOFTWARE ABGEGEBEN ODER GEWÄHRT UND VON ALTOVA KEINERLEI HAFTUNG DIESBEZÜGLICH ÜBERNOMMEN. UNABHÄNGIG VON IRGENDWELCHEN SUPPORTLEISTUNGEN NACH IRGEND EINEM TECHNISCHEN STANDARD IST DIE SOFTWARE KEINESFALLS FÜR DIE VERWENDUNG IN ODER IM ZUSAMMENHANG MIT KERNKRAFTANLAGEN, FLUGGERÄTENAVIGATION, KOMMUNIKATIONSSYSTEMEN ODER LUFTVERKEHRSKONTROLLEINRICHTUNGEN, MEDIZINISCHEN GERÄTEN ODER LEBENSERHALTUNGSSYSTEMEN, MEDIZINISCHEN ODER DIE GESUNDHEITSVORSORGE BETREFFENDEN ANWENDUNGEN, ODER SONSTIGEN ANWENDUNGEN KONZIPIERT, BEI DENEN FEHLER DER SOFTWARE ODER FEHLER BEI DER DATENVERARBEITUNG ZU TODESFÄLLEN, PERSONENSCHÄDEN ODER SCHWEREN SACH- ODER UMWELTSCHÄDEN FÜHREN KÖNNTEN. SIE ERKLÄREN SICH DAMIT EINVERSTANDEN, DASS SIE DIE ALLEINIGE VERANTWORTUNG FÜR DIE EIGNUNG UND ZWECKMÄSSIGKEIT DER SOFTWARE UND ALLE MIT DER SOFTWARE ERSTELLTEN ODER VERARBEITETEN DATEN FÜR DEN BEABSICHTIGTEN VERWENDUNGSZWECK TRAGEN UND SIE WERDEN ALTOVA UND DEREN GESCHÄFTSFÜHRER SOWIE ANGESTELLTE HINSICHTLICH ALLER ANSPRÜCHE, FORDERUNGEN UND KLAGEN DRITTER, DIE AUF DIE EIGNUNG UND ZWECKMÄSSIGKEIT DER VON IHNEN VERWENDETEN SOFTWARE ODER DIE VON DER SOFTWARE ERSTELLTEN DATEN GEGRÜNDET SIND, SCHAD- UND KLAGLOS HALTEN.**

## 2. GEISTIGE EIGENTUMSRECHTE

**Anerkennung der Rechte von Altova.** Die Software und alle Kopien, deren Anfertigung Ihnen von Altova gestattet ist, sind das geistige Eigentum von Altova und dessen Lieferanten und stehen in deren Besitz. Struktur, Organisation und Code der Software stellen wertvolle Betriebsgeheimnisse und vertrauliche Informationen von Altova und dessen Lieferanten dar. Die Software ist durch gesetzliche Bestimmungen urheberrechtlich geschützt. Diese gesetzlichen Bestimmungen beinhalten (ohne darauf beschränkt zu sein) das Urheberrecht der Vereinigten Staaten, internationale Verträge und das in den Ländern, in denen die Software genutzt wird, geltende Recht. Sie anerkennen, dass sich Altova das Eigentum an allen Patenten, Urheberrechten, Branchengeheimnissen, Warenzeichen und sonstigen geistigen Eigentumsrechten, die hinsichtlich der Software bestehen, vorbehält. Das Eigentumsrecht von Altova erstreckt sich auch auf alle Bilder, Fotografien, Animationen, Videos, Audioaufzeichnungen, Musikstücke, Texte und „Applets“, die Teil der Software sind, und alle dazugehörigen Unterlagen in gedruckter Form. Sie dürfen keine Handlungen vornehmen, die sich nachteilig auf die geistigen Eigentumsrechte von Altova an der Software auswirken. Warenzeichen dürfen nur in Übereinstimmung mit den anerkannten Standards für die Verwendung von Warenzeichen (einschließlich der namentlichen Nennung der Warenzeicheninhaber) verwendet werden. Die Verwendung von Warenzeichen ist nur zur Kennzeichnung von Druckmaterialien, die mit der Software hergestellt wurden, gestattet. Es entstehen Ihnen daraus keinerlei Eigentumsrechte an dem betreffenden Warenzeichen. XMLSpy®, Authentic®, StyleVision®, MapForce®, UModel®, DatabaseSpy®, DiffDog®, SchemaAgent®, SemanticWorks®, MissionKit®, Markup Your Mind®, Nanonull™ und Altova® sind (in mehreren Ländern eingetragene) Warenzeichen von Altova GmbH. Unicode und das Logo von Unicode sind Warenzeichen von Unicode, Inc. Windows, Windows XP, Windows Vista und Windows 7 sind Warenzeichen von Microsoft. W3C, CSS, DOM, MathML, RDF, XHTML, XML und XSL sind (in mehreren Ländern eingetragene) Warenzeichen des World Wide Web Consortium (W3C). Marken des W3C sind von den Dachinstitutionen des Konsortiums (MIT, INRIA und Keio) eingetragen und stehen in deren Eigentum. Sofern dies nicht ausdrücklich hierin festgelegt ist, entstehen Ihnen aus diesem Lizenzvertrag keinerlei geistigen Eigentumsrechte an der Software. Mitteilungen über geltend gemachte Verstöße gegen geistige Eigentumsrechte sind an den Copyright Agent von Altova zu richten (nähere Angaben dazu finden Sie auf der Website von Altova).

## 3. ÜBERTRAGUNGSBESCHRÄNKUNG

Ungeachtet der vorstehenden Bestimmungen dürfen Sie alle Ihre Rechte zur Verwendung der Software an eine andere natürliche oder juristische Person unter der Voraussetzung übertragen, dass: (a) Sie den vorliegenden Lizenzvertrag, die Software und sonstige Soft- oder Hardware, die mit der Software, die Gegenstand dieses Lizenzvertrags ist, geliefert, verpackt oder auf dieser vorinstalliert ist, einschließlich aller Kopien, Updates und früherer Versionen sowie aller Exemplare von Fontsoftware, die in andere Formate konvertiert wurde, an die betreffende Person übertragen; (b) Sie keine Kopien, einschließlich Sicherungskopien und sonstiger, auf einem Computer gespeicherter Kopien, zurückbehalten; (c) der Empfänger von Altova einen persönlichen Freischaltcode von Altova erhält; und (d) der Empfänger die Bestimmungen dieses Lizenzvertrags sowie alle anderen Bestimmungen akzeptiert, zu denen Sie die Softwarelizenz rechtmäßig erworben haben. Ungeachtet der vorstehenden Ausführungen dürfen Sie keine Schulungs-, Vorab- oder Musterkopien der Software an Dritte übertragen.

## 4. ZUSÄTZLICHE BEDINGUNGEN FÜR VORAB- UND TESTVERSIONEN

Wenn es sich bei dem von Ihnen mit dieser Lizenz erworbenen Produkt um eine unverkäufliche Vorabversion oder um Beta-Software („Pre-Release-Software“) handelt, gelten die in diesem Abschnitt enthaltenen Bedingungen. Diese gelten außerdem für alle Test- und/oder Demoversionen von Altova-Software („Testsoftware“) und bleiben so lange gültig, bis Sie eine Lizenz erwerben. Wenn Bestimmungen dieses Abschnitts im Widerspruch zu anderen, in

diesem Lizenzvertrag enthaltenen Bestimmungen stehen, dann gehen die Bedingungen dieses Abschnitts hinsichtlich Pre-Release-Software und Testsoftware allen anderen Bestimmungen dieses Lizenzvertrags vor, soweit dies für die Auflösung des Widerspruchs erforderlich ist. Sie erkennen an, dass es sich bei der Pre-Release-Software um eine Vorabversion handelt, die nicht das endgültige Produkt von Altova darstellt, und in der Fehler, Funktionsstörungen oder andere Probleme auftreten können, die möglicherweise zu einem System- oder Hardwareabsturz oder -fehler bzw. zu Datenverlust führen. DIE PRE-RELEASE-SOFTWARE UND/ODER TESTSOFTWARE WIRD IHNEN DAHER **OHNE GEWÄHR FÜR DEREN VERWENDUNG ODER LEISTUNGSMERKMALE** ZUR VERFÜGUNG GESTELLT, UND ALTOVA ÜBERNIMMT DIESBEZÜGLICH KEINERLEI AUSDRÜCKLICHEN ODER STILLSCHWEIGENDEN GEWÄHRLEISTUNGS- ODER HAFTUNGSVERPFLICHTUNGEN. SOFERN UND SOWEIT FÜR PRE-RELEASE- UND/ODER TESTSOFTWARE KEIN HAFTUNGSSAUSCHLUSS, SONDERN NUR EINE HAFTUNGSBESCHRÄNKUNG GESETZLICH ZULÄSSIG IST, BESCHRÄNKT SICH DIE HAFTUNG VON ALTOVA UND DESSEN LIEFERANTEN AUF INSGESAMT FÜNFZIG US-DOLLAR (50 USD). Wenn die Testversion mit einer Time-out-Funktion ausgestattet ist, kann sie nach Ablauf des Evaluierungszeitraums nicht mehr verwendet werden. Mit dem Ablauf dieses Zeitraums endet Ihre Lizenz automatisch, wenn Sie sie nicht verlängern. Ihre Lizenz zur Nutzung sämtlicher mit der Testsoftware erstellten Ergebnisse, welche einen generierten Programmcode (inklusive den unbeschränkten Quellcode) wie Java, C++, C#, VB.NET oder XSLT und verbundene Projektdaten und Build Scripts sowie generierte XML, XML Schematas, Dokumentationen, UML Diagramme und Datenbankstrukturen enthalten, erlischt automatisch nach Ablauf der vorgesehenen Testperiode. Die Lizenz zum Gebrauch dieser Ergebnisse lebt nach dem Kauf einer Lizenz für die Software, die Sie getestet und mit der Sie das Ergebnis erstellt haben, wieder auf. Der Zugriff auf Dateien, die mit der Testsoftware erstellt wurden, erfolgt auf Ihr alleiniges Risiko. Sie bestätigen, dass Altova Ihnen weder zugesichert noch garantiert hat, die Pre-Release-Software zu einem späteren Zeitpunkt anderen Personen anzukündigen oder zur Verfügung zu stellen. Sie bestätigen ferner, dass Altova sich Ihnen gegenüber weder ausdrücklich noch stillschweigend verpflichtet hat, die Pre-Release-Software öffentlich anzukündigen oder einzuführen, und dass Altova kein Produkt anbieten muss, das der Pre-Release-Software ähnlich oder mit ihr kompatibel ist. Sie bestätigen folglich, dass jegliche Forschungs- oder Entwicklungsarbeit, die Sie in Bezug auf die Pre-Release-Software oder damit verbundene Produkte ausführen, in Ihr alleiniges Risiko fällt. Wenn Altova Sie dazu auffordert, werden Sie für die Dauer dieses Lizenzvertrags Altova Rückmeldung bezüglich Ihrer Tests und der Pre-Release-Software geben, einschließlich entsprechender Berichte über Fehler und Funktionsstörungen. Wenn Sie die Pre-Release-Software im Rahmen einer gesonderten schriftlichen Vereinbarung erhalten haben, unterliegt Ihre Verwendung der Software den darin enthaltenen Bedingungen. Es ist Ihnen nicht gestattet, die Pre-Release-Software unterzulizenzieren, per Leasing zur Verfügung zu stellen, zu verleihen, zu vermieten, weiterzugeben oder auf sonstige Weise Dritten zu überlassen. Sie verpflichten sich, bei Erhalt einer späteren unveröffentlichten Version der Pre-Release-Software bzw. bei Erhalt einer kommerziellen Vollversion von Altova (entweder als Einzelprodukt oder als Teil eines Gesamtprodukts) die zuvor von Altova erhaltene Pre-Release-Software zurückzugeben oder zu vernichten und den Bestimmungen des Lizenzvertrags für die betreffende spätere Version der Pre-Release-Software Folge zu leisten.

## 5. EINGESCHRÄNKTE GARANTIE UND HAFTUNGSBESCHRÄNKUNG

(a) **Eingeschränkte Garantie und Garantieansprüche der Kunden.** Altova garantiert der Person/Rechtspersönlichkeit, die ursprünglich die Lizenz für die Verwendung der Software gemäß den Bestimmungen dieses Lizenzvertrags erworben hat, dass (i) die Software über einen Zeitraum von neunzig (90) Tagen nach Erhalt im Wesentlichen in Übereinstimmung mit der dazugehörigen Dokumentation funktioniert, und (ii) die von Altova zur Verfügung gestellten Supportleistungen im Wesentlichen auf die in Abschnitt 6 dieses Vertrags niedergelegte Weise erfolgen. In einigen Ländern sind Beschränkungen über die Dauer einer stillschweigenden Garantie nicht zulässig, so dass die obigen Beschränkungen und Ausschlüsse eventuell für Sie nicht zutreffen. Im größtmöglichen rechtlich zulässigen Maß sind stillschweigende Garantien in



Bezug auf die Software (sofern solche existieren) auf neunzig (90) Tage beschränkt. Die gesamte Haftung von Altova und dessen Lieferanten, sowie Ihre einzigen Garantieansprüche, sind nach dem Ermessen von Altova auf eine der beiden folgenden Optionen beschränkt: (i) Erstattung des Kaufpreises (wenn zutreffend), oder (ii) Reparatur oder Austausch der Software, die unter die eingeschränkte Garantie von Altova fällt und die unter Vorlage einer Kopie des Kaufbelegs bei Altova reklamiert wird. Diese eingeschränkte Garantie gilt nicht, wenn die Funktionalität der Software durch ein Versehen, durch Missbrauch, falsche Anwendung, Trojaner, Viren oder einen sonstigen schädlichen externen Code beeinträchtigt wurde. Die Garantie für jede Ersatzsoftware erstreckt sich auf die Restdauer des ursprünglichen Garantiezeitraums oder auf dreißig (30) Tage, je nachdem, welcher Zeitraum länger ist. Die eingeschränkte Garantie gilt nicht für Test- und/oder Pre-Release-Software.

**(b) Keine weiteren Garantien, Haftungsausschluss.** MIT AUSNAHME DER VORSTEHEND ANGEFÜHRTEN EINGESCHRÄNKTEN GARANTIE UND DEN DIESBEZÜGLICHEN GARANTIEANSPRÜCHEN BESTEHEN SEITENS ALTOVA ODER DESSEN LIEFERANTEN KEINERLEI WEITEREN GARANTIEVERPFLICHTUNGEN. ALTOVA UND DESSEN LIEFERANTEN ÜBERNEHMEN KEINE GEWÄHR FÜR DIE ANWENDUNG ODER DIE ERGEBNISSE AUS DER NUTZUNG DER SOFTWARE. MIT AUSNAHME DER VORSTEHEND ANGEFÜHRTEN EINGESCHRÄNKTEN GARANTIE SOWIE IM HINBLICK AUF ALLE ANDEREN GEWÄHRLEISTUNGEN, BEDINGUNGEN, ZUSICHERUNGEN ODER ANSPRÜCHE, DIE NACH DER FÜR IHR LAND GELTENDEN RECHTSORDNUNG NICHT AUSGESCHLOSSEN ODER EINGESCHRÄNKT WERDEN KÖNNEN, SCHLIESSEN ALTOVA UND DESSEN LIEFERANTEN ALLE ANDEREN AUSDRÜCKLICHEN ODER STILLSCHWEIGENDEN GEWÄHRLEISTUNGEN, BEDINGUNGEN, ZUSICHERUNGEN UND ANSPRÜCHE AUS, DIE SICH AUS DEM GESETZ, DER RECHTSPRAXIS, EINEM GEWOHNHEITSRECHT, EINEM HANDELSBRAUCH ODER AUS SONSTIGEN GRÜNDEN ERGEBEN. ALTOVA UND DESSEN LIEFERANTEN SCHLIESSEN IM GRÖSSTMÖGLICHEN RECHTLICH ZULÄSSIGEN UMFANG ALLE ANDEREN AUSDRÜCKLICHEN UND STILLSCHWEIGENDEN GARANTIEEN UND GEWÄHRLEISTUNGEN AUS. DIES BEINHÄLTET UNTER ANDEREM GARANTIEEN IM HINBLICK AUF MARKTGÄNGIGKEIT, EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, ZUFRIEDENSTELLENDEN QUALITÄT, INFORMATIONSGEHALT ODER -RICHTIGKEIT, UNGESTÖRTE NUTZUNG, EIGENTUMSRECHT UND NICHTVERLETZUNG VON RECHTEN DRITTER UND DIE ERBRINGUNG ODER NICHTERBRINGUNG VON SUPPORTLEISTUNGEN IN BEZUG AUF DIE SOFTWARE. MIT DIESER EINGESCHRÄNKTEN GARANTIE WERDEN IHNEN BESTIMMTE RECHTE EINGERÄUMT. UNTER UMSTÄNDEN BESITZEN SIE NOCH ANDERE RECHTE; DIE JA NACH LAND/RECHTSORDNUNG UNTERSCHIEDLICH SEIN KÖNNEN.

**(c) Haftungseinschränkung.** SOWEIT DIE ANWENDBAREN GESETZE DIES ZULASSEN, HAFTEN ALTOVA ODER DESSEN LIEFERANTEN, AUCH WENN EINE IM RAHMEN DER GARANTIE DURCHGEFÜHRTE ABHILFEMASSNAHME IHREN WESENTLICHEN ZWECK NICHT ERFÜLLT, IN KEINEM FALL FÜR KONKRETE, ZUFÄLLIG ENTSTANDENE, MITTELBARE, UNMITTELBARE ODER FOLGESCHÄDEN JEGLICHER ART (INSBESONDERE SCHÄDEN AUS ENTGANGENEM GEWINN, GESCHÄFTSUNTERBRECHUNGEN, VERLUST VON GESCHÄFTSINFORMATIONEN ODER ANDEREN FINANZIELLEN VERLUSTEN), DIE DURCH DIE NUTZUNG ODER DIE UNMÖGLICHKEIT DER NUTZUNG DER SOFTWARE ODER DIE NICHTERBRINGUNG VON SUPPORTLEISTUNGEN ENTSTANDEN SIND, SELBST WENN ALTOVA AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WURDE. IN JEDEM FALL IST DIE GESAMTHAFTUNG VON ALTOVA GEMÄSS DIESEM LIZENZVERTRAG AUF DIE HÖHE DES FÜR DAS SOFTWAREPRODUKT ENTRICHTETEN BETRAGES BESCHRÄNKT. Da es in einigen Ländern und Rechtsordnungen nicht zulässig ist, die Haftung für Schäden auszuschließen oder zu beschränken, gilt die obige Einschränkung unter Umständen nicht für Sie. In solchen Ländern und Rechtsordnungen gilt die Beschränkung der Haftung durch Altova im größtmöglichen rechtlich zulässigen Umfang, und der Ausschluss bzw. die Beschränkung der in diesem Lizenzvertrag beinhalteten Garantien und Haftungen hat keine Auswirkung auf

die Konsumentenschutzrechte von Personen, die Waren auf andere Weise als im Zuge ihrer geschäftlichen Tätigkeit erwerben. Der vorstehende Haftungssausschluss/die vorstehende Haftungsbeschränkung sind wesentliche Bestandteile des zwischen Ihnen und Altova abgeschlossenen Lizenzvertrags.

**(d) Ansprüche in Zusammenhang mit Urheberrechtsverletzungen.** Altova wird Sie gegenüber allen Forderungen, Prozessen oder Verfahren schad- und klaglos halten bzw. alle Ansprüche, Prozesse oder Verfahren beilegen, die Dritte mit dem Argument gegen Sie erheben, dass der Inhalt der Software gegen ein Urheberrecht verstößt oder ein geistiges oder sonstiges Eigentumsrecht verletzt, das durch das Recht der Vereinigten Staaten oder der Europäischen Union geschützt ist (insgesamt als „Ansprüche“ bezeichnet). Dies erfolgt – soweit nicht ausdrücklich etwas Anderes festgelegt ist – jedoch nur insoweit, als der betreffende Anspruch sich direkt aus der Verwendung der Software ergibt und nach Maßgabe der in Abschnitt 5 dieses Vertrags festgelegten Beschränkungen. Altova ist von jedem Anspruch innerhalb von zehn (10) Geschäftstagen, nachdem Sie erstmals davon benachrichtigt wurden, in Kenntnis zu setzen. Außerdem haben Sie mit Altova in angemessenem Umfang bei der Abwehr solcher Ansprüche zu kooperieren und Altova dabei zu unterstützen, ohne dass Sie dafür Kosten geltend machen können. Das Recht auf Entscheidungen in Bezug auf solche Ansprüche liegt allein bei Altova (dies beinhaltet auch, ohne darauf beschränkt zu sein, die Auswahl der Rechtsberater und das Recht, für Sie einen Vergleich zu den von Altova für zweckmäßig erachteten Bedingungen einzugehen). Sie können auf eigene Kosten einen Rechtsberater hinzuziehen und an den Verfahrens- oder Vergleichsverhandlungen teilnehmen. Altova kommt bis zu einer Höhe von insgesamt maximal dem Kaufpreis der Software für die Schäden, Kosten und Anwaltsgebühren auf, zu deren Bezahlung Sie in Zusammenhang mit solchen Ansprüchen verpflichtet werden (oder die Sie aufgrund eines Vergleichs zu entrichten haben), soweit diese nicht von einer Versicherung oder dritten Partei übernommen werden. Ist oder wird die Software Gegenstand von aufgrund einer Urheberrechtsverletzung vorgebrachten Ansprüchen, oder wird ihre Verwendung untersagt, oder ist es nach Ansicht des Rechtsberaters von Altova wahrscheinlich, dass ein solcher Umstand eintritt, so wird Altova versuchen, eine Beilegung herbeizuführen, indem alle wirtschaftlich vertretbaren Anstrengungen unternommen werden, um die Software zu modifizieren oder eine Lizenz für die weitere Verwendung der Software zu erwerben. Wenn es nach Ansicht des Rechtsberaters von Altova nicht möglich ist, den bevorstehenden oder bereits vorgebrachten Anspruch bzw. die Verfügung, mit der die Verwendung der Software untersagt wurde, durch angemessene Abänderung oder den Erwerb einer Lizenz beizulegen, so kann Altova diesen Lizenzvertrag ohne negative Konsequenzen für Altova beenden und Ihnen anteilig alle bereits an Altova entrichteten Gebühren zurückerstatten. MIT AUSNAHME DER VORSTEHEND ANGEFÜHRTEN BESTIMMUNGEN OBLIEGEN ALTOVA KEINERLEI HAFTUNGSVERPFLICHTUNGEN FÜR ANSPRÜCHE IN ZUSAMMENHANG MIT URHEBERRECHTSVERLETZUNGEN. Diese Haftungsverpflichtung gilt nicht für Urheberrechtsverletzungen, die ausschließlich auf vom Kunden eingebrachte Elemente zurückzuführen sind.

## 6. SUPPORT UND WARTUNG

Zusätzlich zu der Version des Softwareprodukts, für das Sie eine Lizenz erhalten haben, bietet Altova eine Reihe von optionalen „Support- & Wartungspaketen“ („*Software & Maintenance Package*“, „SMP“) an, die Sie zusätzlich zu Ihrer Lizenz für die Software erwerben können. Der für das jeweilige SMP geltende Supportzeitraum (wie nachfolgend definiert) wird beim Kauf des SMP festgelegt. Die Support- und Wartungsservices und die Upgrades, die Ihnen zur Verfügung stehen, hängen davon ab, ob Sie ein SMP erwerben bzw. für welche Version des SMP Sie sich entscheiden.

(a) Wenn Sie kein SMP erwerben, erhalten Sie beim Kauf lediglich die Software, danach jedoch keinerlei Wartungsreleases oder Updates. Obwohl die Möglichkeit besteht, dass Altova Ihnen in Einzelfällen Wartungsreleases als kostenlose Zusatzleistung zur Verfügung stellt, sind in diesen Releases keine neuen Produktfeatures, die über das beim Kauf der Software bestehende Maß hinausgehen, beinhaltet. In jedem Fall erhalten Sie dreißig (30) Tage lang ab

Kauf der Software (der „Supportzeitraum“ im Sinne dieses Absatzes 6(a)) kostenlosen technischen Support von Altova. Außerdem kann Altova in Einzelfällen auch während des dreißig (30) Tage dauernden Evaluierungszeitraums technischen Support als kostenlose Zusatzleistung zur Verfügung stellen. Technischen Support erhalten Sie ausschließlich über ein webbasiertes Supportformular, wobei es keine garantierte Reaktionszeit gibt.

(b) Mit dem Erwerb eines SMP haben Sie während des dafür geltenden Supportzeitraums **Anspruch auf die von Ihnen erworbene Produktversion** sowie auf alle Wartungsreleases und Updates für diese Produktversion, die während des für Sie geltenden Supportzeitraums freigegeben werden. Während des Supportzeitraums Ihres SMP erhalten Sie auch Upgrades auf die entsprechende Produktversion der Software, mit der auf eine höhere Version der Software, für die Sie die Lizenz erworben haben, gewechselt wird und die während Ihres Supportzeitraums freigegeben werden. Die einzelnen Upgradeeditionen, auf die Sie innerhalb Ihres Supportzeitraums Anspruch haben, sind in dem von Ihnen erworbenen SMP im Detail angeführt. Software, die als gesondertes Produkt eingeführt wird, ist nicht im SMP enthalten. Wartungsreleases, Updates und Upgrades können neue Features enthalten, dies muss aber nicht der Fall sein. Darüber hinaus erhalten Sie während des Supportzeitraums bevorzugten technischen Support von Altova, und zwar ausschließlich über ein webbasiertes Supportformular. Altova wird alle wirtschaftlich vertretbaren Anstrengungen unternehmen, um alle Anfragen per E-Mail innerhalb von achtundvierzig (48) Stunden während der Geschäftszeiten (Montag bis Freitag, 8.00 bis 22.00 Uhr UTC, ausgenommen Feiertage in Österreich und den USA) zu beantworten. Außerdem wird Altova sich in angemessenem Umfang darum bemühen, Workarounds für Fehler, die bei der Software aufgetreten sind, zur Verfügung zu stellen.

Während des Supportzeitraums können Sie Altova Fehler oder Defekte in der Software melden. Wenn nach Altovas Einschätzung ein reproduzierbarer schwerwiegender Fehler vorliegt, der die Verwendung und Funktionalität der Software erheblich beeinträchtigt, wird Altova wirtschaftlich vertretbaren Anstrengungen unternehmen um Korrekturen oder provisorische Lösungen anzubieten, die in zukünftigen Updates oder Wartungsreleases enthalten sind. Diese Updates oder Wartungsreleases werden von Altova von Zeit zu Zeit zur Verfügung gestellt.

Es steht im Ermessen von Altova, einen schriftlichen Nachweis über von Ihnen festgestellte Fehler oder Funktionsstörungen zu verlangen oder Beispieldateien anzufordern, aus denen das aufgetretene Softwareproblem hervorgeht. In einem solchen Fall müssen Sie die angeforderten Nachweise oder Dateien, aus denen ausreichend detailliert hervorgeht, in welchen Aspekten Fehler bei der Software auftreten, so schnell wie möglich per E-Mail, Fax oder Expresspost mit Zustellung am nächsten Tag an Altova übermitteln. Bei der Diagnose oder Analyse von Fehlern haben Sie in zumutbarem Rahmen mit Altova zu kooperieren. Fehlerbehebungen können in Wartungsreleases, Updates oder neuen Hauptversionen der Software enthalten sein. Altova ist nicht verpflichtet, unwesentliche Fehler, d.h. Fehler, die die Benutzung der Software nach dem Ermessen von Altova nicht wesentlich beeinträchtigen, zu beheben. Sie erhalten von Altova unter keinen Umständen Beratung, Unterstützung oder Hilfestellung, die allgemeiner Natur ist und nicht in unmittelbarem Zusammenhang mit der Software steht.

Bei der Aktualisierung der Software kann es unter Umständen vorkommen, dass vor der Installation auch Software aktualisiert werden muss, die nicht unter diesen Lizenzvertrag fällt. Für Updates von Betriebssystem- und Anwendungssoftware, die nicht ausdrücklich Gegenstand dieses Lizenzvertrags ist, sind ausschließlich Sie verantwortlich. Solche Updates sind nicht im Umfang dieser Lizenz enthalten und werden nicht von Altova zur Verfügung gestellt. Die Erfüllung der in diesem Abschnitt 6 festgelegten Verpflichtungen durch Altova versteht sich unter der Bedingung, dass Sie die Software ordnungsgemäß verwenden und diesen Lizenzvertrag ausnahmslos jederzeit einhalten. Altova ist nicht verpflichtet, technischen Support zu leisten, wenn nach Ansicht von Altova die Fehlfunktion der Software auf einen der folgenden Gründe zurückzuführen ist: (i) Fehler, die durch die Verlegung der Software auf einen anderen Rechner oder Speicherort hervorgerufen werden; (ii) Änderungen, Modifikationen oder Versuche, die Software abzuwandeln, die von Altova nicht schriftlich

genehmigt wurden; (iii) äußere Einflüsse auf die Software, wie z.B. Naturkatastrophen, Stromausfälle, Stromschwankungen oder Computerausfälle; (iv) Ihr Versäumnis die Software auf dem von Altova festgelegten Release Level zu halten; oder (v) Nutzung der Software ohne vorherige Genehmigung durch Altova zusammen mit einer anderen Software. Sie alleine sind dafür verantwortlich: (i) allen Betriebs- und Fehlerbehebungsanleitungen von Altova Folge zu leisten, Altova unverzüglich von Fehlern oder Defekten an der Software zu informieren und Altova eine genaue Beschreibung dieser Fehler und/oder Defekte zu liefern; (ii) für den Schutz Ihrer vertraulichen Informationen zu sorgen; (iii) Datensicherungssysteme und -abläufe für die Wiederherstellung verlorener oder geänderter Dateien, Daten oder Programme einzurichten und anzuwenden.

## 7. AKTIVIERUNG DER SOFTWARE, UPDATES UND LIZENZÜBERWACHUNG

**(a) Lizenzüberwachung.** Altova verfügt über ein integriertes Lizenzüberwachungsmodul, das Ihnen hilft, unbeabsichtigte Verletzungen dieses Lizenzvertrags zu vermeiden. Dabei kann Altova zur Lizenzüberwachung von verschiedenen installierten Versionen der Software Ihr internes Netzwerk verwenden.

**(b) Aktivierung der Software.** Die Software von Altova kann Ihr internes Netzwerk und Ihre Internetverbindung verwenden, um Angaben über Ihre Lizenz im Zuge der Installation, Registrierung, Benutzung oder Aktualisierung an einen Altova-Lizenzserver zu übertragen und diese zu verifizieren, um auf diese Weise eine nicht lizenzierte oder illegale Verwendung der Software zu verhindern und den Kundenservice von Altova weiter zu verbessern. Die Aktivierung erfolgt über einen Austausch von lizenzbezogenen Daten zwischen Ihrem Computer und dem Altova-Lizenzserver. Sie stimmen dieser Vorgangsweise von Altova zu und verpflichten sich, allen diesbezüglichen Vorgaben Folge zu leisten. Sie erklären sich damit einverstanden, dass die Verwendung von Freischaltcodes, welche nicht von Altova erstellt werden oder wurden und nicht rechtmäßig von Altova oder einem dazu berechtigten Wiederverkäufer erworben wurden, zum Zweck der Aktivierung oder Nutzung der Software die gewerblichen Schutzrechte von Altova sowie die Bestimmungen dieses Lizenzvertrages verletzt. Sie erklären sich weiters damit einverstanden, dass Versuche mit dem Zweck der Umgehung oder Deaktivierung der urheberrechtlichen Schutzmaßnahmen oder Lizenzmanagementsystemen von Altova die gewerblichen Schutzrechte von Altova sowie die Bestimmungen dieses Lizenzvertrages verletzen. Altova behält sich ausdrücklich das Recht vor, alle rechtlich verfügbaren sowie angemessenen Mittel zu ergreifen, um derartige Praktiken zu verhindern und entgangenen Gewinn, Schäden und Kosten zurückerstattet zu bekommen.

**(c) LiveUpdate.** Altova stellt Ihnen ein neues LiveUpdate Benachrichtigungsservice zur Verfügung, welches kostenlos ist. Altova kann Ihr internes Netzwerk und Ihre Internetverbindung verwenden, um Angaben über Ihre Lizenz an einen LiveUpdate-Server von Altova zu übertragen, um Ihre Lizenz in gewissen Zeitabständen zu verifizieren und festzustellen, ob ein Update für Sie verfügbar ist.

**(d) Verwendung von Daten.** Der gesamte Wortlaut der Datenschutzbestimmungen von Altova kann unter <http://www.altova.com/de/privacy> eingesehen werden. Die Datenschutzbestimmungen sind durch Bezugnahme Teil dieses Lizenzvertrags. Durch Ihre Zustimmung zu den Bestimmungen dieses Lizenzvertrags bzw. durch Benutzung der Software erklären Sie sich damit einverstanden, dass Altova für die in diesem Lizenzvertrag und/oder in den Datenschutzbestimmungen (in der jeweils geltenden Fassung) genannten Zwecke Daten erhebt, verarbeitet und weitergibt. Altova behält sich das Recht vor, diese Bestimmung des Lizenzvertrags und/oder der Datenschutzbestimmungen jederzeit zu ändern. Wir legen Ihnen nahe, die auf der Website von Altova veröffentlichten Datenschutzbestimmungen von Zeit zu Zeit erneut durchzulesen.

(e) **Hinweis für Benutzer in Europa.** Bitte beachten Sie, dass die in Absatz 7(d) beschriebenen Informationen von Altova, Inc., einem Unternehmen mit Sitz in Beverly, Massachusetts, USA, oder seinen Tochterunternehmen, Zweigniederlassungen oder weltweit ansässigen autorisierten Partnern zum Zweck der Datenverarbeitung, Analyse und Überprüfung nach außerhalb der EU transferiert werden können. Sie werden darauf hingewiesen, dass in den USA ein Datenschutzmodell zur Anwendung kommt, das teils auf Gesetzen, teils auf Regierungsverordnungen und zum Teil auf Selbstregulierung beruht. Des Weiteren werden Sie darauf hingewiesen, dass der Rat der Europäischen Union befunden hat, dass dieses amerikanische Modell den in Artikel 25 der Datenrichtlinie der Europäischen Union (Richtlinie 95/46/EC, 1995 O.J. (L 281) 31) festgelegten Datenschutzbestimmungen nicht "in ausreichendem Ausmaß" Rechnung trägt. Laut Artikel 26 der Datenrichtlinie der Europäischen Union, dürfen persönliche Daten dann von der Europäischen Union in ein Drittland übertragen werden, wenn die jeweilige Person ihre Zustimmung zur Übertragung derartiger Informationen eindeutig gegeben hat, unabhängig davon, in welchem Ausmaß diese Daten in anderen Ländern geschützt sind. Durch Ihre Zustimmung zu dieser Software-Lizenzvereinbarung gestatten Sie die Übertragung aller derartiger Informationen an die USA sowie deren Verarbeitung gemäß dieser Software-Lizenzvereinbarung und Altovas Datenschutzbestimmungen.

## 8. DAUER UND BEENDIGUNG

Dieser Lizenzvertrag wird folgendermaßen beendet: (a) indem Sie Altova eine schriftliche Benachrichtigung von der Vertragsbeendigung übermitteln; (b) durch Altova in Form einer diesbezüglichen schriftlichen Benachrichtigung, wenn Sie gegen diesen Lizenzvertrag verstoßen und diesen Verstoß nicht innerhalb von 10 (zehn) Tagen, nachdem Altova Ihnen dies mitgeteilt hat, berichtigen; oder (c) auf Verlangen eines autorisierten Vertriebshändlers von Altova, wenn Sie eine Lizenzzahlung nicht entrichten oder sonstige Zahlungsverpflichtungen nicht einhalten. Der Lizenzvertrag einer früheren Version der Software, für die Sie ein Upgrade oder ein Update durchgeführt haben, wird mit Ihrer Zustimmung zu dem Lizenzvertrag, der für das Upgrade oder Update gilt, automatisch beendet. Bei Beendigung des Lizenzvertrags dürfen Sie die gesamte Software, die Gegenstand dieser Lizenz ist, nicht mehr verwenden, müssen alle Kopien, die in Ihrem Besitz oder Einflussbereich stehen, vernichten, und müssen in zumutbarem Rahmen alle von Altova geforderten Maßnahmen ergreifen, um sicherzustellen, dass keine Kopien der Software in Ihrem Besitz oder Einflussbereich verbleiben. Die in den Abschnitten 1(h), 1(i), 1(j), 1(k), 2, 5(b), 5(c), 5(d), 7(d), 7(e), 9, 10 und 11 niedergelegten Bestimmungen bleiben auch nach Beendigung dieses Lizenzvertrags weiterhin aufrecht.

## 9. EINGESCHRÄNKTE RECHTE UND EXPORTBESCHRÄNKUNGEN

Die Entwicklung der Software wurde ausschließlich privat finanziert. Bei der Software handelt es sich um kommerzielle Computersoftware, die mit **eingeschränkten Rechten** ausgestattet ist. Die Verwendung, Vervielfältigung oder Weitergabe der Software durch die US-Regierung, eine Behörde oder einen Kooperationspartner der US-Regierung unterliegt den Beschränkungen im Rahmen dieses Vertrags sowie den Beschränkungen nach FAR 12.211 und 12.212 (48 C.F.R. §12.211 und 12.212) bzw. DFARS 227.7202 (48 C.F.R. §227-7202). Dabei wird eine Lizenz für kommerzielle Computersoftware und kommerzielle Computerdokumentation an die US-Regierung als Endnutzer ausschließlich in Form einer Lizenz für kommerziell genutzte Güter erteilt, weshalb es sich bei den damit verbundenen Rechten um dieselben Rechte handelt, die allen anderen Endnutzern im Rahmen dieses Lizenzvertrags gewährt werden. Beim Hersteller handelt es sich um Altova GmbH, Rudolfsplatz, 13a/9, A-1010 Wien, Österreich/EU. Die Software und/oder Dokumentation darf ausschließlich unter Einhaltung aller anwendbaren Exportbestimmungen der Vereinigten Staaten von Amerika sowie des Landes, in dem Sie die Software erhalten haben, verwendet, exportiert oder wiedereingeführt werden. Insbesondere darf die Software und/oder Dokumentation nicht (i) in ein Land exportiert oder wiedereingeführt werden, über das die Vereinigten Staaten ein Embargo verhängt haben, oder einem Staatsangehörigen oder Bewohner eines solchen Landes überlassen werden; oder (ii) einer Person überlassen werden, die auf der Liste der Specially Designated Nationals des U.S.

Treasury Department oder dem Table of Denial Orders des U.S. Department of Commerce verzeichnet sind. Indem Sie die Software benutzen, erklären Sie, dass Sie weder in einem dieser Länder ansässig sind noch seiner Kontrolle unterliegen, noch ein Staatsangehöriger oder Bewohner eines dieser Länder sind, noch auf einer der vorstehend erwähnten Listen genannt werden.

## 10. SOFTWARE DRITTER

Die dieser Lizenz unterliegende Software kann Software Dritter enthalten, für die ergänzende Vermerke und/oder Nutzungsbedingungen gelten. Diese Vermerke und ergänzenden Nutzungsbedingungen für die Software Dritter können über unsere Website unter [http://www.altova.com/legal\\_3rdparty.html](http://www.altova.com/legal_3rdparty.html) eingesehen werden und sind durch Bezugnahme Teil dieses Lizenzvertrags. Indem Sie Ihre Zustimmung zu den Bedingungen dieses Vertrags erteilen, stimmen Sie auch automatisch allen ergänzenden Bestimmungen, die möglicherweise darin enthalten sind, zu.

## 11. ALLGEMEINE BESTIMMUNGEN

Wenn Sie sich in der Europäischen Union befinden und die Software in der Europäischen Union und nicht in den Vereinigten Staaten verwenden, unterliegt dieser Lizenzvertrag dem Recht der Republik Österreich (unter Ausschluss von dessen Verweisungsnormen und der UN-Kaufrechtskonvention). Sie erklären sich ausdrücklich damit einverstanden, dass alle Streitigkeiten oder Konflikte mit Altova, die in Zusammenhang mit Ihrer Verwendung der Software stehen, in die alleinige Zuständigkeit des Handelsgerichts Wien fallen. Sie erklären sich weiters ausdrücklich damit einverstanden, dass Sie bezüglich solcher Streitigkeiten oder Konflikte der persönlichen Zuständigkeit des Handelsgerichts Wien unterstellt sind.

Wenn Sie sich in den Vereinigten Staaten befinden und die Software in den Vereinigten Staaten verwenden, unterliegt dieser Lizenzvertrag dem Recht des Commonwealth of Massachusetts, USA (unter Ausschluss von dessen Verweisungsnormen und der UN-Kaufrechtskonvention). Sie erklären sich ausdrücklich damit einverstanden, dass alle Streitigkeiten oder Konflikte mit Altova, die in Zusammenhang mit Ihrer Verwendung der Software stehen, in die alleinige Zuständigkeit der einzel- und bundesstaatlichen Gerichte im Bundesstaat Massachusetts fallen. Sie erklären sich weiters ausdrücklich damit einverstanden, dass Sie bezüglich solcher Streitigkeiten oder Konflikte der persönlichen Zuständigkeit der einzel- und bundesstaatlichen Gerichte im Bundesstaat Massachusetts unterstellt sind.

Wenn Sie sich nicht in der Europäischen Union oder den Vereinigten Staaten befinden und die Software nicht in den Vereinigten Staaten verwenden, unterliegt dieser Lizenzvertrag dem Recht der Republik Österreich (unter Ausschluss von dessen Verweisungsnormen und der UN-Kaufrechtskonvention). Sie erklären sich ausdrücklich damit einverstanden, dass alle Streitigkeiten oder Konflikte mit Altova, die in Zusammenhang mit Ihrer Verwendung der Software stehen, in die alleinige Zuständigkeit des Handelsgerichts Wien fallen. Sie erklären sich weiters ausdrücklich damit einverstanden, dass Sie bezüglich solcher Streitigkeiten oder Konflikte der persönlichen Zuständigkeit des Handelsgerichts Wien unterstellt sind. Die Anwendung von Verweisungsnormen einer Rechtsordnung sowie des UN-Abkommens zum internationalen Warenkauf (CISG) auf diesen Lizenzvertrag wird ausdrücklich ausgeschlossen.

Dieser Lizenzvertrag enthält die gesamte Vereinbarung zwischen den Parteien in Bezug auf den Vertragsgegenstand und tritt an die Stelle aller diesbezüglichen früheren mündlichen oder schriftlichen Vereinbarungen zwischen den Parteien. Benachrichtigungen oder sonstige Mitteilungen im Rahmen dieses Lizenzvertrags müssen schriftlich erfolgen und gelten als ordnungsgemäß übermittelt, wenn sie per Einschreiben mit Rückschein oder per Kurierdienst mit Zustellung am nächsten Tag an die auf der Website von Altova angegebene Adresse (wenn Altova der Empfänger ist) bzw. an die in den Aufzeichnungen von Altova für Sie eingetragene Adresse (wenn Sie der Empfänger sind) oder aber an eine andere, von den Vertragspartnern festgelegte und auf die vorstehend beschriebene Weise bekannt gegebene Adresse geschickt

werden. Dieser Lizenzvertrag ist für die Vertragspartner verbindlich und geht auf ihre Erben, persönlichen und rechtlichen Vertreter, verbundenen Unternehmen, Rechtsnachfolger und zulässigen Abtretungsempfänger über. Die Nichtdurchsetzung oder Nichtausübung von Rechten oder Rechtsmitteln unter diesem Lizenzvertrag durch die Vertragspartner stellt keinen Verzicht auf ein solches Recht oder Rechtsmittel dar und beeinträchtigt in keiner Weise das Recht des betreffenden Vertragspartners, ein solches Recht oder Rechtsmittel sowie alle anderen Rechte oder Rechtsmittel aus diesem Lizenzvertrag später durchzusetzen bzw. auszuüben. Eine Änderung dieses Lizenzvertrags ist nur Form einer schriftlich niedergelegten Vereinbarung möglich, die von beiden Vertragspartnern unterzeichnet wird. Wenn eine Zuwiderhandlung gegen die Bestimmungen dieses Lizenzvertrags durch einen der Vertragspartner erfolgt ist oder droht, so kann der jeweils andere Vertragspartner alle ihm zustehenden Rechte und Rechtsmittel geltend machen. Jeder Vertragspartner ist ordnungsgemäß befugt und ermächtigt, in diesen Lizenzvertrag einzutreten und die daraus erwachsenden Verpflichtungen zu erfüllen. Sollte eine Bestimmung dieses Lizenzvertrags für unwirksam, rechtswidrig oder undurchführbar erklärt werden, so wird dadurch die Wirksamkeit der übrigen Bestimmungen nicht berührt, und dieser Lizenzvertrag bleibt im größtmöglichen rechtlich zulässigen Umfang wirksam und gültig. Die Vertragspartner haben die vorstehenden Vertragsbedingungen zur Kenntnis genommen und erklären sich ausdrücklich damit einverstanden.

Letzte Aktualisierung: 2011-10-01





# Index

■

**.docx (nur Enterprise Edition), 22, 38**

**.NET,**

Integration von StyleVision, 654

Unterschiede zu StyleVision Standalone, 656

**.NET Erweiterungsfunktionen,**

Datentypkonvertierungen, .NET in XPath/XQuery, 1154

Datentypkonvertierungen, XPath/XQuery in .NET, 1153

für XSLT und XQuery, 1149

Instanzmethoden, Instanzfelder, 1152

Konstruktoren, 1151

statische Methoden, statische Felder, 1151

Übersicht, 1149

**.NET-Erweiterungsfunktionen,**

in XPath-Ausdrücken, 691

Unterstützung in der Authentic-Ansicht, 691

## A

**Abdocken,**

Design-Eingabehilfenfenster, 39

**Abfrage,**

siehe Datenbank abfragen, 552

**Abfrage (Schaltfläche),**

in SPS einfügen, 546

**Abfragefenster,**

im Fenster Datenbankabfrage, 560

**Abfragen,**

für DB-Anzeige in der Authentic-Ansicht, 597

**Abkürzungen,**

im Benutzerhandbuch, 6

**Absatz,**

Befehl für das Einfügen im Design, 749

einschließen in, 775

**Abschnitte,**

Dokumentanfangsabschnitt, 412

einfügen, 769

im SPS-Design, 410

löschen, 410

Standardeigenschaften für neue Abschnitte, 412

und Seitenlayout, 410

**ADO,**

Konvertierung von Datentypen in XML-Schema

Generierung von DB, 1170

**Aktive Datei zu Projekt hinzufügen, 732**

**Aktive Konfiguration, 823**

**Aktive und verwandte Dateien zu Projekt hinzufügen, 732**

**Aktualisieren eines Node (Enterprise und Professional Edition),**

mit dem Ergebnis einer automatischen Berechnung, 273

**Alias,**

siehe Globale Ressourcen, 427

**Allgemeine Arbeitsabläufe, 106**

**Alternative Verarbeitung, 446**

**Altova Erweiterungen,**

Diagrammfunktionen (siehe Diagrammfunktionen), 1158

**Altova Erweiterungsfunktionen,**

allgemeine Funktionen, 1158

Diagrammfunktionen (siehe Diagrammfunktionen), 1158

**Altova Globale Ressourcen,**

siehe Globale Ressourcen, 427

**Altova Ressource zu Projekt hinzufügen, 730**

**Altova Website, 837**

**Altova XML Parser,**

Info, 1174

**Altova XSLT 1.0-Prozessor,**

Einschränkungen und implementierungsspezifisches

Verhalten, 1126

**Altova XSLT 2.0-Prozessor,**

allgemeine Informationen, 1129

Informationen, 1128

**Altova-Prozessoren,**

in Altova-Produkten, 1175

**AltovaRowStatus,**

in DB-basierten SPS, 546

**AltovaXML, 626**

und FOP, 636, 637

**Ambiguity,**

of content model, 133

**Andocken,**

Design-Eingabehilfenfenster, 39

**Anhänge, 1124**

**Anhängen,**

Spalte an Tabelle im SPS anhängen, 786

Zeile in Tabelle im SPS anhängen, 786

**Anmerkungen zur Programmierung,**

Skripting-Projekt, 865

**Anpassen von StyleVision, 824****Ansicht wechseln,**

zur Authentic-Ansicht, 571

**Ansichten,**

Layout auf der Benutzeroberfläche, 39

**API,**

aufrufen, 1101  
Dokumentation, 874  
JAVA, 1003  
JAVA Classpath, 1003

**Application,**

ActiveDocument, 904  
Application, 904  
Documents, 904  
Parent, 906  
Quit, 906

**Applikationsebene,**

Integration von StyleVision, 1052

**Applikations-Events, 860****Arabisch, 1178****Arbeitsspeicher-Anforderungen, 1173****Arial Unicode MS, 1177****Atomisierung von Nodes,**

in XPath 2.0- und XQuery 1.0-Auswertung, 1133

**Ausblenden,**

Funktion der Design-Eingabehilfenfenster, 39

**Ausdrücke,**

in DB-Filterkriterien, 540

**Ausgabe-basierte Bedingungen, 291****Ausgabedateien,**

generieren, 115  
über die Befehlszeile generieren, 628  
von DB-basierten StyleVision Power Stylesheet, 550

**Ausgabekodierung, 715****Ausgabevorschauen,**

Beschreibung, 38

**Auswahlliste,**

im Schnellstart-Tutorial, 92

**Auswahllisten, 185****Authentic,**

Daten in einer XML-Datenbak bearbeiten, 804  
mit neuer Version von StyleVision synchronisieren, 114

**Authentic Browser, 29****Authentic Desktop, 29****Authentic Node-Eigenschaften, 444****Authentic Skripterstellung, 612, 615**

Makros, 616

**Authentic toolbar, 796****Authentic XML, 568****Authentic-Ansicht,**

als XML/Text einfügen, 576  
Bearbeitungen speichern, 711  
Benutzeroberfläche, 569  
Beschreibung, 36  
Dokumentanzeige, 573  
Dokumenterstellung, 112  
Hauptfenster, 573  
in Altova-Produkten, 29  
Kontextmenüs, 576  
Markup-Anzeige, 573  
SPS-Tabellen, 586  
Symbolleisten-Schaltfläche, 676  
Symbolleistenschaltflächen, 571  
Tabellen (SPS und XML), 585  
Text formatieren, 571  
Übersicht über die GUI, 570  
und SPS, 112  
und Standard-Industrieschemas, 112  
und XML-Arbeitsdatei, 36  
Verwendung von XML-Tabellen, 588  
Verwendung wichtiger Features, 579  
wechseln zur, 571  
XML-Tabellen, 588  
XML-Tabellensymbole, 592

**Authentic-Ansicht-Events, 860****Authentic-Menü,**

Anzeige von Markup-Symbolen, 571  
Bearbeiten dynamischer Tabellen, 571

**AuthenticRange,**

AppendRow, 923  
Application, 924  
CanPerformAction, 924  
CanPerformActionWith, 924  
Close, 925  
CollapsToBegin, 925  
CollapsToEnd, 925  
Copy, 926  
Cut, 926  
Delete, 926  
DeleteRow, 927  
DuplicateRow, 927  
ExpandTo, 928  
FirstTextPosition, 928  
FirstXMLData, 929  
FirstXMLDataOffset, 930  
GetElementAttributeNames, 931

**AuthenticRange,**

- GetElementAttributeValue, 931
- GetElementHierarchy, 932
- GetEntityNames, 933
- Goto, 933
- GotoNext, 934
- GotoNextCursorPosition, 934
- GotoPrevious, 935
- GotoPreviousCursorPosition, 935
- HasElementAttribute, 936
- InsertEntity, 936
- InsertRow, 937
- IsCopyEnabled, 937
- IsCutEnabled, 937
- IsDeleteEnabled, 938
- IsEmpty, 938
- IsEqual, 938
- IsFirstRow, 938
- IsInDynamicTable, 939
- IsLastRow, 939
- IsPasteEnabled, 939
- IsTextStateApplied, 940
- LastTextPosition, 940
- LastXMLData, 941
- LastXMLDataOffset, 941
- MoveBegin, 942
- MoveEnd, 943
- MoveRowDown, 943
- MoveRowUp, 943
- Parent, 944
- Paste, 944
- PerformAction, 944
- Select, 945
- SelectNext, 946
- SelectPrevious, 946
- SetElementAttributeValue, 947
- SetFromRange, 948
- Text, 949

**AuthenticView, 966**

- Application, 959
- AsXMLString, 960
- DocumentBegin, 961
- DocumentEnd, 961
- Event, 962
- Goto, 963
- IsRedoEnabled, 964
- IsUndoEnabled, 964
- MarkupVisibility, 964

- OnBeforeCopy, 951
- OnBeforeCut, 951
- OnBeforeDelete, 952
- OnBeforeDrop, 952
- OnBeforePaste, 953
- OnDragOver, 954
- OnKeyboardEvent, 955
- OnMouseEvent, 956
- OnSelectionChanged, 957
- Parent, 964
- Print, 965
- Redo, 965
- Selection, 965
- Undo, 966
- WholeDocument, 967
- XMLDataRoot, 967

**Auto-Calculations,**

- and output escaping, 365

**Autokomplettierung in SQL-Scripts, 560****Automatisch ausblenden,**

- Funktion der Design-Eingabehilfenfenster, 39

**Automatische Berechnung,**

- ausblenden, 277
- Befehl für das Einfügen im Design, 746
- Node mit Wert aus autom. Ber. aktualisieren, 746
- Nodes im XML-Dokument mit Wert aktualisieren, 277

**Automatische Berechnungen,**

- auf Basis des Ergebnisses einer anderen automatischen Berechnung, 280
- Beispieldateien, 282
- Beispiele, 301
- erstellen, bearbeiten, formatieren, 274
- Formatieren der Datumsergebnisse, 473
- im Schnellstart-Tutorial, 88
- Java und .NET-Funktionen (nur in der Enterprise Edition), 274
- Nodes im XML-Dokument mit Wert aktualisieren, 280
- Symbol in der Design-Ansicht, 687
- und Bedingungen, 293
- verschieben, 274
- Verwendung, 273

**Automatische Nummerierung, 344****Automatisierte Verarbeitung, 626**

## B

**Barcode, 750**

**Barcodes, 191****Basisjahr,**

in der Eingabeformatierung, 367

**Batch-Dateien,**

erstellen, 642

und geplante Aufgaben, 641

und geplante Tasks (in Windows Vista), 646

und geplante Tasks (in Windows XP), 643

und PDF (nur Enterprise Edition), 642

zum Generieren von Dateien anhand eines SPS über die Befehlszeile, 628

**Bedingte Präsenz, 294****Bedingte Vorlagen,**

Symbol in der Design-Ansicht, 687

**Bedingung,**

Befehl für das Einfügen in das Design, 760

**Bedingungen,**

Ausgabe-basierte, 291

bearbeiten, 290

definieren, 287

einschließen in, 778

für unterschiedliche Ausgabeformate, 291

im Schnellstart-Tutorial, 92

und automatische Berechnungen, 293

**Bedingungsvorlagen, 760**

siehe auch Bedingungen, 286

**Bedingungszweig "Andernfalls", 287****Befehl "(SPS) öffnen", 702****Befehl "(SPS) schließen", 702****Befehl "Alles markieren", 721****Befehl "Ausschneiden", 720****Befehl "Beenden", 719****Befehl "Design speichern", 703****Befehl "Drucken", 718****Befehl "Druckvorschau", 718****Befehl "Einfügen", 720****Befehl "Ersetzen" (Enterprise und Professional Edition), 722****Befehl "Kopieren", 720****Befehl "Löschen", 720****Befehl "Neu", 695****Befehl "Rückgängig", 721****Befehl "Suchen", 722****Befehl "Weitersuchen", 722****Befehl "Wiederherstellen", 721****Befehle,**

anpassen, 824

**Befehlszeile, 626**

Beispiele für Befehle, 632

für XSLT-Transformationen, 639

StyleVision verwenden, 628

Syntax, 629

und geplante Aufgaben, 641

und geplante Tasks (in Windows Vista), 646

und geplante Tasks (in Windows XP), 643

und Parameter, 312

**Befehlszeilentool, 28****Beispiel für mehrere Sprachen, 358****Benutzerdefinierte Elemente, 140, 141, 770, 782****Benutzerdefinierte Schemas, 221****Benutzerdefinierte Textblöcke, 140, 770****Benutzerdefinierte Vorlage,**

einfügen, 763

einschließen in, 773

**Benutzerdefinierte Wörterbücher,**

für SPS Rechtschreibprüfung, 819

**Benutzerdefinierte XML-Textblöcke, 143****Benutzerdefinierte XPath-Funktionen, 454****Benutzerhandbuch,**

siehe auch Bildschirmhilfe, 834

**Benutzerinfo, 444****Benutzeroberfläche,**

Dokumentansichten, 33

Hauptfenster, 33

mehrere Dokumente, 33

siehe GUI, 32

**Benutzerreferenz, 670****Bestellen der Altova-Software, 835****Betriebssysteme für Altova-Produkte, 1173****Bild,**

Befehl zum Einfügen in das Design, 751

**Bilder,**

Aufrufen der Ausgabedarstellung, 172

Definieren von URIs, 172

im Schnellstart-Tutorial, 78

Symbol in der Design-Ansicht, 687

und URIs für ungeparste Entities, 172

unterstützte Typen, 175

**Bildschirmauflösung,**

und in Pixeln definierte Längen, 422

**Bildschirmhilfe,**

Index, 834

Inhaltsverzeichnis, 834

Suchen, 834

**Browser-Fenster,**

im Fenster "Datenbankabfrage", 556

## C

- C#**,
  - integration of StyleVision, 1061
  - Integration von StyleVision, 1061
- CALS/HTML-Tabellen, 161, 792**
- CDATA-Abschnitte, 129**
  - in der Authentic-Ansicht einfügen, 580
- Change To command, 206**
- Character references**,
  - and output escaping, 365
- Class**,
  - JAVA, 1003
- Class ID**,
  - in StyleVision Integration, 1053
- ClassPath statement, 1003**
- Collations**,
  - in XPath 2.0, 1133
- COM-API**,
  - Dokumentation, 874
- Composite styles**,
  - for Authentic View, 400
- Connection Strings**,
  - für Datenbanken über ADO, 511
  - für Datenbanken über ODBC, 518
- Content Model**,
  - Auswirkung von nicht eindeutigen Elementen im Design, 133
- Copyright-Informationen, 1180**
- count() function in XPath 2.0**,
  - siehe fn:count(), 1133
- count()-Funktion**,
  - in XPath 1.0, 1126
- CSS Stylesheets**,
  - externe Stylesheets, 380
  - Importvorrang von externen, 380
  - Medien, 380
  - siehe auch Stile, 380
- CSS-Dateien**,
  - in der Seitenleiste Design-Übersicht verwalten, 43
- CSS-Stile**,
  - im Schnellstart-Tutorial, 82
  - in modularen SPSS, 231
  - siehe auch Stile, 55
- Custom buttons for Authentic toolbar, 796**

## D

- Darstellung**,
  - siehe auch Formate, Formatierung, 362
  - Übersicht über Verfahren, 362
- Data Source Names (DSNs)**,
  - und ODBC-Verbindungen, 518
- Datei (Menü)**,
  - Datei | Generierte Dateien speichern, 712
  - Datei | Kodierung, 715
  - Datei | XML-Authentic-Daten speichern, 711
- Dateien**,
  - zuletzt verwendete öffnen, 719
- Dateien zu Projekt hinzufügen, 730**
- Datenbank**,
  - Connection Strings erstellen, 511, 518
  - Datensätze bearbeiten, 565
  - mit dem Verbindungsassistenten Verbindung herstellen, 507, 511, 518
  - Schema für SPS auswählen (Nicht-XML-Datenbanken), 527
  - Schema für SPS auswählen (XML-Datenbanken), 534
  - Symbolleisten-Schaltflächen für die Bearbeitung, 684
  - über gemeinsame Netzwerke und UNC-Pfade, 507
  - Verbindung herstellen von StyleVision, 507, 511, 518, 521
  - Verbindung über JDBC herstellen, 521
  - Verbindung zu einem MS SQL Server über ADO herstellen, 511
  - Verbindung zu einer IBM DB2 XML-Datenbank herstellen, 507
  - Verbindung zu einer IBM DB2 XML-Datenbank über ADO herstellen, 511
  - Verbindung zu einer IBM DB2 XML-Datenbank über JDBC herstellen, 521
  - Verbindung zu einer IBM DB2 XML-Datenbank über ODBC herstellen, 518
  - Verbindung zu einer MS Access DB herstellen, 507
  - XML-Arbeitsdatei für SPS auswählen (Nicht-XML-Datenbanken), 527
  - XML-Arbeitsdatei für SPS auswählen (XML-Datenbanken), 534
- Datenbank (Enterprise und Professional Edition)**,
  - siehe DB, 3
- Datenbank abfragen**,
  - siehe Datenbankabfrage (Fenster), 552
- Datenbank abfragen (Befehl), 552**
- Datenbankabfrage**,

**Datenbankabfrage,**

- Abfrage erstellen, 564
- Browser-Fenster im Fenster Datenbankabfrage, 556
- Ergebnisse, 565
- Ergebnisse bearbeiten, 565
- Meldungsregister, 565
- Verbindung zur DB für Abfrage herstellen, 554

**Datenbankabfrage (Fenster), 552**

- ein- und ausblenden, 810

**Datenbanken,**

- siehe auch DB, 595
- siehe unter DB, 502
- und globale Ressourcen, 442

**Datenbankschemas, 220****Dateneingabeelemente, 179**

- Symbol in der Design-Ansicht, 687

**Dateneingabe-Steuerelemente,**

- Menübefehle zum Einfügen, 744

**Datensätze bearbeiten, 565****Datensätze, die in der Authentic angezeigt werden,**

- definieren, 546

**Datentypen,**

- in DB-Filtern, 540
- in XPath 2.0 und XQuery 1.0, 1133
- Konvertierung von DB in XML-Schema, 1164
- siehe auch XML-Schema-Datentypen, 1165

**Dates,**

- und Datumswahl, 471

**Datum,**

- Formatierung, 367

**Datumswahl,**

- Befehl für das Einfügen im Design, 748
- Beschreibung der Verwendung, 471
- einfügen in das SPS, 471
- in der Authentic-Ansicht verwenden, 605
- lexikalisches Format von Datumseingaben, 471
- standardmäßig zu Datums-Nodes hinzufügen, 801
- und XSD-Datentypen, 471
- Verwendung in der Authentic-Ansicht, 471

**Datumswahl automatisch hinzufügen, 801****Datumswerte,**

- Beispiele für die Bearbeitung von Datumswerten mit XPath 2.0, 470
- formatieren, 473
- und Datumswahl, 748, 801
- verwenden im SPS, 470

**DB,**

- Abfragen erstellen, 597

Abfragen in der Authentic-Ansicht, 595

Anzeige in der Authentic-Ansicht filtern, 597

Arbeiten mit DB in StyleVision, 502

Bearbeiten in der Authentic-Ansicht, 595

connecting to from StyleVision, 525

connecting to via a global resource, 525

Datenbank-basiertes SPS einrichten, 505

Datentyp-Konvertierung in XML-Schema, 1164

Funktionsweise in StyleVision, 503

generierte XML-Datendateien, 503

generierte XML-Schema-Datei, 503

in der Authentic-Ansicht anzuzeigende Datensätze, 444

in der Authentic-Ansicht bearbeiten, 602

Navigieren in Tabellen in der Authentic-Ansicht, 596

Parameters in DB-Abfragen, 597

Schema für SPS auswählen, 526

Schema-Datei für, 537

Verbindung herstellen von StyleVision, 505

XML-Arbeitsdatei für SPS auswählen, 526

XML-Dateien und HTML/PDF-Ausgabe generieren, 550

XML-Datendatei, 537

**DB-Abfrage (Schaltfläche),**

- einfügen in SPS, 546

**DB-Filter,**

- bearbeiten, 811
- Daten für die XML-Datei filtern, 540
- Datentypen in, 540
- erstellen und ändern, 540
- löschen, 811

**DB-Parameter,**

- erstellen und bearbeiten, 724
- Verwendung, 540

**DB-Parameter-Standard Einstellungen, 540****DB-Steuerelemente, 802**

- automatisch einfügen, 546
- Befehl zum Einfügen, 745

**Debuggen von Makros, 864****deep-equal() function in XPath 2.0,**

- siehe fn:deep-equal(), 1133

**Default Functions Namespace,**

- für XPath 2.0- und XQuery 1.0-Ausdrücke, 1133
- in XSLT 2.0 Stylesheets, 1129

**Design Filters,**

- switching on and off, 739

**Design Fragment,**

- einfügen, 765

**Design-Ansicht, 686**

- Anzeige von Markup, 35

**Design-Ansicht, 686**

- Beschreibung, 35
- Symbole im SPS Design, 687
- und JavaScript Editor, 35

**Design-Eingabehilfen,**

- ausblenden, 39
- automatisch ausblenden, 39
- Beschreibung, 39
- ein- und ausblenden, 738

**Design-Eingabehilfenfenster,**

- abdocken, 39
- andocken, 39

**Design-Elemente, 679****Design-Fragmente, 256****Design-Struktur, 210**

- Eingabehilfenfenster, 48
- siehe auch Design-Eingabehilfenfenster, 39
- und modulare SPSs, 231

**Design-Übersicht,**

- Seitenleiste, 43

**Dezimalzahlen,**

- Formatierung, 367

**Dialogfeld "Anpassen",**

- zum Anpassen von StyleVision, 737

**Dialogfeld "Parameter bearbeiten", 724****Dialogfeld "XPath",**

- Beschreibung, 691

**Dialogfeld "XPath-Ausdruck bearbeiten",**

- siehe Dialogfeld "XPath", 691

**disable-output-escaping, 365****Document,**

- Application, 969
- Close, 969
- FullName, 970
- Name, 970
- Path, 971
- Save, 971
- SaveAs, 971
- Saved, 971

**Dokumentanfangsabschnitt, 412****Dokumentansicht,**

- auf der Benutzeroberfläche, 33

**Dokumentation,**

- Überblick, 6

**Dokumente,**

- öffnen und schließen, 33

**Dokumentebene,**

- Beispiele für die Integration von StyleVision, 1061

- Integration des StyleVisionControl, 1056

- Integration von StyleVision, 1057, 1058, 1059, 1060

**Dokumenteigenschaften, 269****Dokumentelement,**

- Definition, 24

**Dokumentelemente (siehe Root-Elemente), 212****Dokument-Events, 860****Dokument-Node,**

- Definition, 24

**Dokumentstile, 269****DPI, 715**

- und Druckausgabe, 422
- und in Pixeln definierte Längen, 422

**Druckausgabe,**

- siehe paginierte Medien, 408

**DSN,**

- siehe unter Data Source Names (DSNs), 518

**DTDs,**

- als SPS Quelle, 215

**Dynamische (SPS) Tabellen in der Authentic-Ansicht,**

- Verwendung, 586

**Dynamische Listen, 165, 168****Dynamische Tabelle,**

- Symbolleisten-Schaltflächen zum Bearbeiten, 674

**Dynamische Tabellen, 144**

- bearbeiten, 571
- geschachtelte dynamische Tabellen, 148
- in der Authentic-Ansicht bearbeiten, 808
- Kopf- und Fußzeilen, 148
- siehe auch SPS-Tabellen, 148
- siehe auch Tabellen, 155
- und globale Vorlagen, 148
- Unterschied zwischen angehängten/eingefügten Zeilen, 148

**Dynamischer Inhalt,**

- im Schnellstart-Tutorial, 71

## E

**Eclipse Plattform,**

- und StyleVision, 658
- und StyleVision Integrationspaket, 659

**Eclipse-Plattform,**

- StyleVision Perspektive, 663

**Edit Template Match command, 137****Eigenschaften,**

- definieren, 57

**Eigenschaften,**

- für Nodes in der Authentic-Ansicht, 444
- Seitenleistenfenster, 57
- siehe auch Design-Eingabehilfen, 39
- und Eigenschaftsgruppen, 57
- von SPS Tabelle, 791
- von SPS Tabellen, 674

**Eigenschaften und Events (Fenster), 845****Eigenschaften von Ausgabedokumenten, 269****Einfache globale Vorlagen, 243****Einfügen,**

- als Text, 580
- als XML, 580
- Spalte in SPS Tabelle einfügen, 786
- Zeile in Tabelle im SPS einfügen, 786

**Einfügen (Menü),**

- Einfügen | Design Fragment, 765

**Einfügen als,**

- Text, 576
- XML, 576

**Einfügen von Design-Elementen über die Symbolleiste, 679****Eingabefelder, 182****Eingabeformatierung, 367**

- Formatierungsstrings definieren, 814
- von Datumswerten, 473

**Eingabehilfe "Eigenschaften",**

- Gruppe "Ereignis", 479

**Eingabehilfen in der Design-Ansicht,**

- ein- und ausblenden, 738

**Eingabemethoden, 1177****Eingabetaste,**

- drücken, 610

**Einrichten eines neuen SPS Dokuments,**

- im Schnellstart-Tutorial, 67

**Einrichten von StyleVision, 28****Einschließen in (Menü), 771****Einstellung für automatisches Makro, 862****Elemente,**

- benutzerdefinierte, 141
- in der Authentic-Ansicht und im SPS hinzufügen, 133

**Elemente zur Navigation in DB-Tabellen, 802****Elementvorlagen,**

- benutzerdefinierte, 141

**Encoding,**

- für Ausgabedateien, 830

**Endbenutzer-Lizenzvereinbarung, 1180, 1184****Entities,**

- in der Authentic-Ansicht definieren, 580, 607, 676, 805
- in der Authentic-Ansicht einfügen, 580

**Entity references,**

- and output escaping, 365

**Enumerationen,**

- in StyleVisionControl, 1121

**Erstellen eines neuen SPS Dokuments,**

- im Schnellstart-Tutorial, 67

**Erweiterungsfunktionen für XSLT und XQuery, 1139****Erweiterungsfunktionen in .NET für XSLT und XQuery,**

- siehe .NET Erweiterungsfunktionen, 1149

**Erweiterungsfunktionen in Java für XSLT und XQuery,**

- siehe Java-Erweiterungsfunktionen, 1140

**Erweiterungsfunktionen in MSXSL Scripts, 1155****Evaluierungs-Keycode,**

- für Ihre Altova-Software, 835

**Evaluierungszeitraum,**

- bei Altova Software-Produkten, 1183
- für Altova-Software-Produkte, 1181
- von Altova Software-Produkten, 1180

**Event, 951, 952, 953, 954, 955, 956, 957****Event Handler,**

- Funktionen zuweisen, 479
- im Skripting-Projekt, 860
- Übersicht, 849

**Events, 860****Externes Verzeichnis / Webverzeichnis zum Projekt hinzufügen, 733**

## F

**Features,**

- von StyleVision, 22

**Filter,**

- zum selektiven Anzeigen von Vorlagen, 682

**Filter (für DB),**

- bearbeiten, 811
- löschen, 811

**Filter an Node-Vorlagen, 252****Filters for design templates,**

- switching on and off, 739

**fn:base-uri in XPath 2.0,**

- Unterstützung in Altova-Prozessoren, 1135

**fn:collection in XPath 2.0,**

- Unterstützung in Altova-Prozessoren, 1135

**fn:count() in XPath 2.0,**



**fn:count() in XPath 2.0,**  
und Whitespace, 1133

**fn:current-date in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:current-dateTime in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:current-time in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:data in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:deep-equal() in XPath 2.0,**  
und Whitespace, 1133

**fn:id in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:idref in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:index-of in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:in-scope-prefixes in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:last() in XPath 2.0,**  
und Whitespace, 1133

**fn:lower-case in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:normalize-unicode in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:position() in XPath 2.0,**  
und Whitespace, 1133

**fn:resolve-uri in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:static-base-uri in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**fn:upper-case in XPath 2.0,**  
Unterstützung in Altova-Prozessoren, 1135

**FO-Prozessor (Enterprise Edition),**  
Einrichten, 28

**Format,**  
Befehl für das Einfügen im Design, 749

**Formatieren,**  
von Nodes beim Einfügen, 132  
von numerischen Feldern, 367

**Formatierung,**  
für Tabellen, 155  
siehe auch Darstellung, 362  
Übersicht über Verfahren, 362

**Formatierungsstrings,**  
für Eingabeformatierung definieren, 814

**Formularbasierte Designs, 195, 695**

**Formulare,**  
benennen, 853  
Eigenschaften, 853  
in Skripting-Projekten, 852  
neues erstellen, 853  
Übersicht, 849  
und Event-Behandlung, 858  
und Formularobjekte, 855  
und vordefinierte Befehle, 867

**Formularobjekteigenschaften, 855**

**Formularobjektpalette, 845**

**Formular-Steuerelemente,**  
Menübefehle zum Einfügen, 744

**FO-Transformationen, 636, 637**

**Fragen und Antworten zu StyleVision, 837**

**functions,**  
XPath 2.0 and XQuery 1.0, 1132

**Funktionen,**  
in XPath, benutzerdefiniert, 454

**Funktionen,**  
siehe unter XSLT 2.0-Funktionen, 1131

**Fußzeilen,**  
in Tabelle hinzufügen, 785  
in Tabellen, 155

**Fußzeilen und Kopfzeilen,**  
auf paginierten Ausgabeseiten, 415, 418  
mit Zwischensummen, 418

## G

**Ganzzahl,**  
Formatierung, 367

**Gemischte Markup-Symbole, 444**

**Generierte Dateien, 115**

**Generierte Dateien speichern (Befehl), 712**

**Geplante Aufgaben,**  
Erstellen eines StyleVisionBatch Befehls, 641  
StyleVisionBatch Batch-Dateien, 641

**Geplanter Task,**  
Erstellen eines StyleVisionBatch Befehls, 643, 646  
StyleVisionBatch Batchdateien, 643, 646

**Gesamtseitenzahl (Enterprise Edition), 769**

**Global, 1176**

**Global Resource,**  
and databases, 525

**Globale Deklarationen, 851**

**Globale Deklarationen, 851**

Übersicht, 849

**Globale Ressourcen, 427**

definieren, 428

Dialogfeld, 822

Konfiguration über Symbolleiste auswählen, 683

Konfigurationen ändern, 443

Konfigurationen kopieren, 437

Symbolleiste, 683

Typ Datei definieren, 431

Typ Datenbank definieren, 435

Typ Ordner definieren, 434

verwenden, 438, 442, 443

vom Typ Datei und Ordner verwenden, 439

**Globale Stile,**

Seitenleistenfenster, 52

**Globale Typen,**

in Vorlagen, 243

**Globale Vorlage, 241****Globale Vorlagen, 242, 243**

Auswirkung auf restlichen Inhalt, 136

im Schnellstart-Tutorial, 99

**Globaler Stil,**

siehe Stile, 384

**Grafiken,**

Beispieldateien, 178

siehe auch Bilder, 171

Übersicht über die Verwendung im SPS, 171

**Grafikformate,**

in der Authentic-Ansicht, 609

**Grafikvorlagen für das Layout, 196****Grid-Ansicht-Events, 860****Große Markup-Symbole anzeigen, 571, 573****Gruppieren, 296**

group-by Beispiel (Persons.sps), 299

group-by Beispiel (Scores.sps), 301

**GUI,**

Beschreibung, 32

**H****Hauptschema, 242****Hauptschema (nur Enterprise Edition), 46****Hauptvorlage, 241, 242**

Definition, 24

**Hebräisch, 1178****Hilfe,**

siehe Bildschirmhilfe, 834

**Hintergrundinformationen, 1172****Hiragana, 1177****Horizontale Linie,**

Befehl für das Einfügen im Design, 752

im Schnellstart-Tutorial, 78

**HTM-Import,**

neues SPS erstellen, 483

**HTML,**

Integration von StyleVision, 1061

**HTML-Ausgabe, 115**

und Unterstützung von Bildern, 175

**HTML-Beispiel,**

für StyleVisionControl Integration, 1053, 1054

**HTML-Import,**

Dateien anhand des SPS generieren, 489

Schemastruktur, 485

SPS Design, 485

von HTML-Listen, 487

von HTML-Tabellen, 487

**HTML-Tabellen, 161, 792****Hyperlink,**

Befehl für das Einfügen im Design, 758

**Hyperlinks, 190, 350**

einschließen in, 777

entfernen und löschen, 354

erstellen und bearbeiten, 354

mit externen Ressourcen verknüpfen, 354

mit Lesezeichen verknüpfen, 354

Symbol in der Design-Ansicht, 687

und ungeparste Entities, 354

**I****IBM DB2 XML DB,**

mit dem Verbindungsassistenten Verbindung herstellen, 507

Verbindung per ADO herstellen, 511

Verbindung per ODBC herstellen, 518

**IE 9,**

siehe Kompatibilität mit Internet Explorer, 110

**Implementierungsspezifisches Verhalten,**

von XSLT 2.0-Funktionen, 1131

**Implizite Zeitzone,**

und XPath 2.0-Funktionen, 1133

**Import von XSLT-Vorlagen,**

ins SPS, 259

**Inhalt,**

Befehl für das Einfügen im Design, 741

**Inhaltsverz.-Ebenen,**

einschließen in, 780

**Inhaltsverz.-Markierungen,**

einschließen in, 780

**Inhaltsverzeichnis,**

Beispiel, einfach, 336

Beispiel, hierarchische und sequenzielle, 340

Einträge für das Inhaltsverz. markieren, 323

Menübefehle, 767

Übersicht über die Verwendung, 320

**Inhaltsverzeichnisenebenen, 323, 325****Inhaltsverzeichniseinträge,**

erstellen, 335

formatieren, 335

**Inhaltsverzeichnismarkierungen, 323**

Assistent für, 328

erstellen, 328

und Ebenen, 328

**Inhaltsverzeichnis-Vorlage,**

Ebenenreferenzen darin, 333

erstellen und bearbeiten, 331

formatieren, 335

strukturieren, 333

**Inhaltsverzref,**

siehe Inhaltsverzeichnisreferenzen, 334

**Inhaltsverzeichnisreferenzen, 334****Integrating,**

StyleVision in Applikationen, 1051

**Internet Explorer-Kompatibilität, 110****Internet-Verwendung,**

in Altova-Produkten, 1179

**iSeries,**

timeout muss deaktiviert werden, 505

**ISO/IEC 10646, 1176**

## J

**JAVA,**

API, 1003

ClassPath, 1003

**Java Erweiterungsfunktionen,**

Datentypkonvertierungen, Java in XPath/XQuery, 1147

Instanzmethoden, Instanzfelder, 1145

Konstruktoren, 1144

statische Methoden, statische Felder, 1145

**Java extension functions,**

datatype conversions, XPath/XQuery to Java, 1146

**Java und .NET-Funktionen (nur in der Enterprise Edition),**

in automatischen Berechnungen, 274

**Java-Erweiterungsfunktionen,**

benutzerdefinierte JAR-Dateien, 1143

benutzerdefinierte Klassendateien, 1141

für XSLT und XQuery, 1140

in XPath-Ausdrücken, 691

Übersicht, 1140

Unterstützung in der Authentic-Ansicht, 691

**JavaScript,**

siehe Scripts, 476

**JavaScript Editor, 476, 478**

in der Design-Ansicht, 35

**JDBC-Verbindungen zu Datenbanken, 521****JRE,**

für StyleVision Plugin für Eclipse, 659

## K

**Kanji, 1177****Katakana, 1177****Katalogdateien, 123****Keycodes,**

für Ihre Altova-Software, 835

**Klassenattribute,**

im Schnellstart-Tutorial, 82

**Kleine Markup-Symbole anzeigen, 571, 573****Kodierung (Befehl), 715****Kodierung von Ausgabedokumenten, 715****Kombistile, 396****Kommentare in SQL-Scripts, 560****Kompatibilität mit Internet Explorer, 110****Komplexe globale Vorlagen, 243****Komponenten-Download-Center,**

auf der Altova Website, 837

**Konfigurationen,**

einer globalen Ressource, 428, 823

**Konfigurationen in Globalen Ressourcen, 443****Kontextmenüs,**

in der Authentic-Ansicht, 576

**Kontext-Node,**  
     im Dialogfeld "XPath", 691  
**Kontrollkästchen, 183**  
**Konvertierung von HTML in XML, 482**  
**Kopf- und Fußzeilen,**  
     auf paginierten Ausgabeseiten, 415, 418  
     mit Zwischensummen, 418  
**Kopfzeilen,**  
     in Tabelle hinzufügen, 785  
     in Tabellen, 155  
**Kriterien,**  
     in DB-Filtern, 540

## L

**last() function in XPath 2.0,**  
     siehe fn:last(), 1133  
**last()-Funktion,**  
     in XPath 1.0, 1126  
**Laufende Zwischensummen,**  
     in Kopf- und Fußzeilen, 418  
**Layout,**  
     von Ansichten der Benutzeroberfläche, 39  
**Layout Container, 196, 766**  
**Layout Container und Elemente, 679**  
**Layout-Feld, 766**  
**Layout-Felder, 200**  
**Layout-Module,**  
     Schritte bei der Erstellung, 195  
**Lesezeichen, 190, 350**  
     Befehl für das Einfügen im Design, 757  
     einschließen in, 777  
     erstellen und bearbeiten, 351  
     löschen, 351  
**Lesezeichen (Anchors),**  
     Symbol in der Design-Ansicht, 687  
**Lesezeichen in SQL-Scripts, 560**  
**Linie,**  
     in Layout Containern, 766  
**Links,**  
     in der Authentic-Ansicht verwenden, 580  
     siehe Hyperlinks, 190, 350  
**Listen, 165**  
     Befehl zum Einfügen in das Design, 754  
     einschließen in, 776  
     im Schnellstart-Tutorial, 92

    vom HTML-Dokument importierte, 487

**Listeneigenschaften, 813**

**Lizenz, 1184**

    für Ihre Altova-Software, 835

    Informationen, 1180

**Lizenzüberwachung,**

    bei Altova-Produkten, 1182

**Lokale Stile,**

    siehe Stile, 388

**Lokale Vorlage, 241, 242**

**Löschen,**

    eines DB-Filters, 540

    Spalte aus Tabelle im SPS, 787

    Tabelle im SPS, 784

    Zeile aus Tabellen im SPS, 787

## M

**Macros,**

    running, 863

**Makros,**

    debuggen, 864

    Funktionen für Makros in globalen Deklarationen, 851

    im Skript-Editor als automatisches Makro definieren, 862

    mit dem Skript-Editor bearbeiten, 862

    mit dem Skript-Editor erstellen, 862

    Übersicht, 849

    Verwendung im Skripting-Projekt, 861

**Markup,**

    in Authentic-Ansicht, 806

    in der Authentic-Ansicht, 571

**Markup anzeigen, 35**

**Markup ausblenden, 35**

**Markup-Symbole,**

    in der Authentic-Ansicht, 573

**Markup-Symbole ausblenden, 571, 573**

**Markup-Tags in der Design-Ansicht, 35**

**Mehrere Ausgabedokumente, 261**

    Dateipfad, 267

    und Ausgabevorschauen, 267

    verknüpfen, 264

**Mehrfach-Dokumentausrage, 768**

**Mehrzeilige Eingabefelder, 182**

**Menü,**

    anpassen, 824

**Menü "Ansicht", 736**

**Menü "Authentic", 794****Menü "Bearbeiten", 720****Menü "Datei", 694**

- Befehl "Beenden", 719
- Datei | Design speichern, 703
- Datei | Drucken, 718
- Datei | Druckvorschau, 718
- Datei | Neu, 695
- Datei | Öffnen, 702
- Datei | Schließen, 702
- Datei | Speichern unter, 708

**Menü "Eigenschaften", 812**

- Nummerierung und Aufzählungszeichen, 813

**Menü "Einfügen", 740**

- Einfügen | Absatz, 749
- Einfügen | Automatische Berechnung, 746
- Einfügen | Bedingung, 760
- Einfügen | Bild, 751
- Einfügen | Datumswahl, 748
- Einfügen | Format, 749
- Einfügen | Hyperlink, 758
- Einfügen | Inhalt, 741
- Einfügen | Lesezeichen, 757
- Einfügen | Nummerierung und Aufzählungszeichen, 754
- Einfügen | Restlicher Inhalt, 742
- Einfügen | Seite, 769

**Menü "Extras", 816****Menü "Fenster", 832****Menü "Hilfe", 833****Menü "Tabelle", 783****Menü "Einfügen",**

- Einfügen | Horizontale Linie, 752

**Menüleiste,**

- verschieben, 32

**Microsoft Office 2007 (nur Enterprise Edition), 22, 38****Mixed Markup-Symbole anzeigen, 571, 573****Modulares SPS,**

- aktivieren und deaktivieren, 231
- arbeiten mit, 231
- Auswirkung der Reihenfolge, 231
- Beispielprojekt, 236
- das hinzuzufügende SPS Modul, 231
- das SPS Modul hinzufügen, 231
- erstellen, 231
- Übersicht, 226
- und CSS-Stile, 228, 231
- und Modulobjekte, 228
- und Namespace-Deklarationen, 228

und Schemaquellen, 228

und Scripts, 228

und Shemaquellen, 231

und Warnmeldungen über geänderte Dateien, 231

und XML-Arbeitsdateien, 228

und XML-Vorlagendateien, 228

**Module,**

in der Seitenleiste Design-Übersicht verwalten, 43

**MS Access,**

Konvertierung von Datentypen in XML-Schema  
Generierung von DB, 1165

**MS Access DB,**

mit dem Verbindungsassistenten Verbindung herstellen,  
507

über gemeinsame Netzwerke und UNC-Pfade, 507

**MS SQL Server,**

Konvertierung von Datentypen in XML-Schema  
Generierung von DB, 1166

per ADO Verbindung herstellen, 511

**msxsl:Script, 1155****MySQL,**

Konvertierung von Datentypen in XML-Schema  
Generierung von DB, 1167

## N

**Nachfolgende Markup-Tags, 35****Named Templates, 241****Namen hinzufügen, 444****Namespaces,**

- im SPS, 46
- im XSLT 2.0 Stylesheet, 1129
- Übersicht, 48
- zum SPS hinzufügen, 46, 107, 115, 212

**Netzwerkfreigaben,**

für Datenbankankbindung, 507

**Neu von XSLT-Datei, 451****Neue Dokumentvorlagen, 261****Neue Features,**

v2010, 13

**Neue Funktionalitäten, 10****Neue Release von StyleVision,**

mit StyleVision synchronisieren, 114

**Neues Projekt, 728****New features,**

v2011, 11

**Nicht lateinischen Schriftsysteme, 1177**

**Node,**

changing what it is created as, 206

**Nodes aktualisieren,**

mit dem Wert aus automatischen Berechnungen, 280

**Nodes aktualisieren (Enterprise und Professional Edition),**

mit Werten von automatischen Berechnungen, 277

**Nodes automatisch nummerieren, 344****Node-templates,**

User-Defined, 137

**Node-Vorlagen,**

Operationen an, 252

und an Child-Vorlagen binden, 252

und globale Vorlagen, 252

und XPath-Filter, 252

**Numerische Felder,**

Formatierung, 367

**Nummerierung formatieren,**

bei der automatischen Nummerierung, 344

**Nummerierung und Aufzählungszeichen, 165, 166, 168, 813**

Befehl zum Einfügen in das Design, 754

einschließen in, 776

# O

**Objektsuche,**

im Fenster "Datenbankabfrage", 556

**ODBC,**

Konvertierung von Datentypen in XML-Schema  
Generierung von DB, 1169

**ODBC-Verbindungen zu Datenbanken, 518****Office Open XML (nur Enterprise Edition), 22, 38****Öffnen,**

zuletzt verwendete Dateien, 719

**OOXML (nur Enterprise Edition), 22, 38****Optionsfelder, 189****Oracle,**

Konvertierung von Datentypen in XML-Schema  
Generierung von DB, 1168

**Output escaping, 365**

# P

**Paginierte Medien,**

Design für, 408

Eigenschaften, 408

Kopf- und Fußzeilen, 415

und in Pixeln definierte Längen, 422

**Parameter, 311**

allgemeine Beschreibung, 312

erstellen und bearbeiten, 724

für Design-Fragmente, 314

für Schemaquellen, 317

im SPS, 312

in DB-Abfragen, 597

in DB-Filtern, 540

in der Seitenleiste Design-Übersicht verwalten, 43

Nodes in mehreren Dokumenten mit Param. identifizieren, 317

Übersicht über benutzerdefinierte Parameter, 48

und Authentic-Ansicht, 312

und Befehlszeile, 312

**Parameter-Standardeinstellungen,**

in DB-Filtern, 540

**Parent, 970****Parser,**

in Altova-Produkte integrierter, 1174

**Partner-Software, 837****PDF-Ausgabe,**

siehe paginierte Medien, 415

**PDF-Ausgabe (Enterprise Edition), 115**

und Unterstützung von Bildern, 175

**Pixel,**

und Bildschirmauflösung, 715

und Länge in der Druckausgabe, 715

**Pixel-definierte Längen,**

und paginierte Medien, 422

**Plattformen für Altova-Produkte, 1173****Platzhalter "Contents",**

im Schnellstart-Tutorial, 71

**Platzhalter "Inhalt",**

Node als Inhalt einfügen, 129

**position() function,**

in XPath 1.0, 1126

**position() function in XPath 2.0,**

siehe fn:position(), 1133

**Probleme mit der Vorschau, 28****Produktfeatures,**

Auflistung, 22

**Projekt (Menü), 726**

Aktive Datei zu Projekt hinzufügen (Befehl), 732

**Projekt (Menü), 726**

Aktive und verwandte Dateien zu Projekt hinzufügen (Befehl), 732  
Altova Ressource zu Projekt hinzufügen (Befehl), 730  
Dateien zu Projekt hinzufügen (Befehl), 730  
Externes Verzeichnis / Webverzeichnis zum Projekt hinzufügen (Befehl), 733  
Neu (Befehl), 728  
Neu laden (Befehl), 728  
Öffnen (Befehl), 728  
Projektordner zu Projekt hinzufügen (Befehl), 733  
Schließen (Befehl), 729  
Speichern (Befehl), 729  
URL zu Projekt hinzufügen (Befehl), 730

**Projekt (Seitenleiste), 62****Projekt neu laden, 728****Projekt öffnen, 728****Projekt schließen, 729****Projekt speichern, 729****Projekte,**

Beschreibung, 117  
und Drag-and-Drop, 726  
Verwendung, 117

**Projektoptionen, 830****Projektordner zu Projekt hinzufügen, 733****Prozessoren,**

zum Downloaden, 837

**Prozessorinformationen, 1125****PXF-Dateien, 494**

bearbeiten, 498  
bereitstellen, 500  
erstellen, 495  
speichern unter, 708

## Q

**QName Serialisierung,**

Rückgabe durch XPath 2.0 -Funktionen, 1135

**Quelldateien für das SPS, 107****Querverweise, 348**

## R

**Rechtliches, 1180****Rechtschreibprüfung,**

in StyleVision, 817

**Rechtschreibprüfungsoptionen,**

für SPSs, 819

**Regionen in SQL-Scripts, 560****Registrierung der Altova-Software, 835****Restlicher Inhalt, 136**

Befehl zum Einfügen in ein Design, 742  
im Schnellstart-Tutorial, 99  
und globale Vorlagen, 243

**Rich Edit, 402****RichEdit,**

commands in Authentic View, 807  
einfügen, 743

**RichEdit Symbolleiste, 678****Root-Elemente, 46****Root-Elemente (auch Dokumentelemente genannt),**

Auswahl für das Schema, 212  
und Schemaquellen, 212

**RTF-Ausgabe,**

siehe paginierte Medien, 415

**RTF-Ausgabe (Enterprise Edition), 115**

und Unterstützung von Bildern, 175

**Rückwärtskompatibilität,**

des XSLT 2.0-Prozessors, 1129

## S

**Schaltflächen, 189****Schema für datenbankbasierte SPSs, 527, 534****Schema hinzufügen, 695****Schemafähigkeit,**

des XPath 2.0- und XQuery-Prozessors, 1133

**Schemaquellen, 107, 695**

Auswahl für das SPS, 212  
in der Seitenleiste Design-Übersicht verwalten, 43  
mehrere im SPS (Enterprise Edition), 212  
mehrere Quellen und Nodes identifizieren, 317  
mehrere Quellen und XPath, 317  
Quellen ändern, 317  
Seitenleistenfenster, 46  
Übersicht, 48  
und Root-Elemente (Dokumentelemente), 212

**Schemaquellen-Fenster,**

siehe auch Design-Eingabehilfenfenster, 39

**Schemas,**

als SPS Quelle, 215

**Schemas,**

- benutzerdefinierte, 221
- für das SPS anhand einer Datenbank, 220

**Schemastruktur,**

- und SPS Design, 133

**Schemastrukturoptionen, 830****Schnellstart-Tutorial,**

- Auswahllisten, 92
- Authentic-Ansicht überprüfen (Enterprise und Professional Edition), 103
- automatische Berechnungen, 88
- Bedingungen, 92
- Bilder, 78
- CSS-Stile, 82
- dynamischer Inhalt, 71
- Einführung, 66
- Einrichten eines neuen SPS Dokuments, 67
- erforderliche Dateien, 66
- Erstellen eines neuen SPS Dokuments, 67
- globale Vorlagen, 99
- horizontale Linien, 78
- Klassenattribute, 82
- Listen, 92
- Platzhalter "Contents", 71
- restlicher Inhalt, 99
- statischer Inhalt, 78
- statischer Text, 78
- vordefinierte Formate, 82
- XSLT Stylesheets generieren, 103

**Schriftart, 1177****Schriftsystem, 1177****Schriftsysteme, 1178****Scripts,**

- in der Design-Struktur, 476
- in einem SPS verwenden, 476
- JavaScript-Funktionen als Event Handler, 479
- JavaScript-Funktionen definieren, 478
- und JavaScript-Funktionen, 476

**Scripts in XSLT/XQuery,**

- siehe Erweiterungsfunktionen, 1139

**Scroll-Schaltflächen,**

- im Hauptfenster, 33

**Seite,**

- Design-Befehle, 769

**Seitennummern (Enterprise Edition), 769****Seitenumbrüche, 421, 769****Select Tables (Dialogfeld),**

- für datenbankbasierte SPSs, 527, 534

**Skript-Editor, 615**

- Beschreibung der Benutzeroberfläche, 845
- Hauptfenster, 845

**Skripterstellung,**

- in der Authentic-Ansicht, 612, 615, 616

**Skripting-Projekt,**

- Anmerkungen zur Programmierung, 865
- Application Event Handler, 860
- Event Handler, 849
- Formulare, 849, 852
- Globale Deklarationen, 849, 851
- Makros, 849
- Makros in, 861
- und Events, 860

**Skripting-Projektstrukturfenster, 845****Skripting-Umgebung, 841**

- Übersicht über die Verwendung, 843

**Skripts,**

- Übersicht, 48

**Software-Produktlizenz, 1184****Sortieren, 305**

- Beispieldateien, 308
- Funktionsweise, 306
- Sortierschlüssel, 306
- von Gruppen und innerhalb von Gruppen, 296, 299, 301

**Sortierschlüssel, 305****Spalte,**

- an SPS Tabelle anhängen, 786
- aus Tabelle im SPS löschen, 787
- in SPS Tabelle einfügen, 786

**Spalten,**

- einfügen, 769
- Umbrüche erzwingen, 769

**Spalten (einer Tabelle),**

- in der HTML-Ausgabe ausblenden, 159

**Speichern,**

- XML-Arbeitsdatei, 711

**Spezieller Absatz,**

- einschließen in, 775

**SPS,**

- allgemeine Beschreibung, 19
- neu laden, 702
- öffnen, 702
- schließen, 702
- und Datenbanken, 502
- und die Authentic-Ansicht (Enterprise und Professional Edition), 19
- und StyleVision, 19



- SPS,**
  - und XSLT Stylesheets, 19
- SPS auf Basis einer Datenbank,**
  - siehe Datenbanken, 505
- SPS Design-Übersicht, 108**
- SPS und Authentic-Ansicht, 112**
- SPS-Dateistruktur, 210**
- SPS-Tabellen,**
  - dynamischen Tabellen bearbeiten, 571
  - siehe auch dynamische Tabellen, 144
  - siehe auch statische Tabellen, 144
- SPS-Tabellen in der Authentic-Ansicht,**
  - Verwendung, 586
- SQL Editor,**
  - Abfrage erstellen, 564
  - Beschreibung, 560
  - im Fenster Datenbankabfrage, 560
- Standard-Benutzerwörterbuch,**
  - für SPS Rechtschreibprüfung, 819
- Static text,**
  - and output escaping, 365
- Statische (SPS) Tabellen in der Authentic-Ansicht,**
  - Verwendung, 586
- Statische Listen, 165, 166, 776**
- Statische Tabelle,**
  - einfügen, 784
- Statische Tabellen, 144**
  - in das SPS einfügen, 674
  - siehe auch SPS-Tabellen, 146
  - siehe auch Tabellen, 155
  - Symbolleisten-Schaltflächen zum Bearbeiten, 674
- Statischer Inhalt,**
  - im Schnellstart-Tutorial, 78
- Statischer Text,**
  - im Schnellstart-Tutorial, 78
- Statusleiste, 737**
- Stile,**
  - Arbeiten mit Stilen in StyleVision, 378
  - aus XML-Daten, 393
  - definieren, 55
  - globale Stile im SPS definieren, 384
  - Kombinieren von CSS-Regeln, 396
  - lokale Stile definieren, 388
  - Medien für zugewiesene externe Stylesheets, 380
  - mehrere kombinieren, 396
  - mittels XPath-Ausdrücken, 393
  - Seitenleistenfenster, 55
  - siehe auch Design-Eingabehilfen, 39
  - stufenförmige Reihenfolge, 378
  - Terminologie, 378
  - und Eigenschaftsgruppen, 55
  - Vorrang, 52
  - Vorrangigkeit von Stilen, 384
  - Zuweisen von CSS Stylesheets zu SPS, 380
- Stile von Ausgabedokumenten, 269**
- Stil-Repository,**
  - siehe auch Design-Eingabehilfen, 39
  - und externe CSS Stylesheets, 380
  - und globale Stile, 384
- Struktur des SPS-Designs, 210**
- Stylesheets,**
  - siehe auch unter CSS Stylesheets, 380
  - siehe auch unter XSLT Stylesheets, 380
- StyleVision,**
  - Benutzerhandbuch, 3
  - Einführung, 18
  - Integration, 1051
  - mit Authentic synchronisieren, 114
  - Produktfeatures, 22
  - über die Befehlszeile ausführen, 628
- StyleVision API,**
  - aufrufen, 1101
- StyleVision command table, 1064**
- StyleVision Integration,**
  - Beispiel, 1053, 1054
- StyleVision Integrationspaket, 655, 659**
- StyleVision Perspektive in Eclipse, 663**
- StyleVision Plugin für Eclipse,**
  - installieren, 659
- StyleVision Plugin für VS .NET,**
  - installieren, 655
- StyleVision Power Stylesheet,**
  - siehe SPS, 3
- StyleVisionBatch, 28, 626**
  - Befehlszeilentool, 628
  - Beispiele für Befehle, 632
  - Syntax, 629
- StyleVisionCommand,**
  - in StyleVisionControl, 1103
- StyleVisionCommands,**
  - in StyleVisionControl, 1105
- StyleVisionControl, 1106**
  - Beispiel für die Integration auf Applikationsebene, 1053
  - Beispiel für Integration auf Applikationsebene, 1053, 1054
  - Beispiele für die Integration auf Dokumentenebene, 1061
  - Dokumentation, 1051

**StyleVisionControl, 1106**

- Integration auf Applikationsebene, 1052
- Integration auf Dokumentebene, 1056, 1057, 1058, 1059, 1060
- Integration mittels C#, 1061
- Integration mittels HTML, 1061
- integration using C#, 1061
- object reference, 1102

**StyleVisionControlDocument, 1112****StyleVisionControlPlaceHolder, 1118****Support für StyleVision, 837****Support-Optionen, 6****Sybase,**

- Konvertierung von Datentypen in XML-Schema
- Generierung von DB, 1171

**Symbole in der Design-Ansicht,**

- von automatischen Berechnungen, 687
- von Bedingungsvorlagen, 687
- von Bildern, 687
- von Dateneingabeelementen, 687
- von Hyperlinks, 687
- von Lesezeichen (Anchors), 687
- von vordefinierten Formaten, 687
- von XML-Dokumentinhalt, 687
- von XML-Dokument-Nodes, 687

**Symbolleisten, 671**

- "Design-Elemente einfügen" Symbolleiste, 679
- anpassen, 737
- Anzeige ein-/ausschalten, 671
- auf der Benutzeroberfläche positionieren, 671
- Authentic-Symbolleiste, 676
- ein- und ausblenden, 737
- RichEdit, 678
- Standard-Symbolleiste, 684
- Symbole hinzufügen/entfernen, 671
- Tabellen-Symbolleiste, 674
- verschieben, 32
- zurücksetzen, 671

**Symbolleisten und Fenster wiederherstellen, 829****T****Tabelle,**

- Eigenschaften bearbeiten, 791
- im SPS löschen, 784
- in der Authentic-Ansicht bearbeiten, 808
- Kopf- und Fußzeilen, 783

Kopf- und Fußzeilen hinzufügen, 785

navigieren, 783

Spalte anhängen, 786

Spalte einfügen, 786

Spalte löschen, 787

statische Tabelle einfügen, 784

Umrandung in StyleVision ein oder ausblenden, 790

vertikale Ausrichtung von Zelleninhalt, 793

Zeile anhängen, 786

Zeile einfügen, 786

Zeile löschen, 787

Zelleninhalt, 783

**Tabellen,**

- dynamische (SPS)-Tabellen (SPS) bearbeiten, 571
- dynamische Tabellen erstellen, 148
- erstellen, 753
- leere Spalten ausblenden, 159
- Schließen-Schaltfläche zum Ausblenden von Spalten, 159
- statische Tabellen erstellen, 146
- Stile für aufeinanderfolgende Zeilen, 393
- Übersicht, 144
- vom HTML-Dokument importierte, 487
- Zeilen erweitern/reduzieren, 159
- Zellen verbinden, 788

**Tabellen (SPS),**

- Eigenschaften bearbeiten, 674
- Symbolleisten-Schaltflächen zum Bearbeiten, 674

**Tabellen in der Authentic-Ansicht,**

- Arbeiten mit (statischen und dynamischen) SPS Tabellen, 586
- Arbeiten mit XML-Tabellen, 588
- Symbole zum Bearbeiten von XML-Tabellen, 592
- Verwendung, 585

**Tabellen in der Design-Ansicht,**

- Darstellung, 153
- Einschließen in Vorlagen und Entfernen der Vorlagen, 153

**Tabellenzelle teilen,**

- horizontal, 789
- vertikal, 789

**Tabellenzelleninhalt ausrichten,**

- in SPSs, 793

**Tablellen,**

- formatieren, 155
- Kopf- und Fußzeilen im PDF, 155

**Tags,**

- erweitern und reduzieren, 725

**Tastaturkürzel,**

- für Befehle anpassen, 824

**Technische Informationen, 1172**

**Technischer Support für StyleVision, 837**

**Template,**

changing the node match for, 206

enclosing with, 772

**Templates,**

switching view on and off, 739

**Temporäres Ausgabedokument, 28**

**terminate, 906**

**Terminologie,**

in StyleVision, 24

**Text,**

in der Authentic-Ansicht bearbeiten, 580

in der Authentic-Ansicht formatieren, 580

**Text state icon, 796**

**Textansicht-Events, 860**

**Textreferenzen, 348**

**Textstatus-Symbole, 405**

**Timeout,**

iSeries - muss deaktiviert werden, 505

**Toolbar buttons,**

adding and removing, 672

**Typbasierte Vorlagen, 243**

**Typen als Verarbeitungseinheiten,**

in globalen Vorlagen, 243

## U

**Über StyleVision, 837**

**UCS-2, 1176**

**UCS-4, 1176**

**Umbrechen, 421**

**Umrandung,**

von SPS Tabellen, 790

**Umschlagseiten, 412**

**UNC-Pfade,**

zum Herstellen einer Verbindung zu einer DB, 507

**Unicode,**

Konsortium, 1176

Standard, 1176

Unterstützung, 1177

**Unicode-Zeichen, 1177**

**Universalschriftart, 1177**

**URL zu Projekt hinzufügen, 730**

**User-Defined Templates, 137**

**UTF-8, 1176**

## V

**Validator,**

in Altova-Produkten, 1174

**Variable template,**

inserting, 764

**Variablen, 311, 318**

Zuweisen von Werten über Authentic, 446

**Variablenvorlage, 251**

einschließen in, 774

**Verfahren zur Bearbeitung von Inhalt, 128**

**Version 2010 neuw Features, 13**

**Version 2011 new features, 11**

**Vertikale Ausrichtung von Tabellenzelleninhalt,**

in SPSs, 793

**Vertikaler Text,**

in Layout-Feldern, 200

in Tabellenzellen, 155

**Vertrieb,**

von Altova Produken, 1183

von Altova Software-Produkten, 1180

von Altova-Software-Produkten, 1181

**Verwendung, 106**

**Visual Studio .Net,**

und StyleVision, 654

und StyleVision Unterschiede, 656

**Von rechts nach links, 1178**

**Vordefinierte Formate,**

Befehl für das Einfügen im Design, 749

beim Einfügen eines Node, 132

Symbol in der Design-Ansicht, 687

**Vordefinierte Formate zuweisen,**

im Schnellstart-Tutorial, 82

**Vordefinierte Formatierungsstrings,**

für Eingabeformatierung, 814

**Vorlage,**

einfügen, 762

**Vorlagen,**

Baumstruktur, 48

Tabellenzeilen und Spalten einschließen in Vorlagen, 153

Vorlagen rund um Tabellenzeilen und -spalten entfernen, 153

**Vorlagen für neue Dokumente,**

einfügen, 262

und Design-Struktur, 263

**Vorlagen für neue Dokumente,**

URLs, 264

**Vorlagen für Nodes,**

siehe Node-Vorlagen, 252

**Vorlagenfilter, 682****Vorrang,**

von Stilen, 52

**VS .NET,**

und StyleVision Integrationspaket, 655

# W

**Warnmeldungen über geänderte Dateien,**

in modularen SPSSs, 231

**Whitespace im XML-Dokument,**

Behandlung durch den Altova XSLT 2.0-Prozessor, 1129

**Whitespace Nodes im XML-Dokument,**

und Behandlung durch den XSLT 1.0-Prozessor, 1126

**Whitespace-Behandlung,**

und XPath 2.0-Funktionen, 1133

**Wiedergabe, 1178****Windows,**

Unterstützung für Altova-Produkte, 1173

**Windows 2000, 1177****Windows NT, 1177****Word 2007 (nur Enterprise Edition), 22, 38****WordML (nur Enterprise Edition), 22, 38****Wörterbücher,**

für SPS Rechtschreibprüfung, 819

# X

**XML, 1176**

ins Design einfügen, 143

**XML Parser,**

Info, 1174

**XML validieren,**

in der Authentic-Ansicht, 803

**XML-Arbeitsdatei, 46, 107**

Definition, 24

drucken, 718

Druckvorschau, 718

in Authentic-Ansicht validieren, 803

und Ausgabeansichten, 38

und Authentic-Ansicht, 36

**XML-Authentic-Daten speichern (Befehl), 711****XMLData,**

AppendChild, 987

EraseAllChildren, 988

EraseCurrentChild, 989

GetChild, 989

GetChildKind, 990

GetCurrentChild, 990

GetFirstChild, 991

GetNextChild, 991

HasChildren, 993

HasChildrenKind, 993

InsertChild, 993

IsSameNode, 994

Kind, 995

MayHaveChildren, 995

Name, 995

Parent, 995

TextValue, 996

**XML-Datei,**

mit Daten von DB, 550

**XML-Datei Global Resources, 428****XML-Daten,**

aus mehreren Quellen zusammenführen, 223

in SPS Design einfügen, 129

**XML-Daten für datenbankbasierte SPSSs, 527, 534****XML-Datenbank,**

Neue Datenzeilen in Authentic-Ansicht laden, 804

**XML-DB,**

neue XML-Datenzeile laden, 596

**XML-Dokument-Inhalt,**

Symbol in der Design-Ansicht, 687

**XML-Dokument-Nodes,**

Symbol in der Design-Ansicht, 687

**XML-Schema-Datentypen,**

bei der Generierung von XML-Schema anhand einer ADO DB, 1170

bei der Generierung von XML-Schema anhand einer MS Access DB, 1165

bei der Generierung von XML-Schema anhand einer MS SQL Server DB, 1166

bei der Generierung von XML-Schema anhand einer MySQL DB, 1167

bei der Generierung von XML-Schema anhand einer ODBC DB, 1169

bei der Generierung von XML-Schema anhand einer Oracle DB, 1168

**XML-Schema-Datentypen,**

bei der Generierung von XML-Schema anhand einer Sybase DB, 1171

**XML-Schemas und DTDs,**

als SPS Quelle, 215

**XMLSpy, 29****XMLSpy API,**

Dokumentation, 874

**XMLSpyLib, 874**

Application, 903

AuthenticRange, 922

AuthenticView, 950

Document, 968

Documents, 976

XMLData, 986

**XML-Tabellen (Enterprise und Professional Edition), 144****XML-Tabellen in der Authentic-Ansicht,**

Editiersymbole, 592

Verwendung, 588

**XML-Vorlagendatei (Enterprise und Professional Edition), 107**

Definition, 24

**XPath,**

Nodes in mehreren Dokumenten identifizieren, 317

**XPath 1.0,**

und Datumswerte, 470

**XPath 2.0,**

und Datumswerte, 470

**XPath 2.0 Functions,**

implementation information, 1132

**XPath 2.0-Funktionen,**

allgemeine Informationen, 1133

spezifische Funktionen siehe unter fn:, 1133

**XPath zum ausgewählten Node, 570****XPath-Ausdrücke,**

und Stile, 393

**XPath-Filter,**

an globalen Vorlagen, 243

**XPath-Filter an Node-Vorlagen, 252****XPath-Funktionen,**

benutzerdefiniert, 454

im Dialogfeld "XPath", 691

**XPath-Funktionen - Unterstützung,**

für einzelne Funktionen siehe unter fn:, 1135

**XPath-Operationen,**

im Dialogfeld "XPath", 691

**XPath-Version im SPS, 109****XQuery,**

Erweiterungsfunktionen, 1139

**XQuery 1.0 Functions,**

implementation information, 1132

**XQuery 1.0-Funktionen,**

allgemeine Informationen, 1133

spezifische Funktionen siehe unter fn:, 1133

**XQuery-Prozessor,**

in Altova-Produkten, 1175

**xs:date,**

und Datumswahl, 471

**xs:dateTime,**

und Datumswahl, 471

**xs:QName,**

siehe auch QName, 1135

**xsl:preserve-space, 1126****xsl:strip-space, 1126****XSLT,**

Codefragmente ins Design einfügen, 143

Erweiterungsfunktionen, 1139

Prozessorinformationen, 1125

**XSLT 1.0-Prozessor,**

Einschränkungen und implementierungsspezifisches Verhalten, 1126

**XSLT 2.0 Stylesheet,**

Namespace-Deklarationen, 1129

**XSLT 2.0-Funktionen,**

implementierungsspezifisches Verhalten, 1131

spezifische Funktionen siehe unter fn:, 1131

**XSLT 2.0-Prozessor,**

allgemeine Informationen, 1129

Informationen, 1128

**XSLT in SPS, 451****XSLT Stylesheet-Vorschau,**

in der Ausgabeansicht, 38

**XSLT-Dateien,**

über die Befehlszeile anhand eines SPS generieren, 628

**XSLT-Elemente,**

als Code ins Design einfügen, 141

**XSLT-Import, 451****XSLT-Prozessoren,**

in Altova-Produkten, 1175

**XSLT-Transformationen, 636, 637****XSLT-Version,**

für das SPS konfigurieren, 684

**XSLT-Version im SPS, 109****XSLT-Vorlagen, 48**

in der Seitenleiste Design-Übersicht verwalten, 43

**XSLT-Vorlagen, 48**

ins SPS importieren, 259

# Z

**Zahlen,**

Formatierung, 367

**Zeichen-Entities,**

in der HTML-Ausgabe von XSLT-Transformationen, 1126

**Zeile,**

an SPS Tabelle anhängen, 786

aus Tabelle im SPS löschen, 787

in SPS Tabelle einfügen, 786

**Zeilen (einer Tabelle),**

in der HTML-Ausgabe erweitern/reduzieren, 159

**Zeilenschaltung,**

siehe Eingabetaste, 610

**Zelle (einer Tabelle),**

horizontal teilen, 789

vertikal teilen, 789

**Zellen,**

in SPS Tabellen verbinden, 788

**Zellen verbinden,**

in SPS Tabellen, 788

**Zuletzt verwendete Dateien, 719****Zusammenhalten, 421****Zusätzliche Bearbeitungsverfahren, 426****Zusätzliche Validierung, 449****Zwischensummen,**

in Kopf- und Fußzeilen, 418