

Tutorial



Copyright © 1998–2007. Altova GmbH. All rights reserved. Use of this software is governed by and subject to an Altova software license agreement. XMLSpy, MapForce, StyleVision, SemanticWorks, SchemaAgent, UModel, DatabaseSpy, DiffDog, Authentic, AltovaXML, MissionKit, and ALTOVA as well as their logos are trademarks and/or registered trademarks of Altova GmbH.

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

ALTOVA®

Altova XMLSpy 2009 Tutorial

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: 2008

© 2008 Altova GmbH

Inhaltsverzeichnis

1	XMLSpy-Benutzeroberfläche	2
2	Erstellen eines einfachen XML-Schemas	3
2.1	Anlegen einer neuen XML-Schema-Datei	4
2.2	Definieren von Namespaces	7
2.3	Definieren eines Content Model	9
2.4	Hinzufügen von Elementen mit Drag & Drop	14
2.5	Konfigurieren der Content Model-Ansicht	16
2.6	Fertigstellen des Schemas	18
3	XML-Schemas: komplexere Vorgänge	21
3.1	Arbeiten mit ComplexTypes und SimpleTypes	22
3.2	Referenzieren von globalen Elementen	30
3.3	Attribute und Attribut-Enumerationen	32
4	XML-Schemas: XMLSpy-Funktionen	35
4.1	Schemanavigation	36
4.2	Schema-Dokumentation	39
5	XML Dokumente	43
5.1	Erstellen eines neuen XML-Dokuments	44
5.2	Festlegen des Typs eines Elements	46
5.3	Eingabe von Daten in der Grid-Ansicht	48
5.4	Eingabe von Daten in der Textansicht	49
5.5	Validierung des Dokuments	53
5.6	Hinzufügen von Elementen und Attributen	57
5.7	Editieren in der Datenbank/Tabellen-Ansicht	59
5.8	Ändern des Schemas	63
6	XSLT-Transformationen	65
6.1	Zuweisen einer XSLT-Datei	66
6.2	Transformieren der XML-Datei	67
6.3	Ändern der XSL-Datei	68
7	Projektverwaltung	70
7.1	Vorteile von Projekten	71
7.2	Erstellen eines Projekts	72

8	Fertig!	74
	Index	75

XMLSpy Tutorial

Mit der Ihnen vorliegenden Anleitung erhalten Sie einen umfassenden Einblick in XML. Sie finden darin Anleitungen zu den wichtigsten Aufgaben im Zusammenhang mit XML. Schritt für Schritt werden Sie mit einigen der wichtigsten Funktionen von XMLSpy vertraut.

Das Tutorial besteht aus den folgenden Teilen:

- [Erstellen eines XML-Schemas](#). Sie lernen, wie man in der intuitiven Schema Design-Ansicht von XMLSpy ein XML-Schema erstellt, wie man mit Drag-and-Drop komplexe Inhaltsmodelle erstellt und wie man die Schema Design-Ansicht konfiguriert. Enthält eine Einführung in die Arbeit mit XML-Schemas und den verschiedenen Ansichten, die in XMLSpy zum Anzeigen und Bearbeiten von XML-Schemas zur Verfügung stehen.
- Erstellen von complex und simpleTypes, globalen Elementreferenzen und Attributenumerationen [mit Hilfe von Funktionen der Schema Design-Ansicht](#).
- Sie lernen, wie man in der Schema Design-Ansicht in [Schemas navigiert](#) und wie man [Dokumentation zu Schemas generiert](#).
- [Erstellen eines XML-Dokuments](#). Sie lernen, wie man einem XML-Dokument ein Schema zuweist, wie man ein XML-Dokument in der Grid- und der Text-Ansicht bearbeitet und XML-Dokumente mit Hilfe des integrierten Validators von XMLSpy validiert.
- [Transformieren einer XML-Datei mittels XSLT-Stylesheets](#). Dabei müssen Sie eine XSLT-Datei zuweisen und die Transformation mit Hilfe des in XMLSpy integrierten XSLT-Prozessors durchführen.
- [Arbeiten mit XMLSpy Projekten](#), mit denen Sie Ihre XML-Dokumente ganz einfach organisieren können.

Installation und Konfiguration

In dieser Anleitung gehen wir davon aus, dass Sie XMLSpy erfolgreich auf Ihrem PC installiert haben, dass Sie einen kostenlosen Evaluierungs-Key erhalten haben oder dass Sie sich als User bereits registriert haben. Die Evaluierungsversion von XMLSpy ist voll funktionsfähig. Allerdings ist diese auf dreißig Tage begrenzt. Sie können danach eine reguläre Lizenz über unseren sicheren Webserver oder über einen Vertriebspartner erwerben.

Anleitung: Beispieldateien

Beispieldateien, die in der vorliegenden Anleitung angesprochen werden, finden Sie im Applikationsordner unter:

```
C:\Documents and Settings\\My Documents\Altova\XMLSpy\  
Examples\Tutorial
```

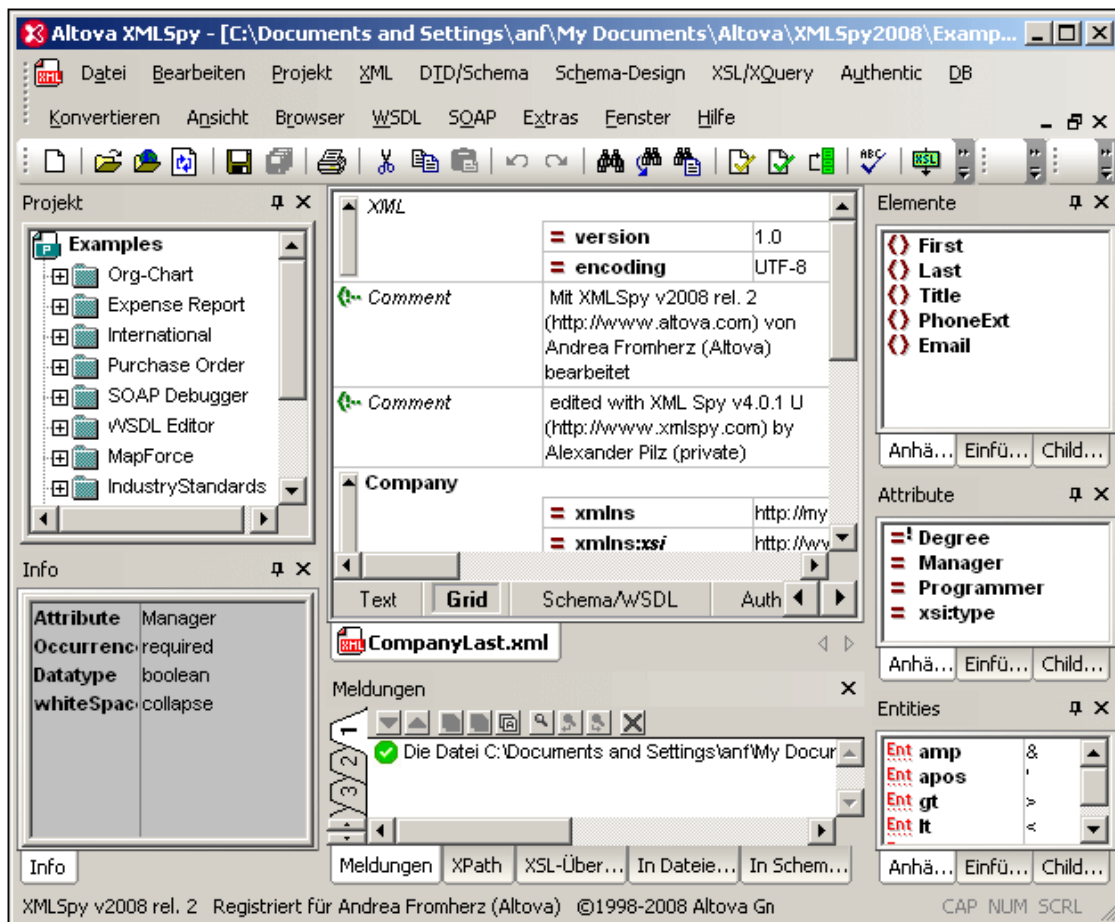
Im Verzeichnis `Examples` finden Sie viele XML-Dateien, mit denen Sie experimentieren können. Im Verzeichnis `Tutorial` sind alle Dateien abgelegt, die in dieser Anleitung verwendet werden.

Das Verzeichnis `Template` (**Vorlagen**) im Applikationsordner enthält sämtliche XML-Vorlagen, die verwendet werden, wenn Sie die Menüoption **Datei | Neu** aufrufen. Diese Dateien enthalten alle nötigen Daten (Namespace und XML-Deklarationen) die Sie benötigen, um mit dem XML-Dokument arbeiten zu können.

1 XMLSpy-Benutzeroberfläche

Die Oberfläche von XMLSpy ist vertikal in drei Bereiche gegliedert. Im mittleren Bereich stehen verschiedene Ansichten für Ihr Dokument bereit. Die beiden seitlichen Bereiche enthalten Informationen, Editierhilfen und Funktionen zur Dateiverwaltung.

- Das **Projekt-** und das **Info-Fenster** befinden sich auf der linken Seite.
- Der mittlere Teil besteht aus dem **Hauptfenster**. Hier bearbeiten Sie Ihr XML-Dokument. Sie können zwischen verschiedenen Ansichten wählen: Textansicht, Grid-Ansicht, Schema/WSDL Design-Ansicht, Authentic-Ansicht und Browser-Ansicht. Diese Ansichten werden in den entsprechenden Abschnitten dazu näher beschrieben.
- Auf der rechten Seite des Bildschirms befinden sich drei **Eingabehilfen-Fenster**. Hier können Sie Elemente, Attribute oder Entities einfügen oder anhängen. Die verfügbaren Einträge ändern sich je nachdem, welche Auswahl in der Grid-Ansicht vorgenommen wurde, bzw. je nach Positionierung des Cursors in der XML-Datei.



Die Einzelheiten der Benutzeroberfläche werden im Laufe des Handbuchs näher beschrieben. Beachten Sie, dass die Oberfläche sich dynamisch je nach gewählter Ansicht ändert und je nachdem, welches Dokument im Hauptfenster aktiv ist.

2 Erstellen eines einfachen XML-Schemas

Ein XML-Schema beschreibt die Struktur eines XML-Dokuments. Ein XML-Dokument kann gegen ein XML-Schema validiert werden, um zu überprüfen, ob es mit den im Schema definierten Vorgaben übereinstimmt und somit gültig ist. Falls dies nicht der Fall ist, ist es ungültig. Mit Hilfe von Schemas kann der Verfasser eines Dokuments die Struktur und den zulässigen Inhalt eines XML-Dokuments definieren und überprüfen, ob das XML-Dokument gültig ist.

Ein XML-Schema hat eine komplexe Struktur und Syntax. Da es sich bei einem XML-Schema selbst um ein XML-Dokument handelt, muss es den Regeln der XML-Schema-Spezifikation entsprechen. In XMLSpy können Sie in der Schema/WSDL Design-Ansicht mit Hilfe von Drag & Drop-Techniken gültige XML-Schemas auf einfache Art erstellen. Das von Ihnen erzeugte XML-Schema-Dokument kann auch in der Text- und der Grid-Ansicht bearbeitet werden, viel einfacher lässt sich dies jedoch in der Schema/WSDL-Ansicht bewerkstelligen.

Ziel

In diesem Abschnitt lernen Sie, wie man ein XML-Schemas in der Schema/WSDL Design-Ansicht bearbeitet. Insbesondere wird auf folgende Punkte eingegangen:

- Erstellen einer neuen Schema-Datei
- Definieren von Namespaces für das Schema
- Definieren eines einfachen Content Model
- Hinzufügen von Elementen zum Content Model über Kontextmenüs und mittels Drag & Drop
- Konfiguration der Content Model-Ansicht

Nachdem Sie mit der Erstellung des Schema-Grundgerüsts fertig sind, können Sie zum [nächsten Abschnitt des Tutorials](#) übergehen, in dem Sie erfahren, wie man mit komplexeren XML-Schema-Funktionen in XMLSpy arbeitet. Auf diesen Abschnitt folgt ein Abschnitt über [Schema-Navigation und Dokumentation](#) in XMLSpy.

In diesem Abschnitt verwendete Befehle

In diesem Abschnitt arbeiten Sie ausschließlich in der Schema/WSDL-Ansicht. Dabei werden die folgenden Befehle verwendet:



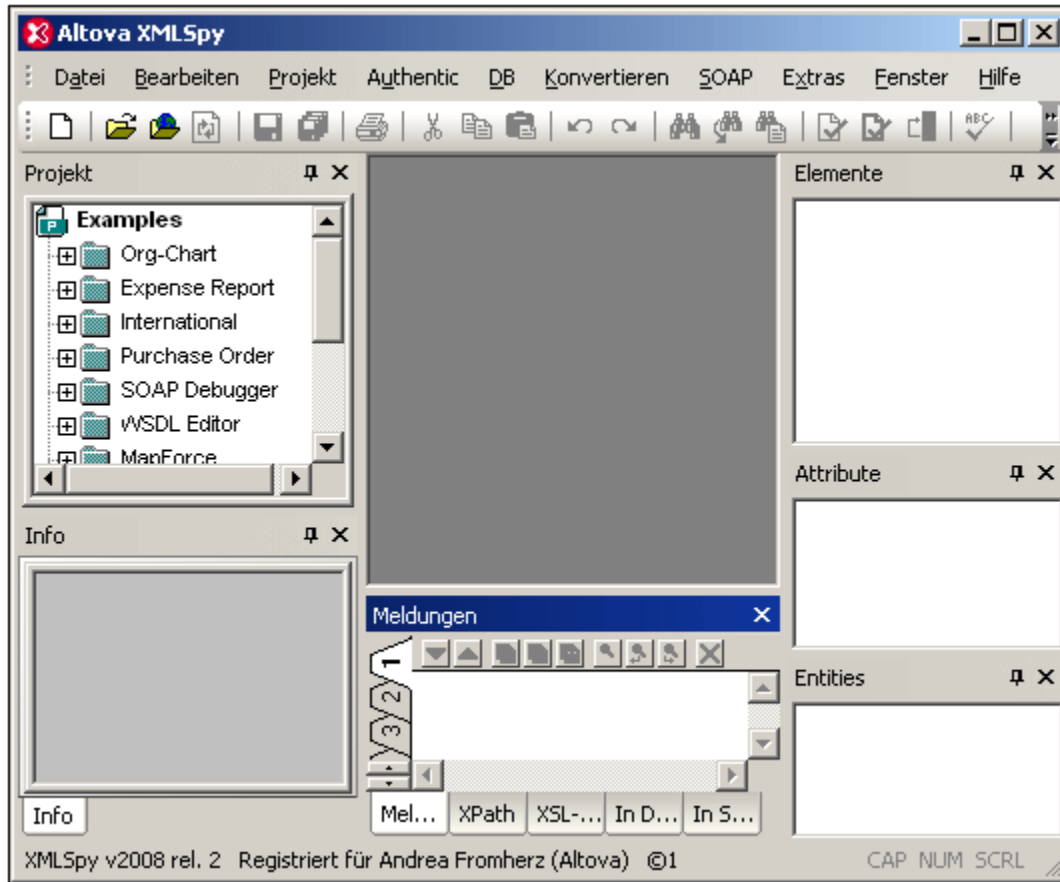
Diagramm anzeigen (oder Content Model-Ansicht anzeigen). Dieses Symbol befindet sich links von allen globalen Komponenten in der Schema-Übersicht. Wenn Sie auf das Symbol klicken, wird das Content Model der dazugehörigen globalen Komponente angezeigt.

2.1 Anlegen einer neuen XML-Schema-Datei

Um in XMLSpy eine neue XML-Schema-Datei anzulegen, müssen Sie zuerst XMLSpy starten und dann ein neues XML-Schema (.xsd)-Dokument erstellen.

Starten von XMLSpy

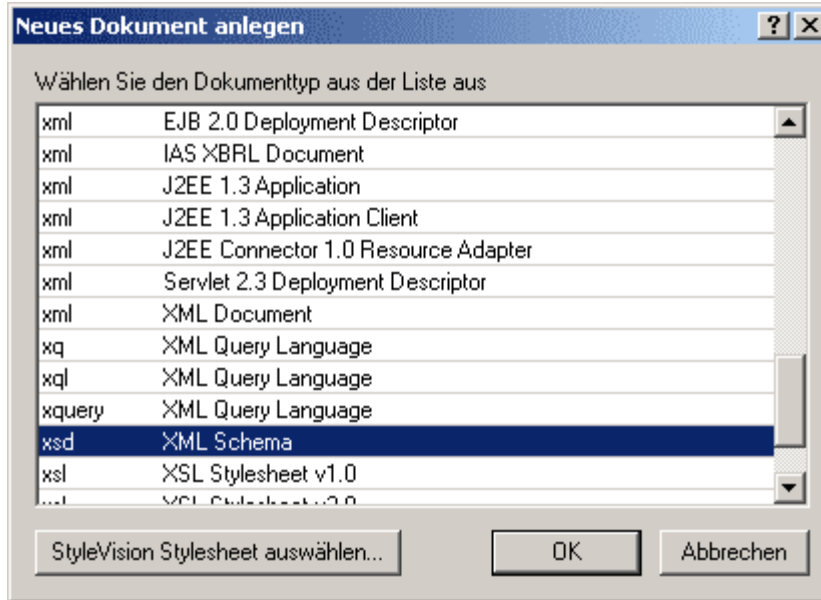
Doppelklicken Sie zum Starten von XMLSpy auf das XMLSpy-Symbol auf Ihrem Desktop oder rufen Sie XMLSpy über **Start | Programme** auf. XMLSpy wird mit einem leeren Hauptfenster gestartet, in dem kein Dokument geöffnet ist.



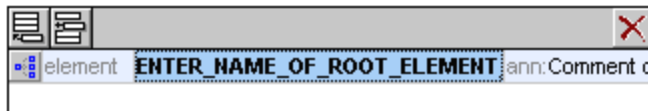
Beachten Sie, dass sich die Benutzeroberfläche in vier Teile gliedert: (i) das Projekt- und das Info-Fenster auf der linken Seite; (ii) das Hauptfenster in der Mitte; (iii) die Eingabehilfen auf der rechten Seite und (iv) die Ausgabefenster am unteren Rand.

So legen Sie eine neue Schema-Datei an:

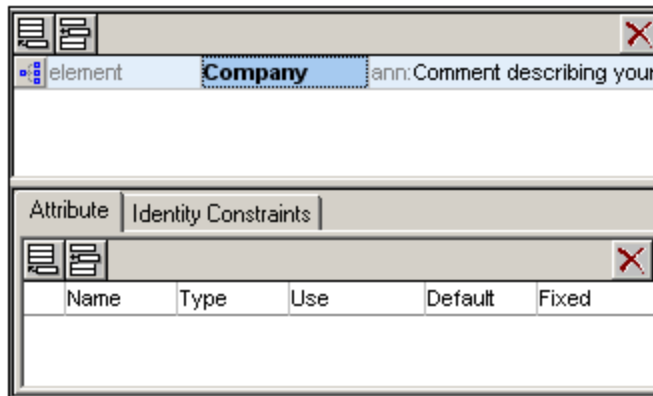
1. Wählen Sie die Menüoption **Datei | Neu**. Das Dialogfeld "Neues Dokument anlegen" wird angezeigt.



2. Wählen Sie im Dialogfeld den Eintrag `xsd W3C Schema` (die Dokumentbeschreibung und die Liste im Fenster sind unter Umständen nicht mit denen in der Abbildung identisch) Bestätigen Sie mit OK. Eine leere Schema-Datei wird im Hauptfenster in der Schema/WSDL Design-Ansicht angezeigt. Sie werden aufgefordert, den Namen des Root-Elements anzugeben.



3. Klicken Sie in das markierte Feld, und geben Sie `Company` ein. Bestätigen Sie mit der Eingabetaste. `Company` ist somit das "Root" Element dieses Schemas und gleichzeitig ein globales Element. Die im Hauptfenster angezeigte Ansicht (siehe Abbildung unten) nennt sich Schema-Übersicht und enthält eine Liste aller globalen Komponenten im oberen Bereich Fenster; die Attribute und Identity Constraints der ausgewählten Komponente werden im unteren Fenster angezeigt. (Sie können das Content Model einzelner globaler Komponenten ansehen und bearbeiten, indem Sie auf das Symbol "Diagramm anzeigen" klicken, das links von der globalen Komponente zu sehen ist.)



4. Geben Sie im Annotationsfeld (`ann`) des Elements `Company` die Beschreibung für das Element ein, in diesem Fall `Root element`.
5. Klicken Sie auf die Menüoption **Datei | Speichern** und benennen Sie Ihr Schema (z.B. `AddressFirst.xsd`).

2.2 Definieren von Namespaces

XML Namespaces spielen eine wichtige Rolle in XML-Schemas und XML-Dokumenten. Ein XML-Schema muss den XML-Schema-Namespace referenzieren und kann optional auch einen Target Namespace für das XML-Instanzdokument definieren. Als Schema-Autor müssen Sie festlegen, wie Sie diese beiden Namespaces definieren (im Grunde mit Hilfe von Präfixen).

Sie werden nun im XML-Schema, das Sie erstellen, einen Target Namespace für das XML-Instanzdokument definieren. (Die erforderliche Referenz zum XML Schema Namespace wird von XMLSpy automatisch erstellt, wenn Sie ein neues XML Schema-Dokument anlegen.) So erstellen Sie einen Target Namespace:

1. Wählen Sie die Menüoption **Schema-Design | Schema-Einstellungen**, um das Dialogfeld "Schema-Einstellungen" aufzurufen.

Prefix	Namespace
	http://my-company.com/namespace
xs	http://www.w3.org/2001/XMLSchema

2. Klicken Sie auf das Target Namespace-Optionsfeld und geben Sie `http://my-company.com/namespace` ein. In XMLSpy wird der von Ihnen als Target Namespace eingegebene Namespace als Default Namespace des XML-Schema-Dokuments erstellt und in der Liste der Namespaces im unteren Bereich des Dialogfelds angezeigt.
3. Bestätigen Sie mit OK.


Bitte beachten Sie:

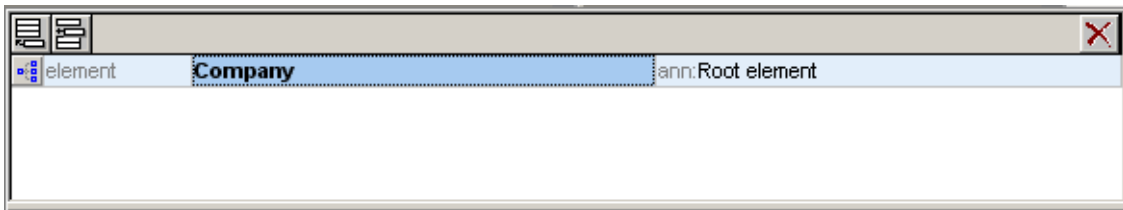
- Der XML Schema Namespace wird von XMLSpy automatisch erstellt und erhält das Präfix `xs`.

- Wenn Sie das XML-Instanzdokument erstellen, muss es den im XML-Schema für das XML-Dokument definierten Target Namespace haben, um gültig zu sein.

2.3 Definieren eines Content Model

Sie haben in der Schema-Übersicht bereits ein globales Element mit dem Namen `Company` erstellt. Dieses Element soll ein `Address`-Element und eine unbegrenzte Anzahl an `Person`-Elementen enthalten, nämlich das Content Model. Globale Elemente, die Content Models haben können, sind Elemente, `complexType`s und Elementgruppen.


In XMLSpy wird das Content Model einer globalen Komponente in der Content Model-Ansicht der Schema/WSDL-Ansicht angezeigt. Um das Content Model einer globalen Komponente anzeigen und editieren zu können, klicken Sie auf das Symbol "Diagramm anzeigen"  links neben der globalen Komponente.

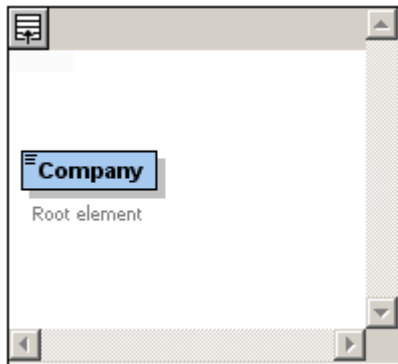


In diesem Abschnitt werden Sie das Content Model des Elements `Company` erstellen.

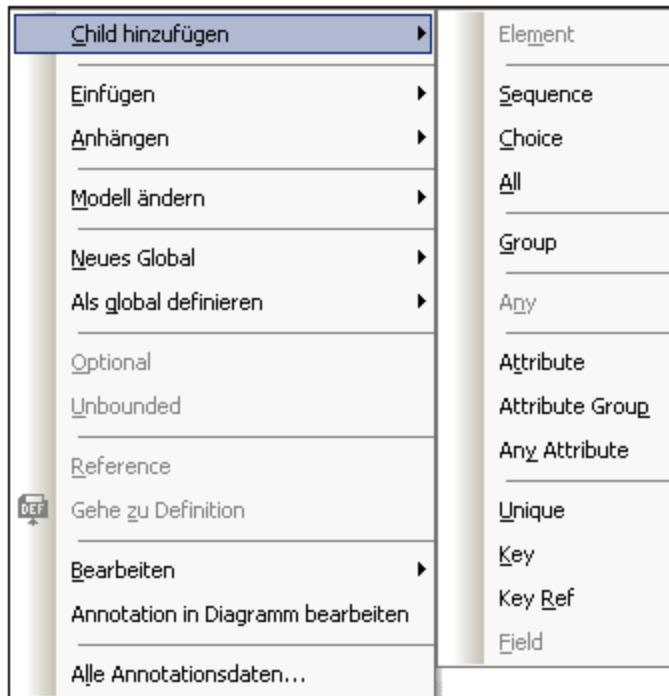
Erstellen eines einfachen Content Model

So erstellen Sie das Content Model des Elements `Company`:

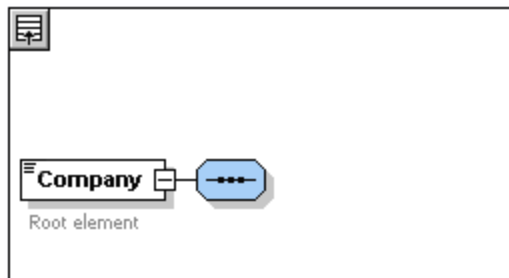
1. Klicken Sie in der Schema-Übersicht auf das Symbol "Diagramm anzeigen"  links neben dem Element `Company`. Hierdurch wird das Content Model des Elements `Company` angezeigt, das derzeit noch leer ist. (Alternativ dazu können Sie auch auf den Eintrag `Company` im Komponentenfenster doppelklicken, um das Content Model anzuzeigen.)



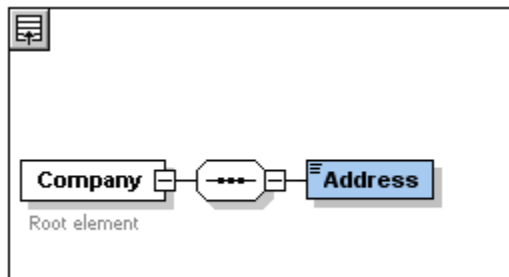
2. Ein Content Model besteht aus Kompositoren und Komponenten. Die Kompositoren definieren die Beziehung zwischen zwei Komponenten. Sie müssen nun im `Company` Content Model einen Child-Kompositor zum Element `Company` hinzufügen, um ein Child-Element hinzufügen zu können. Rechtsklicken Sie dazu auf das Element "Company" und wählen Sie im Kontextmenü die Option **Child hinzufügen | Sequence**. (Sequence, Choice und All sind die drei Kompositoren, die in einem Content Model verwendet werden können.)



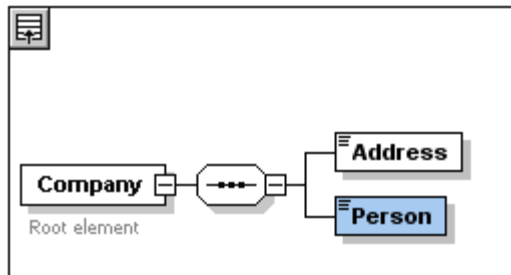
Daraufhin wird der Sequence-Kompositor eingefügt, der definiert, dass die Komponenten, die darauf folgen in der angegebenen Reihenfolge erscheinen müssen.



3. Rechtsklicken Sie auf den Sequence-Kompositor und wählen Sie **Child hinzufügen | Element**. Daraufhin wird ein unbenanntes Element hinzugefügt.
4. Benennen Sie das Feld `address` und bestätigen Sie mit der Eingabetaste.

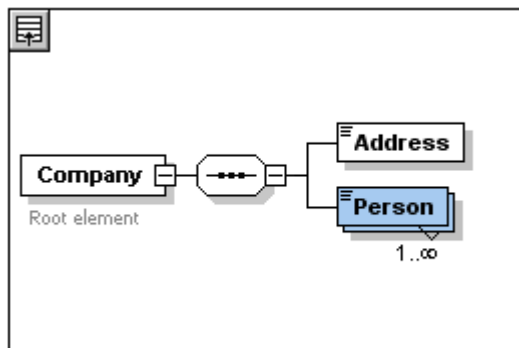


5. Rechtsklicken Sie erneut auf den Sequence-Kompositor und wählen Sie **Child hinzufügen | Element**, benennen Sie die neue Elementkomponente `person`.



Sie haben nun ein Schema mit einem Eintrag für die Firmenadresse und einem Eintrag für einen Angestellten (`person`) definiert. Wir müssen nun die Anzahl der `person`-Elemente erhöhen.

6. Klicken Sie mit der rechten Maustaste auf das Element `person` und wählen Sie **Unbounded** aus dem Kontextmenü. Unter dem Element `person` wird nun die Anzahl der zulässigen Instanzen angezeigt: 1 bis unendlich.

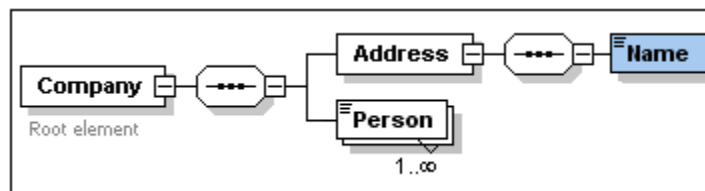


Sie können alternativ dazu in der Details-Eingabehilfe die Felder `minOcc` und `maxOcc` direkt editieren und die Anzahl der zulässigen Instanzen angeben, in diesem Fall 1 bzw. unbounded.

Hinzufügen von zusätzlichen Ebenen zur Struktur des Content Model

Bisher enthält das Content Model, das Sie erstellt haben, eine Ebene: eine Child-Ebene für das Element `company`, das die Elemente `Address` und `person` enthält. Nun wollen wir den Inhalt des Elements `Address` definieren, und zwar die Subelemente `Name`, `Street` und `City`. Dies ist eine zweite Ebene. Wir müssen also zum Element `Address` einen Child-Kompositor hinzufügen und anschließend die Elementkomponenten selbst. Gehen Sie dabei folgendermaßen vor:

1. Klicken Sie mit der rechten Maustaste auf das Element `Address`, um das Kontextmenü zu öffnen, und wählen Sie **Child hinzufügen | Sequence**.
2. Klicken Sie mit der rechten Maustaste auf den **Sequence**-Kompositor und wählen Sie **Child hinzufügen | Element**. Benennen Sie die neu erstellte Elementkomponente mit "Name".



ComplexTypes, simpleTypes und XML Schema-Datentypen

Bisher haben wir noch keinen Elementtyp explizit definiert. Klicken Sie auf das Register **Text**

um die Textansicht für Ihr Schema anzuzeigen (*siehe Liste unten*). Sie werden feststellen, dass jedes Mal, wenn ein Sequence-Kompositor eingefügt wurde, das Element `xs:sequence` im Element `xs:complexType` eingefügt wurde. Kurz gesagt, die Elemente `Company` und `Address` sind ComplexTypes, da sie Child-Elemente enthalten. Ein ComplexType ist ein Element, das Attribute oder Elemente enthält.

```
<xs:element name="Company">
  <xs:annotation>
    <xs:documentation>Root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Address">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Person" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

SimpleType-Elemente dagegen enthalten nur Text und haben keine Attribute. Beim Text kann es sich um Strings, Daten, Zahlen usw. handeln. Wir möchten, dass das Child-Element `Name` von `Address` nur Text enthält. Es ist ein SimpleType, dessen Textinhalt wir durch Restriktion auf einen String einschränken möchten. Dies lässt sich durch Verwendung des XML-Schema-Datentyps `xs:string` bewerkstelligen. Um das Element `Name` als diesen Datentyp zu definieren, gehen Sie folgendermaßen vor.

1. Klicken Sie auf das Register **Schema/WSDL** um zur Schema/WSDL-Ansicht zurückzukehren.
2. Klicken Sie auf `Name`, um das Element auszuwählen.
3. Klicken Sie in der Details-Eingabehilfe auf das Listenfeld `type` in der mittleren Eingabehilfe und wählen Sie `xs:string` aus der Auswahlliste.

Details	
name	Name
isRef	<input type="checkbox"/>
minOcc	1
maxOcc	1
type	xs:string
content	xs:normalizedE
derivedBy	xs:NOTATION
default	xs:positiveInte
fixed	xs:QName
nullable	xs:short
block	xs:string
form	xs:time

Beachten Sie, dass sowohl `minOcc` als auch `maxOcc` den Wert 1 hat, d.h. dieses Element hat nur eine Instanz.

In der Textdarstellung sieht das Element `Name` folgendermaßen aus:

```
<xs:element name="Name" type="xs:string" />
```

Bitte beachten Sie: Ein einfaches Type-Element kann jedem beliebigen von mehreren

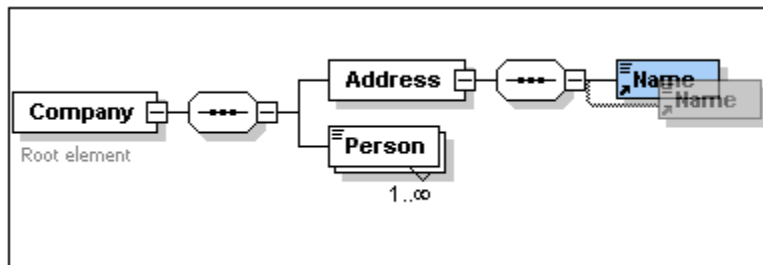
XML-Schema-Datentypen angehören. In all diesen Fällen erscheint das Symbol für Textinhalt nur im Elementfeld.

2.4 Hinzufügen von Elementen mit Drag & Drop

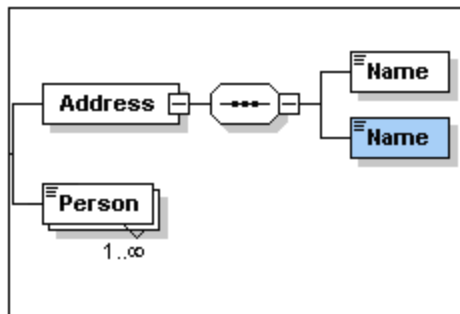
Sie haben Elemente bisher über das Kontextmenü hinzugefügt, das erscheint, wenn Sie mit der rechten Maustaste auf ein Element oder einen Kompositor klicken. Schneller lassen sich Elemente aber mit Drag & Drop erstellen. In diesem Abschnitt lernen Sie, wie man mit Drag & Drop mehr Elemente zur Definition des Elements Address hinzufügt und so diese Definition fertig stellt.

So stellen Sie die Definition des Elements Address mit Hilfe von Drag & Drop fertig:

1. Klicken Sie auf das Element Name des Elements Address, halten Sie die Maustaste sowie die STRG-Taste gedrückt und ziehen Sie die Maus leicht nach unten. Die Markierung mit einem +Zeichen erscheint. Hierdurch wird angezeigt, dass das Element kopiert werden kann. Es erscheint eine Kopie des Elements sowie eine Konnektorlinie, sodass Sie sehen, wo das Element erstellt wird.

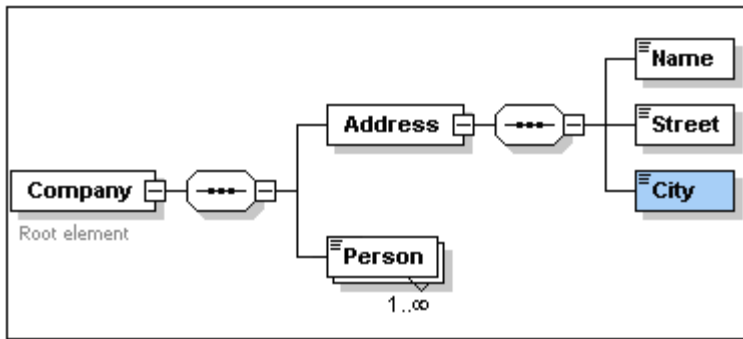


2. Lassen Sie nun die Maustaste und die STRG-Taste los um das neue Element in der Address-Sequenz anzulegen. Wurde das Element an der falschen Stelle angelegt, so ziehen Sie es einfach mit der Maus unter das Element Name.



3. Doppelklicken Sie in das Elementfeld und geben Sie Street ein, um den Namen des Elements zu ändern.
4. Wiederholen Sie obige Schritte um ein weiteres Element zu erzeugen. Benennen Sie dieses Feld um und geben Sie ihm den Namen City.

Das Content Model sollte nun folgenden Inhalt haben:




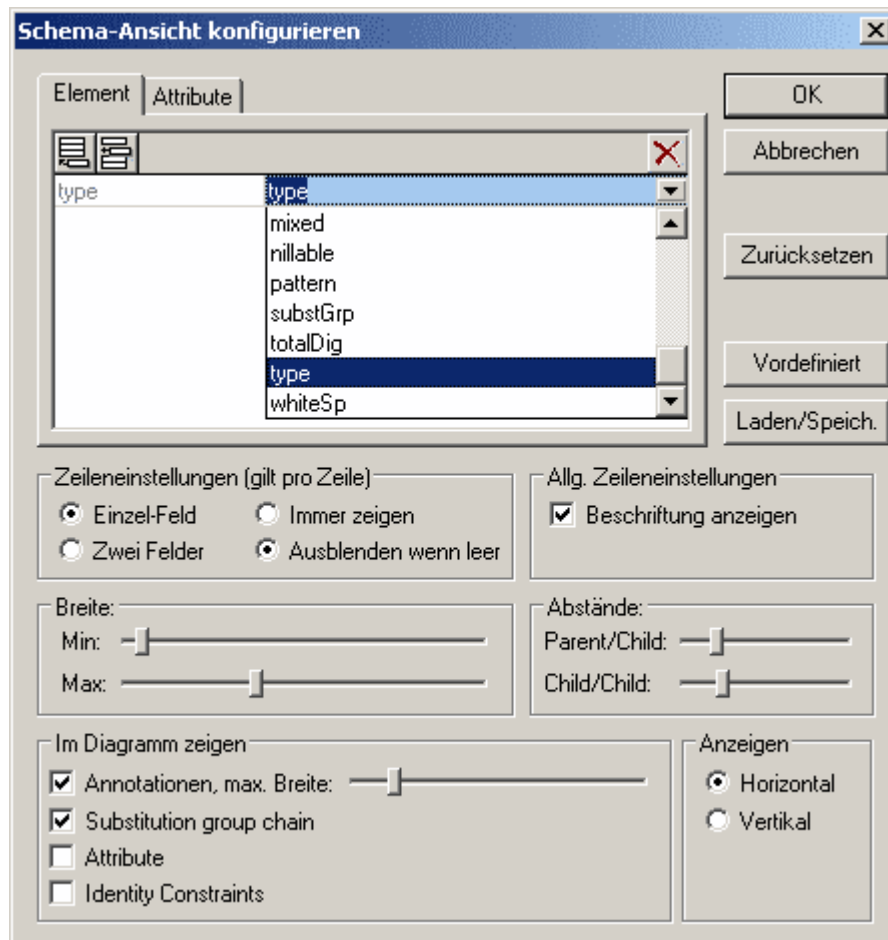
Das Element Address enthält eine Sequence mit einem Element Name, einem Element Street und einem Element City, und zwar in der obigen Reihenfolge.


2.5 Konfigurieren der Content Model-Ansicht

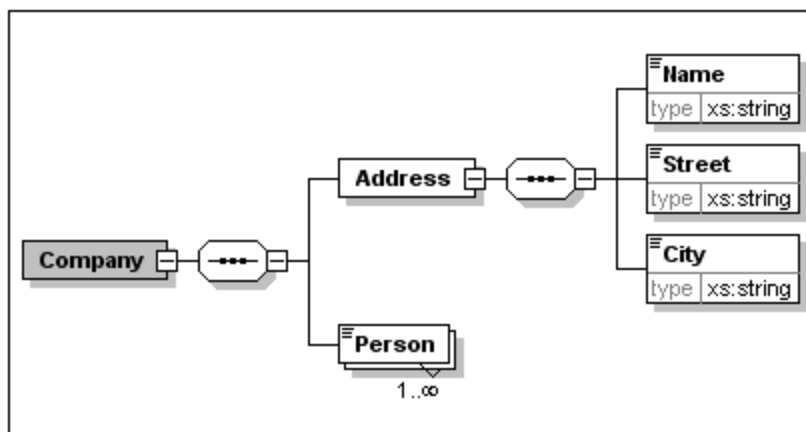
Sie können jetzt das Aussehen der Content Model-Ansicht konfigurieren. Wir wollen die Content Model-Ansicht so konfigurieren, dass der Elementtyp für jedes einzelne Element angezeigt wird.

So konfigurieren Sie die Content Model-Ansicht:

1. Gehen Sie zur Content Model-Ansicht einer Komponente (klicken Sie auf das Symbol Content Model-Ansicht ) , um den Befehl "Ansicht konfigurieren" zu aktivieren.
2. Wählen Sie die Menüoption Schema-Design | Ansicht konfigurieren, um das Dialogfeld "Schema-Ansicht konfigurieren" aufzurufen.

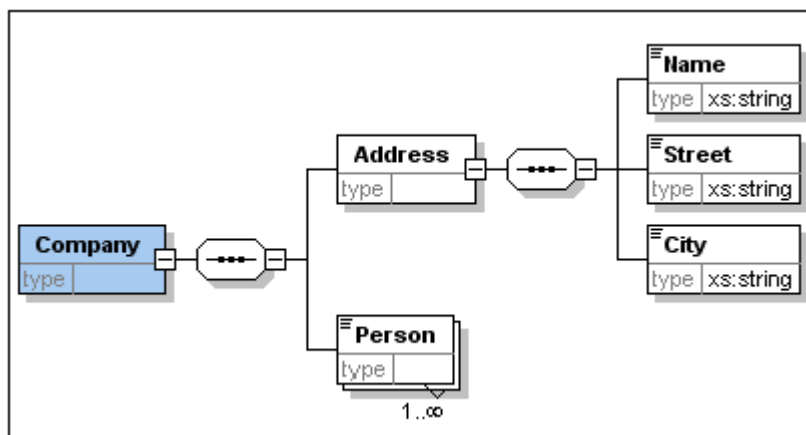


3. Klicken Sie im Element/Attribut-Fenster auf dem Register "Element" auf das Symbol Anhängen  um eine Zeile zur Beschreibung einer Eigenschaft zu jedem Elementfeld hinzuzufügen.
4. Wählen Sie type aus der Auswahlliste (oder doppelklicken Sie in die Zeile und tippen Sie "type"). Dadurch wird der Datentyp jedes Elements in der Content Model-Ansicht angezeigt.
5. Wählen Sie im Bereich "Zeileneinstellungen (Gilt pro Zeile)" die Option "Ausblenden wenn leer", damit die Beschreibung des Datentyps im Elementfeld ausgeblendet wird, wenn das Element keinen Datentyp hat (z.B. wenn das Element ein ComplexType ist)



Beachten Sie, dass die Beschreibungszeile für den Typ bei den Elementen Name, Street und City aufscheint, da es sich dabei um SimpleTypes vom Typ xs:string handelt, nicht aber bei den ComplexType-Elementen, da die Option "Ausblenden wenn leer" aktiv ist.

6. Aktivieren Sie im Bereich "Zeileneinstellungen (Gilt pro Zeile)" die Option "Immer zeigen".
7. Klicken Sie auf OK, um die Änderungen zu speichern.



Beachten Sie, dass die Beschreibungszeile für den Datentyp immer angezeigt wird - selbst in Elementfeldern von ComplexTypes, in denen keine Werte in dieser Zeile angezeigt werden.

Bitte beachten Sie:

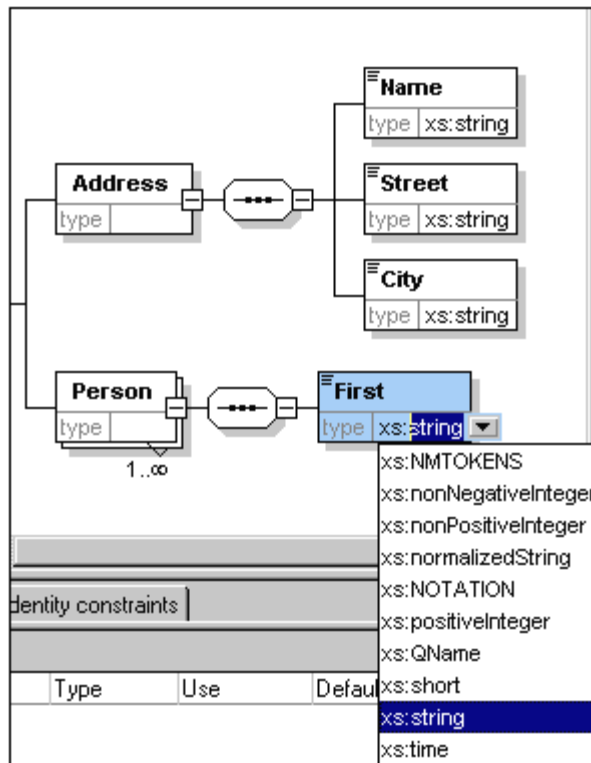
- Die Eigenschaftsbeschreibungszeilen können editiert werden. Werte, die Sie dort eingeben, werden Teil der Elementdefinition.
- Die Einstellungen, die Sie im Dialogfeld "Schema-Ansicht konfigurieren" vornehmen, gelten für die Ausgabe der Schema-Dokumentation ebenso wie für die Druckausgabe.

2.6 Fertigstellen des Schemas

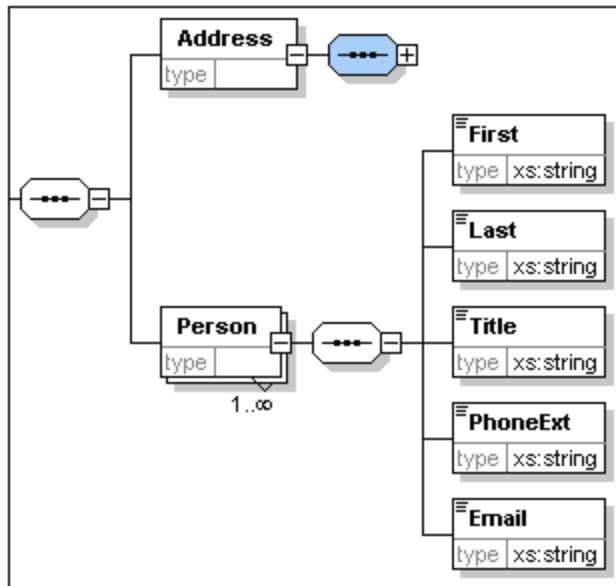
Nachdem Sie nun den Inhalt des Elements Address definiert haben, müssen Sie den Inhalt des Elements Person definieren. Das Element Person soll die folgenden Child-Elemente - alle vom Typ SimpleType - enthalten: First, Last, Title, PhoneExt und Email. Alle diese Elemente sind obligatorisch mit Ausnahme des Elements Title (das optional ist), und alle Elemente müssen in der oben angegebenen Reihenfolge aufscheinen. Alle Elemente sollen vom Datentyp xs:string sein, mit Ausnahme von PhoneExt, das vom Datentyp xs:integer sein muss und maximal 2 Stellen haben darf.

Um das Content Model für Person zu erstellen, gehen Sie folgendermaßen vor:

1. Rechtsklicken Sie auf das Element Person, um das Kontextmenü aufzurufen und wählen Sie Child hinzufügen | Sequence. Der Sequence-Kompositor wird eingefügt.
2. Klicken Sie mit der rechten Maustaste auf den Sequence-Kompositor und wählen Sie Child hinzufügen | Element.
3. Nennen Sie das Element First und betätigen Sie die Tabulatortaste Ihrer Tastatur. Sie gelangen automatisch in das Feld type.



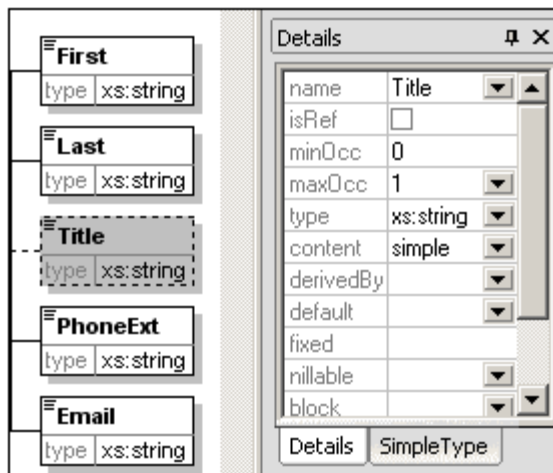
4. Wählen Sie den Eintrag xs:string aus der Auswahlliste oder geben Sie ihn in das Wertefeld von type ein.
5. Erzeugen Sie weitere vier Elemente mittels der Drag & Drop Methode. Benennen Sie diese der Reihe nach Last, Title, PhoneExt und Email.



Bitte beachten Sie: Sie können mehrere Elemente auswählen, indem Sie die Strg-Taste gedrückt halten, während Sie die Elemente anklicken. Auf diese Art können Sie mehrere Elemente gleichzeitig kopieren.

Ein Element optional machen

Rechtsklicken Sie auf das Element Title und wählen Sie Optional im Kontextmenü. Die Umrahmung des Elementes wechselt von durchgezogener Linie auf gestrichelt. Hierdurch wird angezeigt, dass dieses Element optional ist.

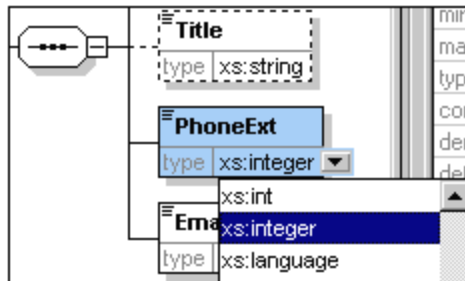


Sie sehen in der Details-Eingabehilfe, dass $\text{minOcc}=0$ und $\text{maxOcc}=1$ ist, d.h. das Element ist optional. Alternativ dazu können Sie das Element auch über das Kontextmenü optional machen und für $\text{minOcc}=0$ einstellen, um das Element zu einem optionalen zu machen.

Den Inhalt eines Elements einschränken

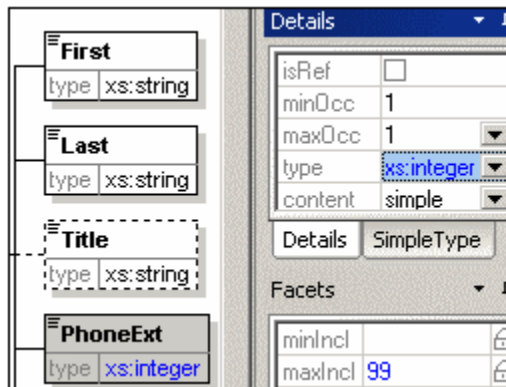
Um das Element PhoneExt als Typ `xs:integer` mit maximal 2 Stellen zu definieren, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in das Feld `type` unter dem Element PhoneExt und wählen Sie den Eintrag `xs:integer` aus der Auswahlliste oder geben Sie ihn ein.



Änderungen werden in der Facets-Eingabehilfe angezeigt.

2. Doppelklicken Sie in der Facets-Eingabehilfe in das Feld maxIncl, geben Sie 99 ein und bestätigen Sie dies durch Drücken der Eingabetaste.



Hierdurch wird definiert, dass alle Durchwahlnummern bis inkl. 99 gültig sind.

3. Wählen Sie die Menüoption Datei | Speichern zum Speichern der Änderungen im Schema.

Bitte beachten Sie:

- Beim Auswählen eines XML-Schema-Datentyps vom Typ simpleType (z.B. xs:string oder xs:date) wird das Content Model automatisch in der Details-Eingabehilfe auf simple (content = simple) gesetzt.
- Beim Hinzufügen eines Kompositors zu einem Element (selection, choice oder all), wird das Inhaltsmodell automatisch auf complex (content = complex) gesetzt.
- Das verwendete Schema ist als AddressFirst.xsd im Verzeichnis C:\Documents and Settings\\My Documents\Altova\XMLSpy\Examples\Tutorial Ihres XMLSpy-Anwendungsordners abgelegt.

3 XML-Schemas: komplexere Vorgänge

Nachdem Sie nun ein einfaches Schema erstellt haben, können wir uns den komplizierteren Aspekten der Schema-Entwicklung widmen.

Ziel

In diesem Abschnitt werden folgende Punkte behandelt:

- Arbeiten mit [complexTypees und simpleTypes](#), die später als die Typen von Schema-Elementen verwendet werden können.
- Erstellen von [globalen Elementen](#) und Referenzieren dieser Elemente von anderen Elementen aus
- Erstellen von [Attributen](#) und deren Eigenschaften, einschließlich Enumerationswerten.

Wir beginnen diesen Abschnitt mit dem einfachen Schema AddressFirst.xsd, das Sie im ersten Teil dieses Tutorials erstellt haben.

In diesem Abschnitt verwendete Befehle

In diesem Abschnitt des Tutorials wird ausschließlich die Schema/WSDL-Ansicht verwendet. Dabei kommen die folgenden Befehle zum Einsatz:



Diagramm anzeigen (oder Content Model-Ansicht anzeigen). Mit diesem Symbol, das in der Schema-Übersicht links von allen globalen Komponenten angezeigt wird, sehen Sie das Content Model der dazugehörigen globalen Komponente



Globale Elemente anzeigen. Dieses Symbol befindet sich in der linken oberen Ecke des Content Model. Mit diesem Befehl kommen Sie zurück zur Schema-Übersicht, in der alle globalen Komponenten angezeigt werden.



Anhängen. Mit diesem Symbol (in der linken oberen Ecke der Schema-Übersicht) können Sie eine globale Komponente hinzufügen.

3.1 Arbeiten mit ComplexTypes und SimpleTypes

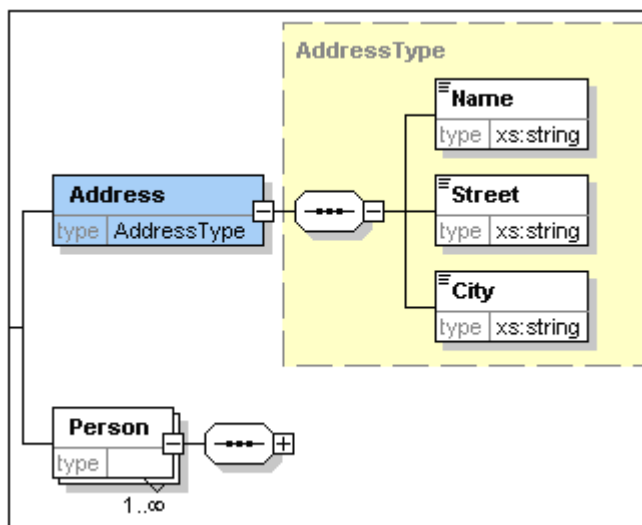
Nachdem Sie ein Element definiert haben, möchten Sie es möglicherweise auch an einer anderen Stelle im Schema verwenden. Dies erreichen Sie dadurch, dass Sie diese Elementdefinition als globalen ComplexType oder als globales Element erstellen. In diesem Abschnitt werden Sie mit globalen ComplexTypes arbeiten. Wir werden zuerst einen ComplexType auf globaler Ebene erstellen und diesen dann erweitern, um ihn in einem Content Model verwenden zu können. Mehr über globale Elemente erfahren Sie weiter hinten in diesem Tutorial.


Erstellen eines globalen complex type

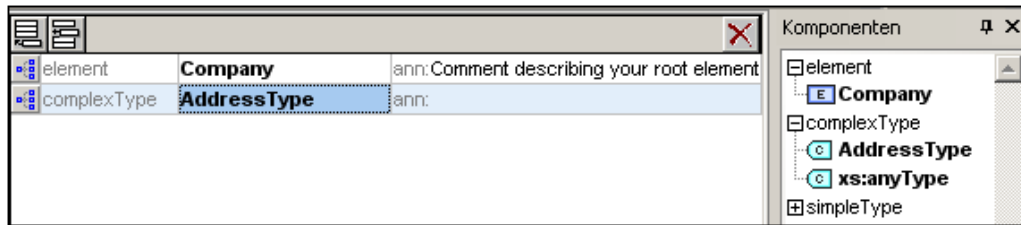
Das Basiselement Address, das wir zuvor definiert haben (und das die Elemente Name, Street und City enthält), kann in verschiedenen Adressformaten wiederverwendet werden. Wir wollen daher diese Elementdefinition als wiederverwendbaren ComplexType erstellen. Gehen Sie dazu folgendermaßen vor:


So erstellen Sie ein globales Element vom Typ Complex Type:

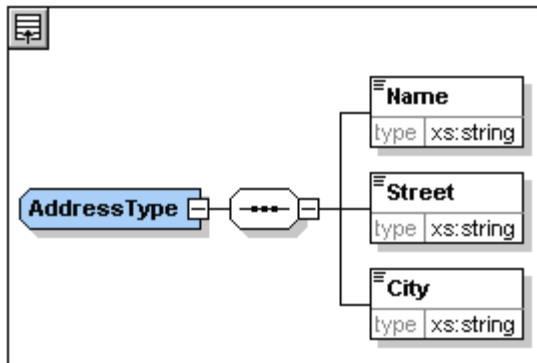
1. Klicken Sie in der Content Model-Ansicht mit der rechten Maustaste auf das Element Address.
2. Wählen Sie im Kontextmenü nun die Option Als global definieren | Complex Type. Es wird ein globaler ComplexType mit dem Namen AddressType erstellt, und dem Element Address im Content Model Company wird dieser Typ zugewiesen. Der Inhalt des Elements Address ist das Content Model von AddressType, das in einem gelb hinterlegten Kasten angezeigt wird. Beachten Sie, dass der Datentyp des Elements Address nun AddressType ist.



3. Klicken Sie auf das Symbol "Globale Elemente anzeigen" . Damit kehren Sie zur Schema-Übersicht zurück, in der nun alle globalen Komponenten angezeigt werden.
4. Klicken Sie in der Komponenten-Eingabehilfe auf das Erweiterungssymbol für Element- und ComplexType-Einträge, um die jeweiligen Schema-Konstrukte zu sehen. In der Schema-Übersicht werden nun zwei globale Komponenten angezeigt: das Element Company und der ComplexType AddressType. Außerdem wird in der Komponenten-Eingabehilfe der ComplexType AddressType angezeigt.



- Klicken Sie auf das Symbol "Content Model-Ansicht"  von AddressType, um dessen Content Model anzuzeigen (siehe Abbildung unten). Beachten Sie die Form des ComplexType-Behälters.

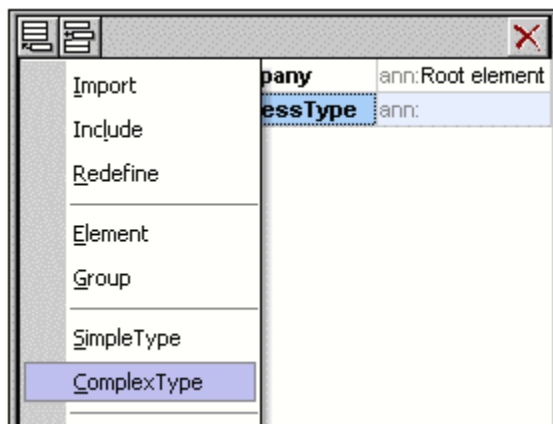


- Klicken Sie auf das Symbol "Globale Elemente anzeigen", um zur Schema-Übersicht zurückzukehren.

Erweitern einer "complex type" Definition

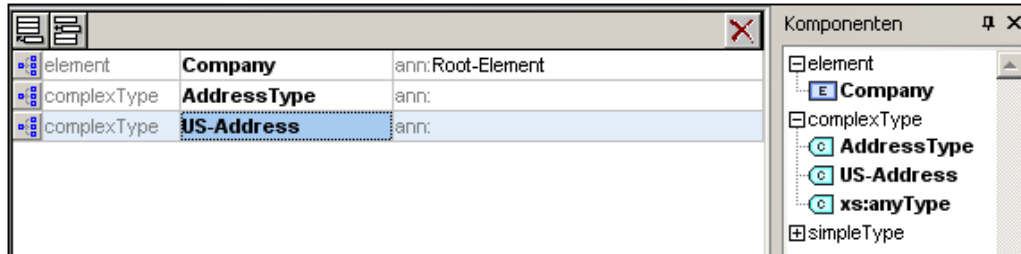
Nun werden wir die globale Komponente AddressType verwenden, um zwei Arten von länderspezifischen Adressen zu erstellen. Hierzu wird zuerst ein neuer complexType erzeugt, welcher auf der Komponente AddressType basiert. Anschließend wird diese Definition erweitert. Gehen Sie dazu folgendermaßen vor:


- Wechseln Sie zur Schema-Übersicht (Wenn Sie sich in der Content Model-Ansicht befinden, klicken Sie auf "Globale Elemente anzeigen").
- Klicken Sie auf das Symbol Anhängen (oben links im Komponentenfenster). Daraufhin wird das folgende Menü angezeigt:

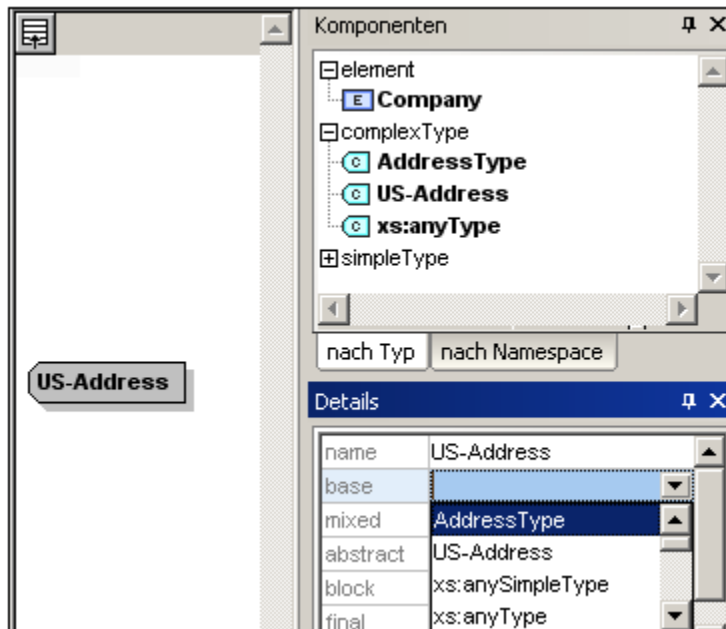


- Wählen Sie ComplexType aus dem Kontextmenü. Eine neue Leerzeile wird in der Komponentenliste angehängt, in der Sie den Namen der Komponente eingeben können.

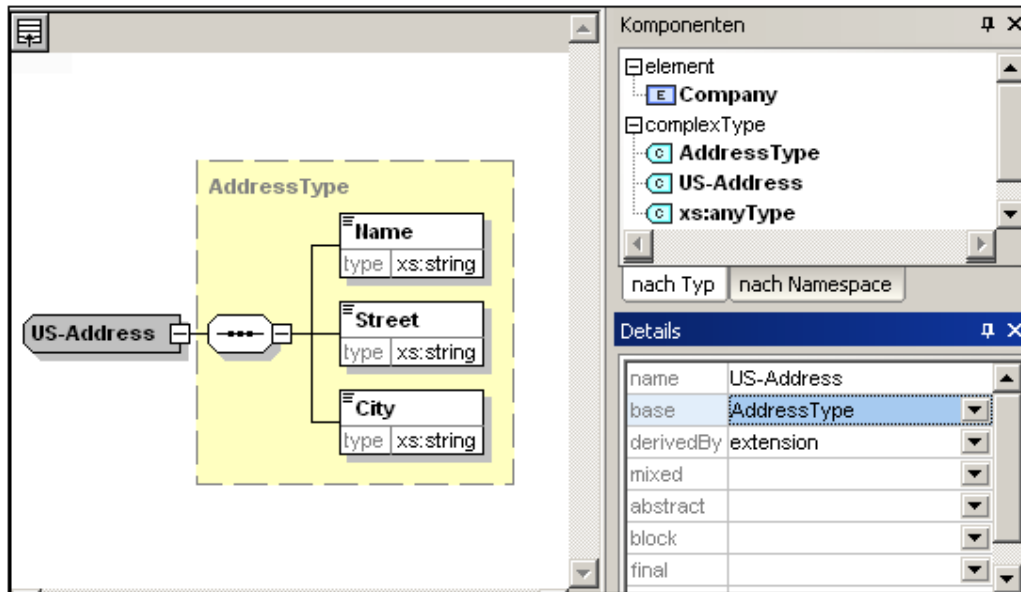
- Geben Sie US-Address ein und bestätigen Sie mit der Eingabetaste. (Sollten Sie den Bindestrich "-" beim Eingeben vergessen, und stattdessen ein Leerzeichen verwenden, wird der Name in Rot angezeigt. D.h., ein nicht erlaubtes Schriftzeichen wurde verwendet.)



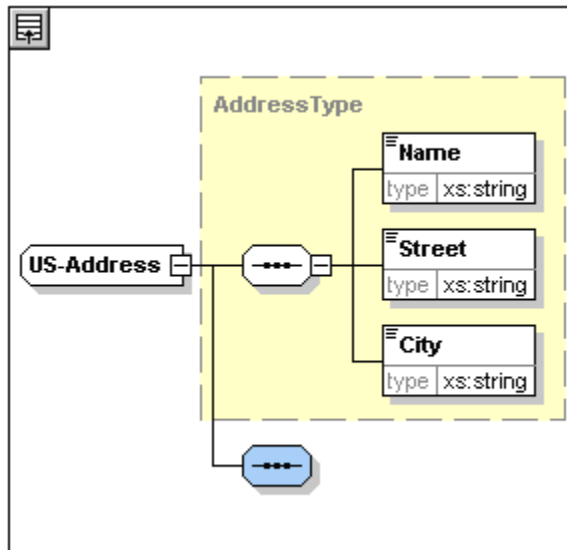
- Klicken Sie auf das Symbol "Content Model-Ansicht"  in der Zeile US-Address, um das Content Model des neuen ComplexType zu sehen. Das Content Model ist leer (siehe Abbildung unten).
- Klicken Sie in der Details-Eingabehilfe auf das Listenfeld base und wählen Sie AddressType.



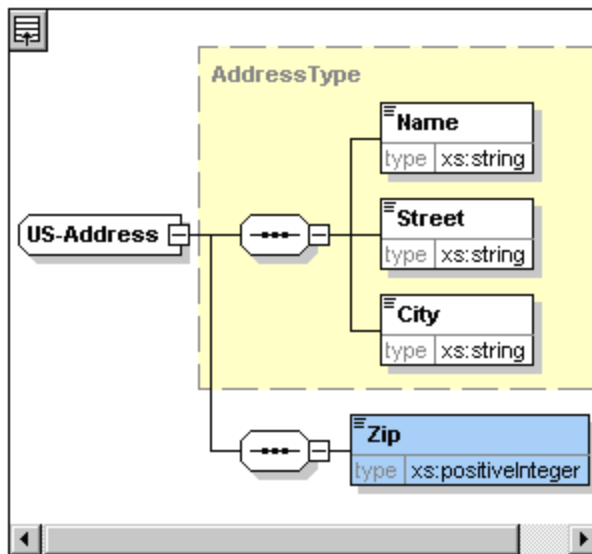
In der Content Model-Ansicht wird nun das Content Model AddressType als Content Model von US-Address (siehe Abbildung unten) angezeigt.



7. Wir können nun das Content Model des ComplexType US-Address um ein ZIP Code-Element erweitern. Klicken Sie dazu mit der rechten Maustaste auf das Element US-Address und wählen Sie Child hinzufügen | Sequence. Ein neuer Sequence-Kompositor wird außerhalb des Kastens AddressType (Abbildung unten) angezeigt; d.h., es handelt sich um die Erweiterung zu einem Element.



8. Klicken Sie mit der rechten Maustaste auf den neuen Sequence-Kompositor und wählen Sie Child hinzufügen | Element.
9. Benennen Sie das neu erstellte Element mit Zip und drücken Sie die Tabulatortaste. Daraufhin wird der Cursor in das Wertefeld der Typbeschreibungzeile gesetzt.
10. Wählen Sie in der Auswahlliste xs:positiveInteger aus und bestätigen Sie mit der Eingabetaste.



Sie haben nun einen ComplexType namens US-Address, der auf dem ComplexType AddressType basiert und diesen um einen ZIP Code (Postleitzahl) erweitert.

Globale SimpleTypes

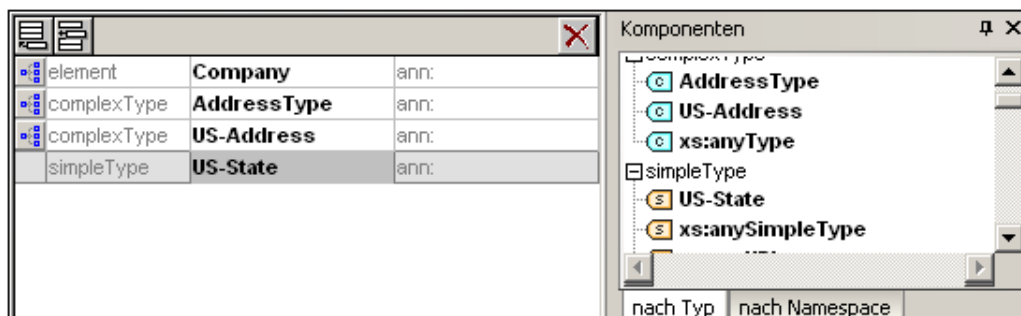
Genau wie der ComplexType US-Address auf dem ComplexType AddressType basiert, kann ein Element auch auf einem SimpleType basieren. Der Vorteil ist derselbe wie bei globalen ComplexTypes: Der SimpleType kann wiederverwendet werden. Dazu muss der SimpleType zuerst als globales Element definiert werden. In diesem Tutorial werden wir ein Content Model für US-Bundesstaaten in einem SimpleType definieren. Dieser SimpleType wird als Basis für ein weiteres Element verwendet.

Erstellen eines globalen SimpleType

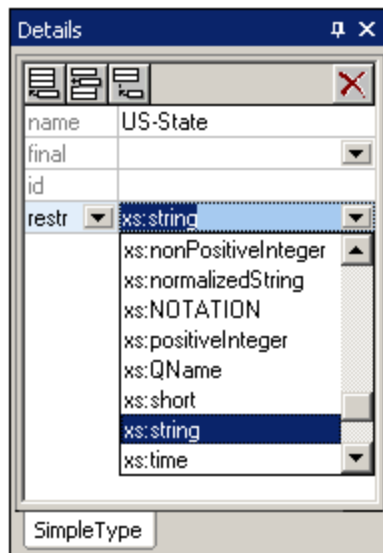
Um einen globalen SimpleType zu erstellen, müssen Sie einen neuen SimpleType an die Liste der globalen Komponenten anhängen, diesen benennen und seinen Datentyp definieren.

So erstellen Sie einen globalen SimpleType:

1. Wechseln Sie in die Schema-Übersicht (Klicken Sie auf "Globale Elemente anzeigen", wenn Sie sich in der Content Model-Ansicht befinden).
2. Klicken Sie auf das Symbol Anhängen und wählen Sie im Kontextmenü Simple Type.
3. Geben Sie US-State als Elementname ein.
4. Bestätigen Sie mit der Eingabetaste. Der SimpleType US-State wird erstellt und in der Komponenten-Eingabehilfe in die Liste der SimpleTypes aufgenommen (Klicken Sie auf das Erweiterungssymbol des SimpleType-Eintrags, um es zu sehen).




- Platzieren Sie in der Details-Eingabehilfe (Abbildung unten) den Cursor in das Wertefeld von restr und geben Sie xs:string ein oder wählen Sie xs:string in der Dropdown-Liste im Wertefeld restr aus.

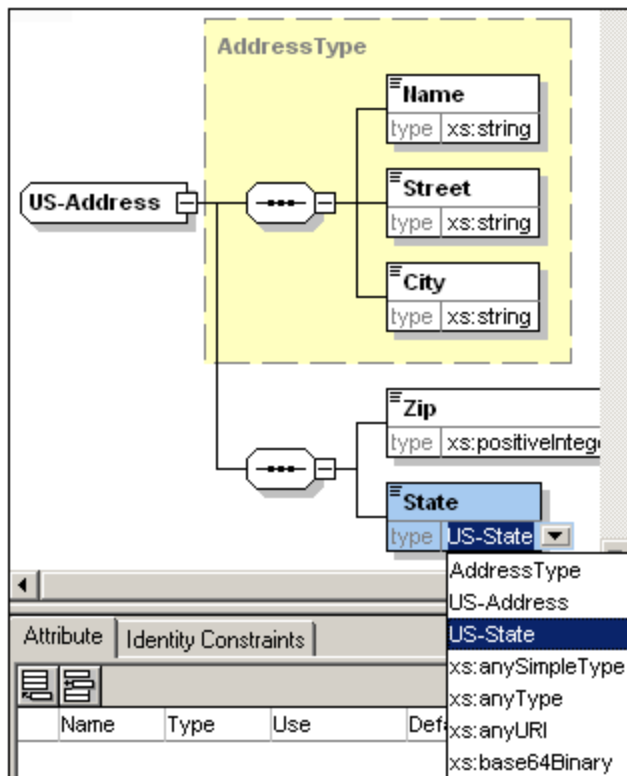


Damit wird ein SimpleType namens US-State vom Datentyp xs:string erstellt. Diese globale Komponente kann nun im Content Model von US-Address verwendet werden.

Verwendung eines globalen SimpleType in einem Content Model

Ein globaler SimpleType kann in einem Content Model dazu verwendet werden, den Typ einer Komponente zu definieren. Wir werden US-State verwenden, um im Content Model von US-Address ein Element namens State zu definieren. Gehen Sie dazu folgendermaßen vor:

- Klicken Sie in der Schema-Übersicht auf das Symbol "Content Model-Ansicht"  von US-Address.
- Klicken Sie mit der rechten Maustaste auf den unteren Sequence-Kompositor und wählen Sie Child hinzufügen | Element.
- Geben Sie State als Elementnamen ein.
- Drücken Sie die Tabulatortaste, um den Cursor in das Wertefeld der Typbeschreibungseile zu setzen.
- Wählen Sie in der Dropdown-Liste den Eintrag US-State aus.



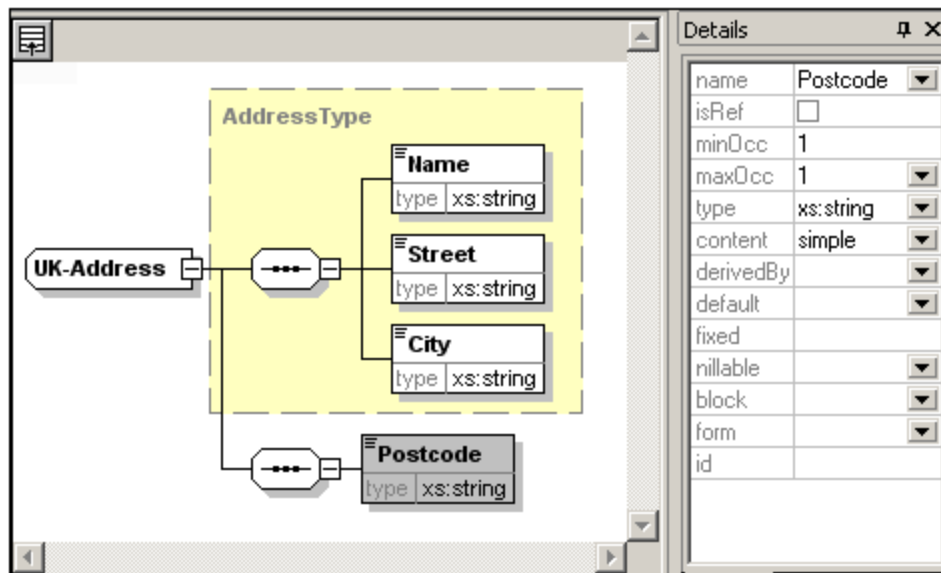
Das Element State basiert nun auf dem SimpleType US-State.

Erstellen eines zweiten ComplexType auf Basis von AddressType

Wir erstellen nun einen globalen ComplexType für Adressen in Großbritannien. Der ComplexType basiert auf dem Element AddressType und wird erweitert, bis er dem englischen Adressformat entspricht.

Gehen Sie folgendermaßen vor:



1. Erstellen Sie in der Schema-Übersicht auf Basis von AddressType einen globalen ComplexType mit dem Namen UK-Address. (base=AddressType).
2. Hängen Sie in der Content Model-Ansicht von UK-Address das neue Element Postcode an und weisen Sie ihm den Typ xs:string zu. bDas Content Model von UK-Address sollte nach erfolgreicher Erstellung folgendermaßen aussehen:

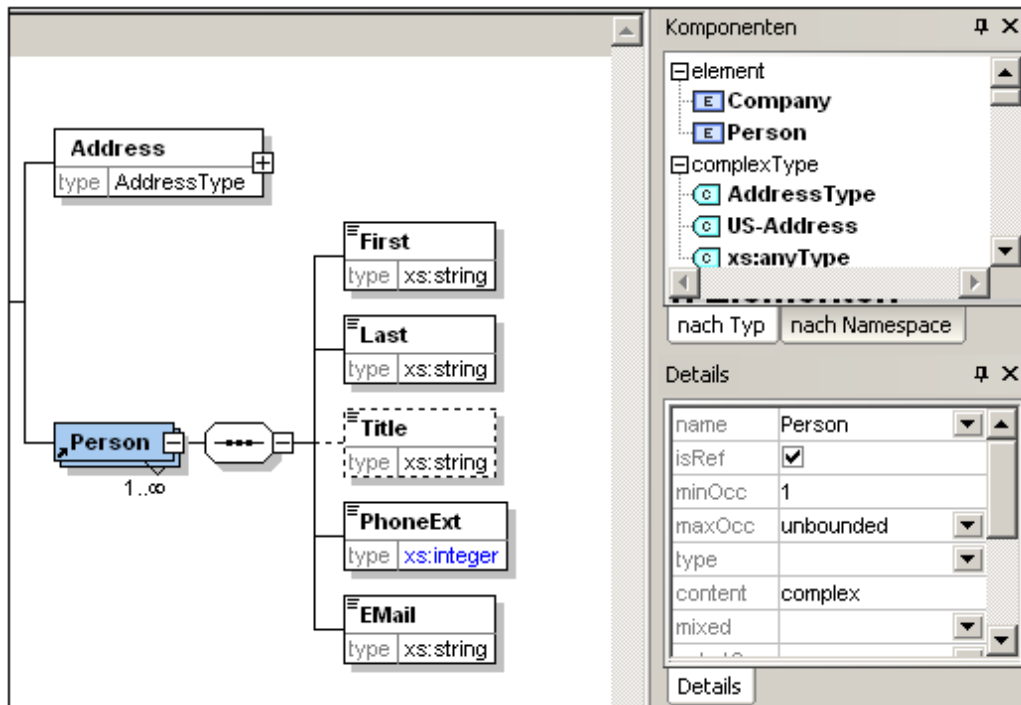



Bitte beachten Sie: In diesem Abschnitt haben Sie globale Simple und ComplexTypes erstellt und anschließend in Content Model-Definitionen verwendet. Der Vorteil von globalen Typen liegt darin, dass diese in mehreren Definitionen wiederverwendet werden können.

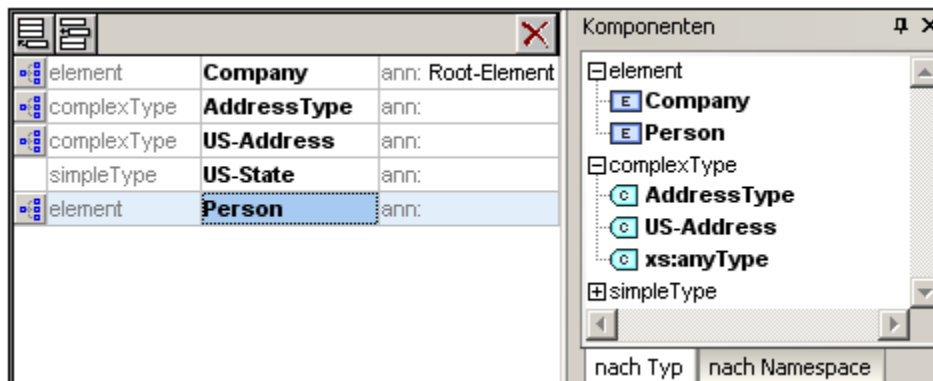
3.2 Referenzieren von globalen Elementen

In diesem Abschnitt werden wir das lokal definierte Element Person als globales Element definieren und es dann vom Element Company aus referenzieren.

1. Wechseln Sie in die Schema-Übersicht  (Globale Elemente anzeigen).
2. Klicken Sie auf das Symbol "Diagramm anzeigen"  in der Zeile Company.
3. Klicken Sie mit der rechten Maustaste auf das Element Person und wählen Sie Als global definieren | Element. Ein kleines "link"-Symbol wird in das Element Person eingefügt, das anzeigt, dass das Element das als global deklarierte Element Person referenziert. Das Kontrollkästchen isRef in der Details Eingabehilfe ist nun auf aktiv gesetzt.

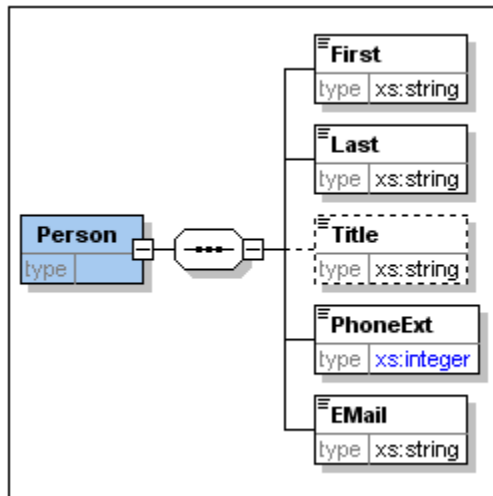


4. Klicken Sie auf das Symbol "Globale Elemente anzeigen"  um zur Schema-Übersicht zurückzukehren. Das Element Person wird nun als globales Element aufgelistet und erscheint auch in der Komponenten-Eingabehilfe.



5. Klicken Sie in der Komponenten-Eingabehilfe auf das Element Person, um das Content

Model des globalen Elements Person anzuzeigen.



Bitte beachten Sie, dass das Kästchen für das globale Element kein Link-Symbol hat, da es sich hierbei um das referenzierte Element und nicht das referenzierende Element handelt. Nur das referenzierende Element ist mit einem Link-Symbol gekennzeichnet.

Bitte beachten Sie:

- Ein Element, das ein globales Element referenziert, muss denselben Namen haben, wie das globale Element, das es referenziert.
- Globale Deklarationen geben keine Auskunft über die Positionierung der Elemente innerhalb des XML-Dokumentes. Sie bestimmen nur den Inhalt des jeweiligen Elementes. Nur wenn eine globale Deklaration von einer anderen Komponente aus referenziert wird, wird ihre Position im XML-Dokument definiert.

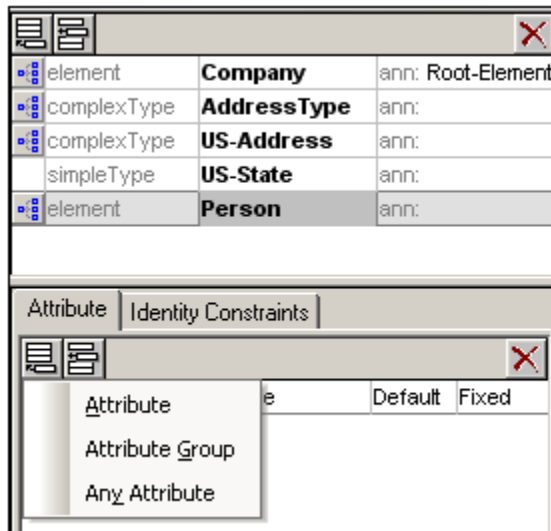
Ein global deklariertes Element kann an mehreren Stellen wiederverwendet werden. Es unterscheidet sich insofern von einem global deklarierten ComplexType, als sein Content Model nicht geändert werden kann, ohne dass auch das globale Element selbst geändert wird. Wenn Sie das Content Model eines Elements ändern, das ein globales Element referenziert, so wird auch das Content Model des globalen Elements geändert und damit auch das Content Model aller anderen Elemente, die das globale Element referenzieren.

3.3 Attribute und Attribut-Enumerationen

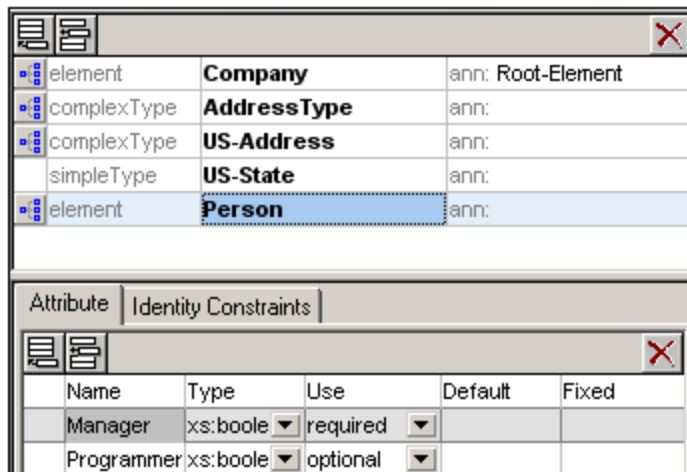
In diesem Abschnitt lernen Sie, wie man Attribute und Enumerationen für Attribute erstellt.

Definieren von Element-Attributen

1. Klicken Sie in der Schema-Übersicht auf das Element `person` um es aktiv zu setzen.
2. Klicken Sie auf das Symbol "Anhängen", (oben links im Fenster "Attribute/Identity Constraints", das den unteren Bereich des Schema-Übersichtsfensters einnimmt) und wählen Sie den Eintrag "Attribute".



3. Geben Sie `Manager` in das Namensfeld ein.
4. Wählen Sie in der Dropdown-Liste Type den Eintrag `xs:boolean`.
5. Wählen Sie in der Dropdown-Liste Use den Eintrag `required`.



6. Erstellen Sie nun auf dieselbe Art ein Attribut `Programmer` mit einem `Type=xs:boolean` und `Use=optional`.

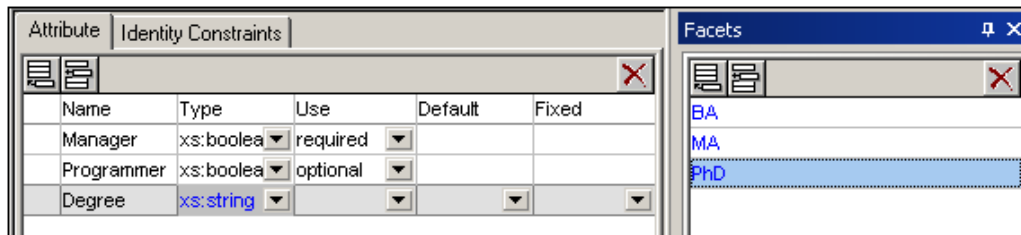
Definieren von Enumerationen für Attribute



Enumerationen sind Werte, die bei einem bestimmten Attribut zulässig sind. Wenn der Wert

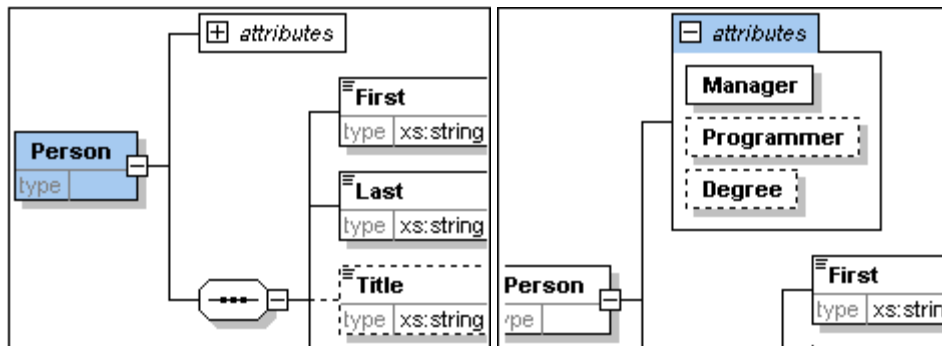
des Attributs im XML-Instanzdokument nicht in einer im XML-Schema definierten Enumeration enthalten ist, ist das Dokument ungültig. Wir werden Enumerationen für das Attribut Degree des Elements `Person` erstellen.

Gehen Sie folgendermaßen vor:

1. Klicken Sie in der Schemaübersicht auf das Element `Person`, um es zum aktiven zu machen.
2. Klicken Sie auf das Anhängen-Symbol oben links im Attribut-Fenster und wählen Sie den Eintrag "Attribute".
3. Geben Sie Degree als Attributnamen ein und wählen Sie `xs:string` als Attribut-Type.
4. Klicken Sie in der Facets-Eingabehilfe auf das Register Enumerations (siehe Abbildung unten).



5. Klicken Sie auf dem Register "Enumerations" auf das Anhängen-Symbol .
6. Geben Sie BA (Bachelor of Arts) ein und bestätigen Sie mit der Eingabetaste.
7. Mit derselben Methode geben Sie zwei weitere Werte in die Enumerations-Liste ein: MA und PhD.
8. Klicken Sie auf das Symbol "Content Model Ansicht"  des Elements `Person`.



Die zuvor definierten Attribute werden in der Content Model-Ansicht angezeigt. Wenn Sie auf das Erweiterungssymbol klicken, werden alle für dieses Element definierten Attribute angezeigt. Sie können diesen Anzeigemodus ein- und ausschalten, indem Sie die Menüoption **Schema-Design | Ansicht** konfigurieren wählen und auf die Kontrollkästchen für Attribute und Identity Constraints klicken.

9. Klicken Sie auf das Symbol "Globale Elemente anzeigen", um zur Schema-Übersicht zurückzukehren.

Speichern des fertigen Schemas

Bitte beachten Sie: Bevor Sie Ihre Schema-Datei speichern, sollten Sie zuerst die vorhandene Datei `AddressLast.xsd`, die mit XMLSpy mitgeliefert wird, umbenennen (z.B. in `AddressLast_original.xsd`), um sie nicht zu überschreiben.

Speichern Sie das fertige Schema unter einem Namen Ihrer Wahl (**Datei | Speichern unter**). Wir empfehlen Ihnen, es unter dem Namen `AddressLast.xsd` zu speichern, da die XML-Datei,

die Sie im nächsten Teil des Tutorials erstellen werden, auf dem Schema `AddressLast.xsd` basieren wird.

4 XML-Schemas: XMLSpy-Funktionen

Nach Fertigstellung Ihres XML-Schemas sollten Sie sich nun mit einigen [Navigationskürzeln](#) vertraut machen und etwas über die [Schemadokumentation](#), die Sie in XMLSpy generieren können, erfahren. Diese Themen werden in den folgenden Unterabschnitten erörtert.

In diesem Abschnitt verwendete Befehle

Wir verwenden in diesem Abschnitt des Tutorials ausschließlich die Schema/WSDL-Ansicht. Dabei werden die folgenden Befehle verwendet:




Diagramm anzeigen (oder Content Model-Ansicht anzeigen). Mit diesem Symbol, das in der Schema-Übersicht links von allen globalen Komponenten angezeigt wird, sehen Sie das Content Model der dazugehörigen globalen Komponente.

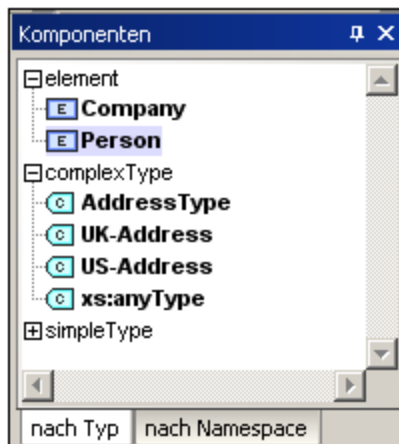
4.1 Schemanavigation

In diesem Abschnitt wird beschrieben, wie Sie effizient innerhalb der Schema/WSDL-Ansicht navigieren können. Probieren Sie diese Navigationsverfahren am besten aus, um sich damit vertraut zu machen.

Anzeigen des Content Model einer globalen Komponente

Globale Komponenten, die ein Content Model haben können, sind ComplexTypes, Elemente und Elementgruppen. Die Content Model-Ansicht dieser Komponenten lässt sich auf folgende Arten aufrufen:

- Klicken Sie in der Schema-Übersicht auf das Symbol "Diagramm anzeigen"  links vom Namen der Komponente.
- Doppelklicken Sie entweder in der Schema-Übersicht oder in der Content Model-Ansicht in der Komponenten-Eingabehilfe auf das Element, den ComplexType oder die Elementgruppe (siehe Abbildung unten).



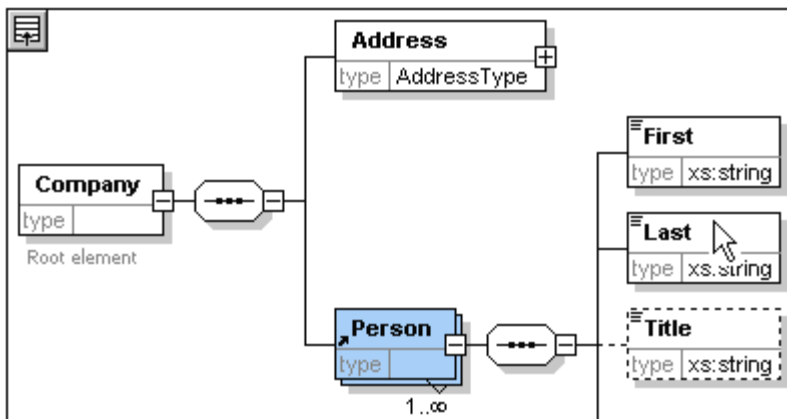
Wenn Sie in der Komponenten-Eingabehilfe auf einer der anderen globalen Komponenten (SimpleType, Attribut, Attributgruppe) doppelklicken, erscheint diese Komponente in der Schema-Übersicht (da diese Komponenten kein Content Model haben) markiert.

Das Anzeigen der Komponenten durch Doppelklick funktioniert in der Komponenten-Eingabehilfe sowohl auf dem Register "By Type" als auch auf dem Register "By Namespace".

So gelangen Sie von einem referenzierenden Element aus zur Definition eines globalen Elements:

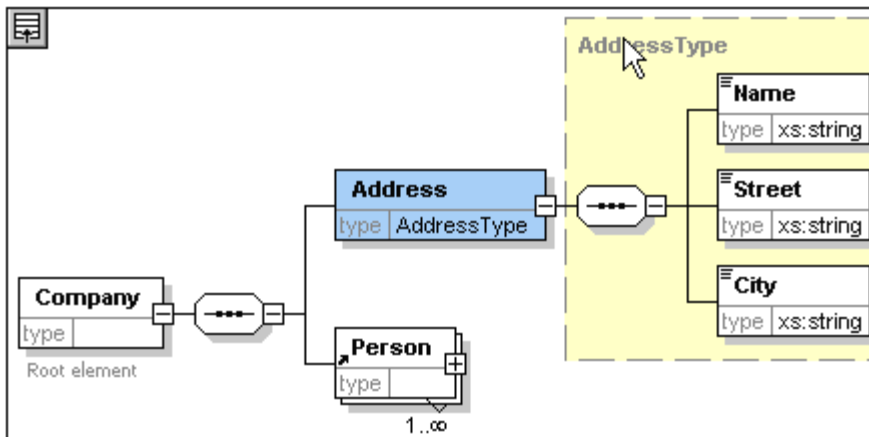
Wenn ein Content Model ein Element enthält, das ein globales Element referenziert, können Sie das Content Model dieses globalen Elements oder das seiner darin enthaltenen Komponenten direkt aufrufen, indem Sie die Strg-Taste gedrückt halten, während Sie auf das gewünschte Element doppelklicken.

Wenn Sie z.B. während der Ansicht des Content Model von `Company` die **Strg**-Taste gedrückt halten, während Sie auf `Last` doppelklicken, wird das Content Model `Person` geöffnet und das Element `Last` darin wird markiert.

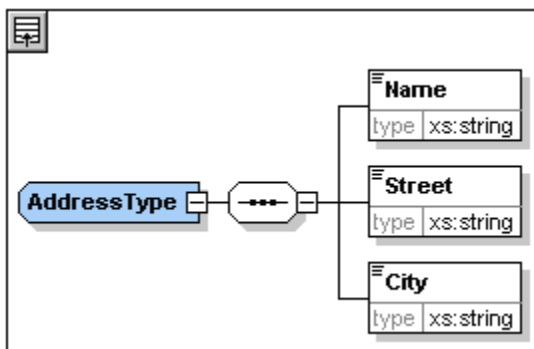


Wenn das Element `Last` markiert ist, werden alle seine Eigenschaften sofort in den entsprechenden Eingabehilfen und Informationsfenstern angezeigt.

So gelangen Sie zur Definition eines ComplexType
ComplexTypes werden oft als Elementtyp in einem Content Model verwendet. Um die Definition eines ComplexType direkt von einem Content Model aus aufzurufen, doppelklicken Sie im gelben Kästchen auf den Namen des ComplexType (siehe Mauszeiger in der Abbildung unten).



Auf diese Art gelangen Sie zur Content Model-Ansicht des ComplexType.



Bitte beachten Sie: Wie auch bei referenzierten globalen Elementen können Sie direkt zu einem Element innerhalb der ComplexType-Definition gehen, indem Sie die **Strg**-Taste gedrückt halten und auf das gewünschte Element im Content Model, das den ComplexType enthält,

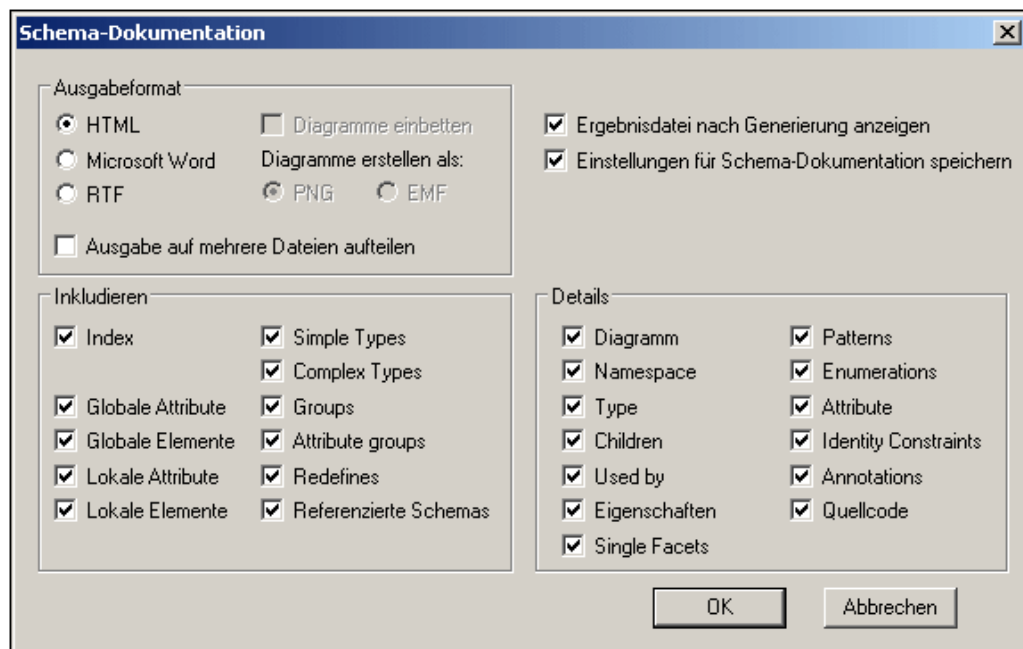
doppelklicken.

4.2 Schema-Dokumentation

XMLSpy verfügt über Funktionen zur Erstellung einer detaillierten Dokumentation von XML-Schemas im HTML- und MS Word-Format. Sie können die Komponenten und die Genauigkeit der Dokumentation auswählen. Miteinander in Zusammenhang stehende Komponenten werden im HTML- und im Word-Dokument durch Hyperlinks verbunden. Um Dokumentation im MS Word-Format zu generieren, muss MS Word auf Ihrem Computer (oder Netzwerk) installiert sein.

Sie werden in diesem Abschnitt Dokumentation für das XML-Schema `AddressLast.xsd` generieren. Gehen Sie dazu folgendermaßen vor:

1. Wählen Sie die Menüoption **Schema-Design | Schema-Dokumentation generieren**, um das Dialogfeld "Schema-Dokumentation" aufzurufen.



2. Wählen Sie das Ausgabeformat HTML und bestätigen Sie mit OK.
3. Wählen Sie im Dialogfeld "Speichern unter" das Verzeichnis und geben Sie einen Namen für die HTML-Datei ein (z.B. `AddressLast.html`). Bestätigen Sie mit **Speichern**. Das HTML-Dokument wird in der Browser-Ansicht von XMLSpy angezeigt. Wenn Sie auf einen Link klicken, gelangen Sie zur damit verknüpften Komponente.

Schema **AddressLast.xsd**

schema location: **C:\Documents and Settings\ala\My Documents\Altova\XMLSpy2007\Examples\Tutorial\AddressLast.xsd**

attribute form **unqualified**

default:

element form **qualified**

default:

targetNamespace: **http://my-company.com/namespace**

Elements [Complex types](#) [Simple types](#)

[Company](#) **[AddressType](#)** **[US-State](#)**

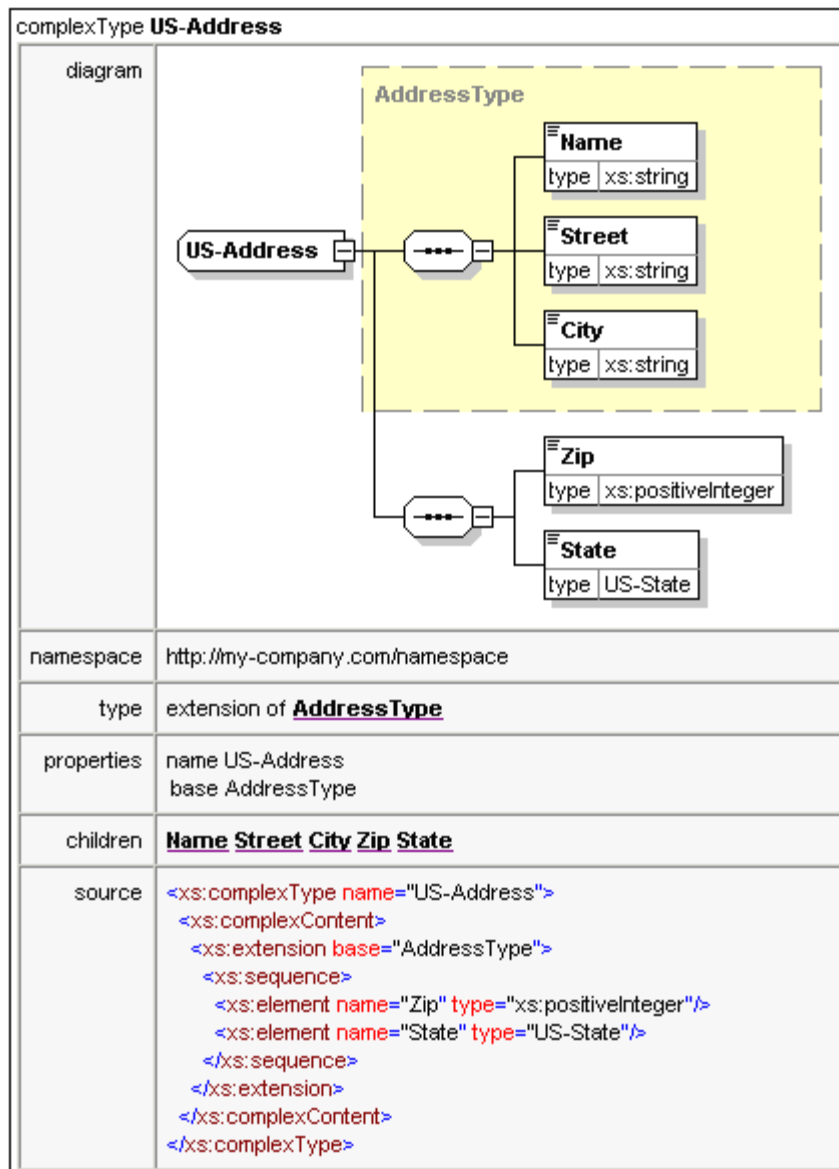
[Person](#) **[UK-Address](#)**

[US-Address](#)

element **Company**

diagram	<p>Root element</p> <p>1..∞</p>
namespace	http://my-company.com/namespace
properties	content complex
children	Address Person
annotation	documentation Root element
source	<pre> <xs:element name="Company"> <xs:annotation> <xs:documentation>Root element</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="Address" type="AddressType"/> <xs:element ref="Person" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>

Das oben abgebildete Diagramm stellt die **erste Seite** der Schema-Dokumentation im HTML-Format dar. Falls Komponenten anderer Schemas miteinbezogen wurden, werden diese ebenfalls dokumentiert.



Das oben abgebildete Diagramm stellt dar, wie ComplexTypes dokumentiert werden.

element US-Address/Zip	
diagram	
namespace	http://my-company.com/namespace
type	xs:positiveInteger
properties	name Zip isRef 0 content simple
source	<code><xs:element name="Zip" type="xs:positiveInteger"/></code>
element US-Address/State	
diagram	
namespace	http://my-company.com/namespace
type	US-State
properties	name State isRef 0 content simple
source	<code><xs:element name="State" type="US-State"/></code>
simpleType US-State	
namespace	http://my-company.com/namespace
type	xs:string
properties	name US-State
used by	element US-Address/State
source	<code><xs:simpleType name="US-State"> <xs:restriction base="xs:string"/> </xs:simpleType></code>

Das oben abgebildete Diagramm zeigt an, wie Elemente und SimpleTypes dokumentiert werden.

Sie können nun die MS Word-Ausgabeoption ausprobieren. Das Word-Dokument wird in MS Word geöffnet. Um Hyperlinks im MS Word-Dokument zu verwenden, halten Sie die **Strg**-Taste gedrückt, während Sie auf den Link klicken.

5 XML Dokumente

In diesem Abschnitt lernen Sie:

Erstellen eines neuen XML Dokumentes in XMLSpy, Verwenden von verschiedenen Ansichten und Eingabehilfen für erleichtertes Bearbeiten und Validieren von Daten in XMLSpy.

Ziel

In diesem Abschnitt werden folgende Punkte behandelt:

- Erstellen eines neuen XML-Dokumentes basierend auf dem Schema `AddressLast.xsd`.
- Definition des Elementtyps, um für das Element bei der Validierung ein erweitertes Content Model für dieses Element bereitzustellen.
- Einfügen von Elementen und Attributen und Eingabe von Inhalt für diese in der Grid- und der Text-Ansicht mit Hilfe der intelligenten Eingabehilfen.
- Kopieren von XML-Daten von XMLSpy in ein MS Excel-Blatt, Einfügen von Daten in MS Excel und Zurückkopieren der Daten nach XMLSpy. Diese Funktion steht in der Datenbank/Tabellen-Ansicht der Grid-Ansicht zur Verfügung.
- Sortieren von XML-Elementen mit Hilfe der Sortierfunktion der Datenbank/Tabellen-Ansicht.
- Validieren des XML-Dokumentes.
- Ändern des Schemas, so dass dreistellige Telefondurchwahlnummern eingegeben werden können.

In diesem Abschnitt verwendete Befehle



Datei | Neu erstellt eine neue XML-Datei.



Ansicht | Text wechselt in die Text-Ansicht.



Ansicht | Enhanced Grid wechselt in die Enhanced Grid-Ansicht.



XML | Tabelle | Als Tabelle anzeigen zeigt Mehrfachinstanzen eines einzelnen Elementtyps auf einer einzelnen hierarchischen Ebene als Tabelle an. Diese Ansicht des Elements nennt sich Datenbank/Tabellen-Ansicht (oder einfach Tabellenansicht). Das Symbol dient zum Wechseln zwischen der Tabellen,- und der normalen Enhanced Grid-Ansicht.



F7 Zum Kontrollieren, ob das Dokument wohlgeformt ist.



F8 Validiert das XML-Dokument gegen das damit verknüpfte Schema bzw. die DTD.



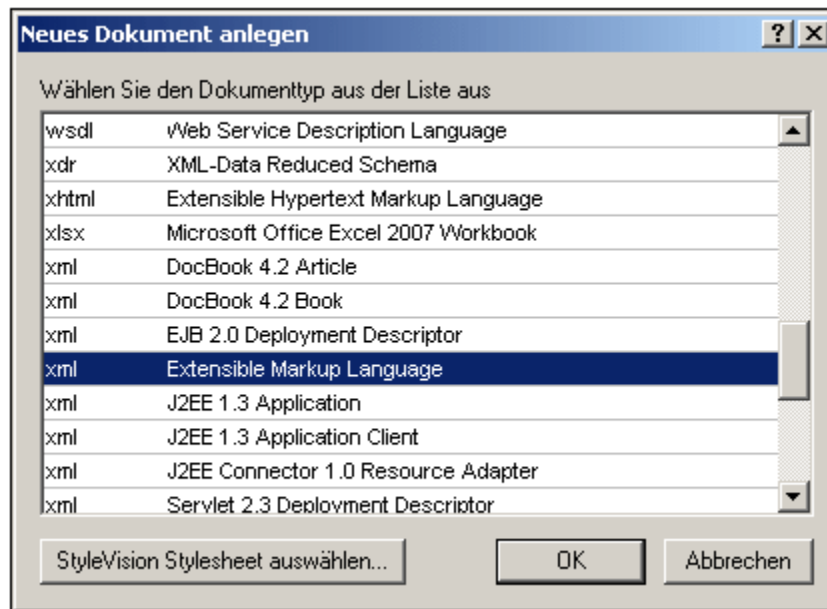
Öffnet die dazugehörige DTD oder XML-Schema-Datei.

5.1 Erstellen eines neuen XML-Dokuments

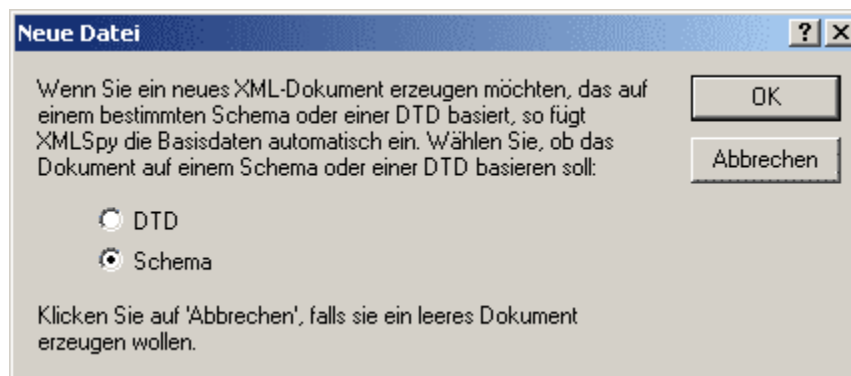
Beim Erstellen einer neuen XML-Datei in XMLSpy haben Sie die Option, dieser ein Schema (DTD oder XML-Schema) zugrunde zu legen oder auch nicht. In diesem Abschnitt werden wir eine neue Datei erstellen, die auf dem zuvor im Rahmen dieses Tutorials erzeugten Schema `AddressLast.xsd` basiert.

So erstellen Sie eine neue XML-Datei:

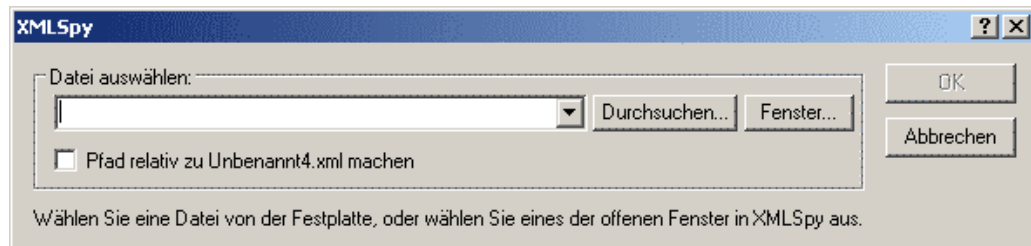
1. Wählen Sie die Menüoption **Datei | Neu**, um Dialogfeld "Neues Dokument anlegen" aufzurufen.



2. Wählen Sie im Dialogfeld den Eintrag `Extensible Markup Language` (oder den Eintrag `XML-Dokument` oder einen generischen XML-Dokumenteintrag) und bestätigen Sie mit **OK**. Im nächsten Fenster werden Sie gefragt, ob das XML-Dokument auf einer DTD oder einem Schema basieren soll.



3. Wählen Sie das Optionsfeld "Schema" und bestätigen Sie mit OK. Sie werden nun aufgefordert, die Schema-Datei auszuwählen, auf der Ihr XML-Dokument basieren soll.



4. Wählen Sie "Durchsuchen" oder "Fenster", um die Schema-Datei zu suchen. Bei Klicken auf die Schaltfläche "Fenster" werden alle Dateien, die in XMLSpy und Projekten geöffnet sind, aufgelistet. Wählen Sie `AddressLast.xsd` und bestätigen Sie Ihre Auswahl mit OK. Im Hauptfenster wird ein neues XML-Dokument geöffnet. Das Dokument enthält alle Hauptelemente, die im Schema definiert wurden.
5. Klicken Sie auf das Register **Grid**, um die Enhanced Grid-Ansicht auszuwählen.
6. In der Grid-Ansicht ist das gesamte Dokument ausgewählt. Wenn Sie auf ein Element klicken, wird die Auswahl auf dieses Element reduziert. Ihr Dokument sollte etwa folgendermaßen aussehen:

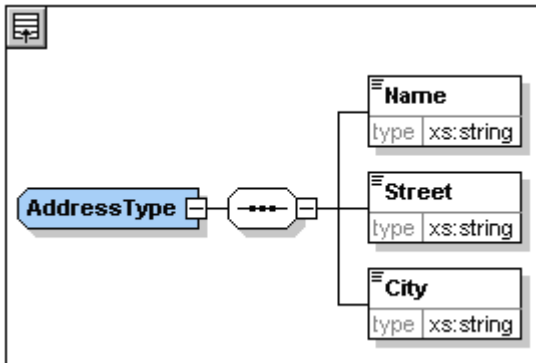
XML	
version	1.0
encoding	UTF-8
Company	
xmlns	http://my-company.com/namespace
xmlns:xsi	http://www.w3.org/2001/XMLSchema-...
xsi:schemaLoca...	http://my-company.com/namespace AddressLast.xsd
Address	
Person Manager=	

7. Klicken Sie auf das Symbol  neben Address, um die Child-Elemente von Address anzuzeigen. Ihr Dokument sollte aussehen wie dieses:

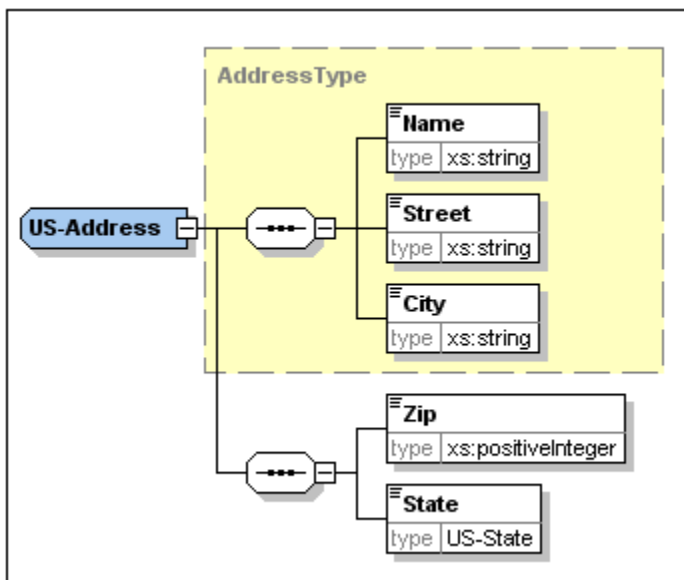
Company	
xmlns	http://my-company.com/namespace
xmlns:xsi	http://www.w3.org/2001/XMLSchema-insta...
xsi:schema...	http://my-company.com/namespace AddressLast.xsd
Address	
	Name
	Street
	City
Person Manager=	

5.2 Festlegen des Typs eines Elements

Die in der Grid-Ansicht angezeigten Child-Elemente von `Address` sind die, welche für den globalen ComplexType `AddressType` (*Content Model dazu in der Abbildung unten*) definiert wurden.



Wir möchten jedoch einen spezifischen US- oder UK-Adresstyp verwenden anstelle des generischen Adresstyps. Sie werden sich erinnern, dass wir im Schema `AddressLast.xsd` globale ComplexTypes für `US-Address` und `UK-Address` erstellt haben, indem wir den ComplexType `AddressType` erweitert haben. Unten sehen Sie das Content Model von `US-Address`.



Um festzulegen, dass das Element `Address` im XML-Dokument einem der erweiterten `Address`-Typen entsprechen muss (`US-Address` oder `UK-Address`) und nicht dem generischen `AddressType`, müssen wir den erforderlichen erweiterten ComplexType als ein Attribut des Elements `Address` definieren.

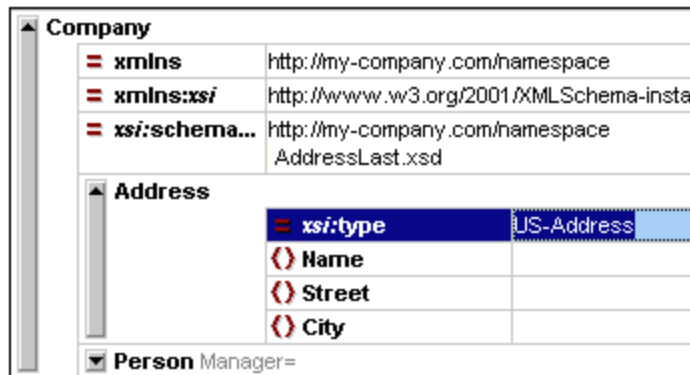
Gehen Sie folgendermaßen vor:

1. Klicken Sie in der Grid-Ansicht mit der rechten Maustaste auf das Element `Name` und wählen Sie im Kontextmenü **Einfügen | Attribute**.



Dem Element `Address` wird ein Attributfeld hinzugefügt.

2. Stellen Sie sicher, dass als Name des Attributs `xsi:type` eingegeben wird (*Abbildung unten*).
3. Drücken Sie die Tabulatortaste um in das nächste (Werte)-Feld zu gelangen.



4. Geben Sie `US-Address` als Attributwert ein.

Bitte beachten Sie: Das vorgesetzte `xsi` ermöglicht die Verwendung von speziellen XML-Schema-spezifischen Befehlen in Ihrer XML-Dokumentinstanz. Beachten Sie, dass der Namespace für das Präfix `xsi` automatisch zum Dokumentelement hinzugefügt wurde, als Sie Ihrer XML-Datei ein Schema zugewiesen haben. Sie haben soeben einen Typ für das Element `Address` festgelegt. Nähere Informationen dazu finden Sie in der [XML Schema-Spezifikation](#).

5.3 Eingabe von Daten in der Grid-Ansicht

Sie können jetzt Daten in Ihr XML-Dokument eingeben.

Gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in das Wertefeld `Name` (oder verwenden Sie die Pfeiltasten) und geben Sie `US dependency` ein. Bestätigen Sie mit der **Eingabetaste**.

Company	
xm:ns	http://my-company.com/namespace
xm:ns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schema...	http://my-company.com/namespace AddressLast.xsd
Address	
xsi:type	US-Address
Name	US dependency
Street	
City	
Person Manager=	

2. Geben Sie auf dieselbe Art einen `Street` und `City` Namen ein (z.B. `Noble Ave.` und `Dallas`).
3. Klicken Sie auf das Element `person` und drücken Sie die **Entf**-Taste, um es zu löschen. (Wir werden es im nächsten Abschnitt des Tutorials wieder hinzufügen). Daraufhin erscheint nun das gesamte Element `Address` markiert.
4. Klicken Sie auf ein beliebiges Child-Element des Elements `Address`, um die Auswahl für alle Child-Elemente von `Address` mit Ausnahme des gewählten Elements aufzuheben. Ihr XML-Dokument sollte folgendermaßen aussehen:

Company	
xm:ns	http://my-company.com/namespace
xm:ns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schema...	http://my-company.com/namespace AddressLast.xsd
Address	
xsi:type	US-Address
Name	US dependency
Street	Noble Ave
City	Dallas

5.4 Eingabe von Daten in der Textansicht

Die Textansicht eignet sich aufgrund ihrer intelligenten Editierfunktionen für DTD/XML-Schema ideal zum Editieren der eigentlichen Daten und des Markup-Codes von XML-Dateien.

Funktionen zum Bearbeiten der Struktur

Die Textansicht bietet eine Reihe von Funktionen zur Bearbeitung der Struktur, die das Editieren größerer Textabschnitte vereinfachen. Dazu zählen die folgenden Funktionen, die Sie durch Anklicken des entsprechenden Symbols aktivieren bzw. deaktivieren können.



Aktiviert/deaktiviert die Zeilennummerierung



Aktiviert/deaktiviert die Klappleiste



Aktiviert/deaktiviert die Lesezeichenfunktion



Fügt Lesezeichen ein und entfernt diese



Aktiviert/deaktiviert die Einrücklinien

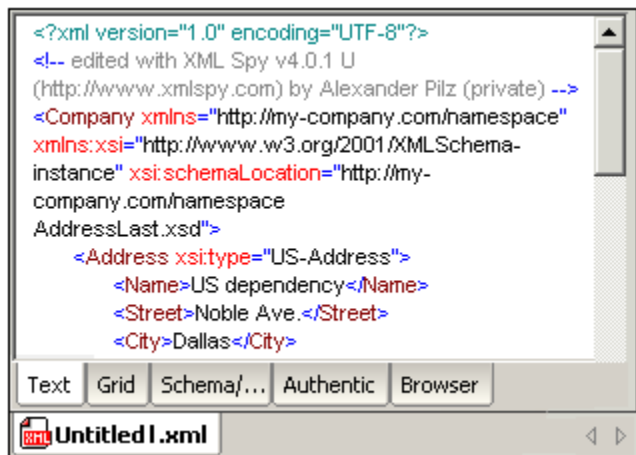
In der unten gezeigten Abbildung sehen Sie die Textansicht der XML-Datei, die sich in Bearbeitung befindet. Alle Sonderfunktionen der Textansicht sind aktiviert. Aus Gründen der Übersichtlichkeit werden diese Funktionen (Zeilennummerierung, Einrücklinien usw.) in weiterer Folge in dieser Anleitung in Textansichten nicht mehr angezeigt. Nähere Informationen zur Textansicht finden Sie im Benutzerhandbuch.

```
1      <?xml version="1.0" encoding="UTF-8"?>
2      =<Company xmlns="http://my-company.com/narr
3      C:\PROGRAM~1\Altova\XMLSPY2004\Examples\
4      =<Address xsi:type="US-Address">
5      ..... <Name>US dependency</Name>
6      ..... <Street>Noble Ave</Street>
7      ..... <City>Dallas</City>
8      </Address>
```

Bearbeitung in der Textansicht

In diesem Abschnitt werden wir Daten in der Textansicht eingeben und bearbeiten, um die Funktionen der Textansicht zu erläutern. Gehen Sie folgendermaßen vor:

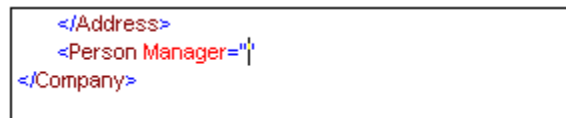
1. Wählen Sie die Menüoption **Ansicht | Text** oder klicken Sie auf das Register Text. Sie sehen nun das XML-Dokument in seiner Textform mit der Syntaxfärbung.



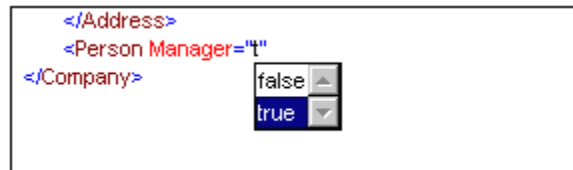
2. Setzen Sie den Cursor hinter den End-Tag des Elements "Address" und drücken Sie die Eingabetaste um eine neue Zeile einzufügen.
3. Geben Sie ein "Kleiner als..." Zeichen (<) an dieser Stelle ein. Eine Auswahlliste aller an dieser Stelle (laut Schema) zulässigen Elemente erscheint. Da an dieser Stelle nur das Element `person` zulässig ist, wird nur dieses eine Element in der Liste angezeigt.



4. Wählen Sie den Eintrag `Person` aus. Das Element `person` wird mit seinem Attribut `Manager` eingefügt, wobei sich der Cursor innerhalb des Wertefelds des Attributs `Manager` befindet.



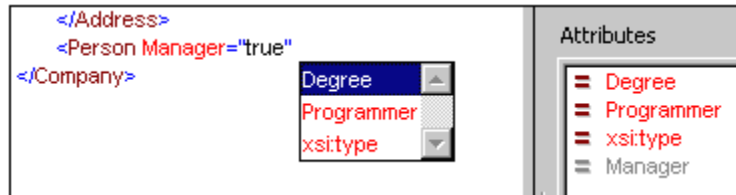
5. Wählen Sie in der Dropdown-Liste des Attributs `Manager` den Eintrag `true` aus.



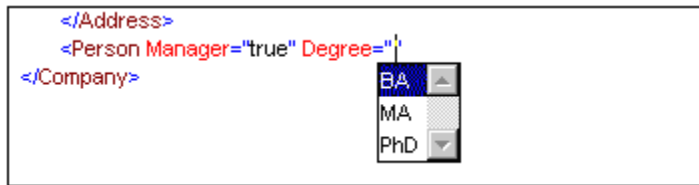
Nach Betätigen der Eingabetaste wird der Wert `true` an der Cursorposition eingefügt.

6. Setzen Sie den Cursor an das Ende der Zeile (oder betätigen Sie die "Ende"-Taste Ihrer Tastatur). Drücken Sie die Leerzeilentaste einmal. Daraufhin wird eine Dropdown-Liste geöffnet, die diesmal eine Liste von an dieser Stelle zulässigen

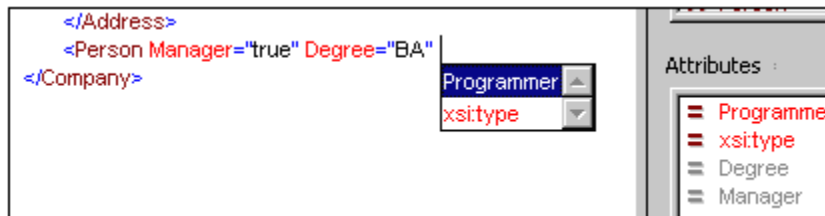
Attributen enthält. In der Attribut-Eingabehilfe sind verfügbare Attribute rot markiert. Das Attribut "Manager" ist ausgegraut, da es bereits verwendet wurde.



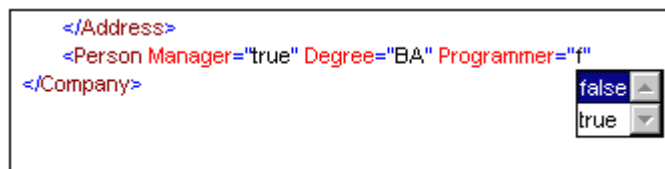
- Wählen Sie `Degree` durch Betätigen des 'Nach unten'-Pfeiles und drücken Sie die Eingabetaste. Eine weitere Liste wird geöffnet. Hier können Sie eine der vordefinierten Enumerationen auswählen (`BA`, `MA` oder `PhD`).



- Wählen Sie `BA` aus der Liste, drücken Sie die Eingabetaste und setzen Sie den Cursor durch Drücken der Ende-Taste an das Ende der Zeile. Drücken Sie die Leerzeilentaste. Nun sind `Manager` und `Degree` in der Attribut-Eingabehilfe ausgegraut.



- Wählen Sie `Programmer` aus der Liste und drücken Sie die Eingabetaste.



- Geben Sie den Buchstaben "f" ein und drücken Sie die Eingabetaste.
- Setzen Sie den Cursor ans Ende der Zeile, und geben Sie das "größer als..." Zeichen (>) ein. XMLSpy fügt automatisch alle erforderlichen Child-Elemente von `person` ein. (Beachten Sie, dass das optionale Element `title` nicht eingefügt wird). Jedes Element hat einen Start- und einen End-Tag aber keinen Inhalt.

```

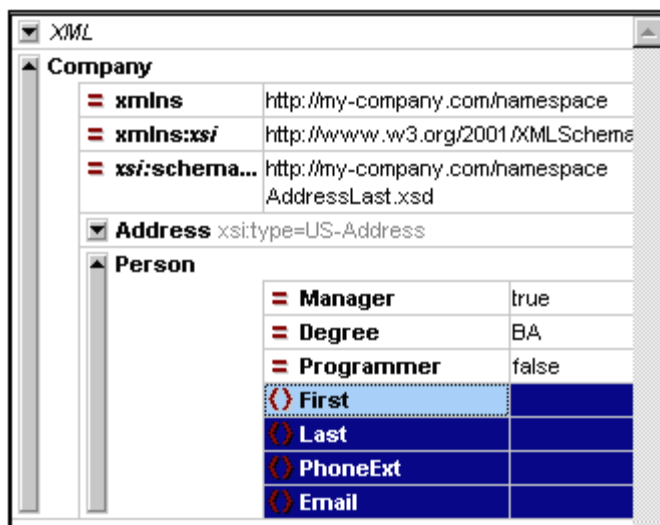
<?xml version="1.0" encoding="UTF-8"?>
<Company xmlns="http://my-company.com/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://my-company.com/namespace
AddressLast.xsd">
  <Address xsi:type="US-Address">
    <Name>US dependency</Name>
    <Street>Noble Ave</Street>
    <City>Dallas</City>
  </Address>
  <Person Manager="true" Degree="BA" Programmer="false">
    <First></First>
    <Last></Last>
    <PhoneExt></PhoneExt>
    <Email></Email>
  </Person>
</Company>

```

Sie könnten nun die Personendaten hier eingeben, doch wollen wir nun in die Grid-Ansicht wechseln, damit Sie sehen, wie einfach sich Ansichten beim Bearbeiten eines Dokuments wechseln lassen.

Wechseln zur Grid-Ansicht

Um zur Grid-Ansicht zu wechseln, wählen Sie die Menüoption **Ansicht | Enhanced Grid** oder klicken Sie auf das Register Grid. Die neu hinzugefügten Child-Elemente von `Person` erscheinen markiert.



Wir wollen das Dokument nun validieren und etwaige Fehler korrigieren.

5.5 Validierung des Dokuments

XMLSpy bittet zwei Funktionen zur Überprüfung des XML-Dokuments:


- Wohlgeformtheitprüfung
- Validierung

Wenn ein Fehler bei einer dieser beiden Überprüfungen gefunden wird, muss das Dokument entsprechend geändert werden.

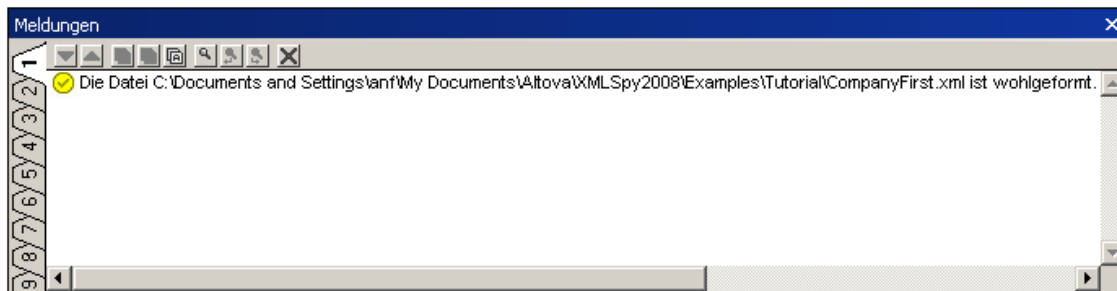
Überprüfung der Wohlgeformtheit

Ein XML-Dokument ist wohlgeformt, wenn es zu jedem Start-Tag einen End-Tag gibt, Elemente korrekt verschachtelt sind und keine Zeichen fehlen oder an der falschen Stelle stehen (z.B. eine Entity ohne das Semikolon usw.).

Wählen Sie zum Prüfen der Wohlgeformtheit die Menüoption **XML | Wohlgeformtheit prüfen**

oder drücken Sie die Taste **F7** oder klicken Sie auf . Am unteren Rand des Hauptfensters erhalten Sie eine Meldung, welche anzeigt, ob das Dokument wohlgeformt ist.

Beachten Sie, dass das Validierungsfenster neun Ausgaberegister enthält. Das Ergebnis der Validierung wird immer auf dem aktuellen Register angezeigt. Sie können daher die Wohlgeformtheit für eine Schemadatei auf Register 1 überprüfen und das Ergebnis aufbewahren, indem Sie zu Register 2 wechseln, bevor Sie das nächste Schema-Dokument validieren (andernfalls wird Register 1 durch das Validierungsergebnis überschrieben.)




Bitte beachten Sie: Bei der Wohlgeformtheitsprüfung nicht geprüft wird, ob die Struktur der XML-Datei schemakonform ist. Dies wird bei der Validierung der Datei überprüft.

Überprüfung der Gültigkeit (Validierung)

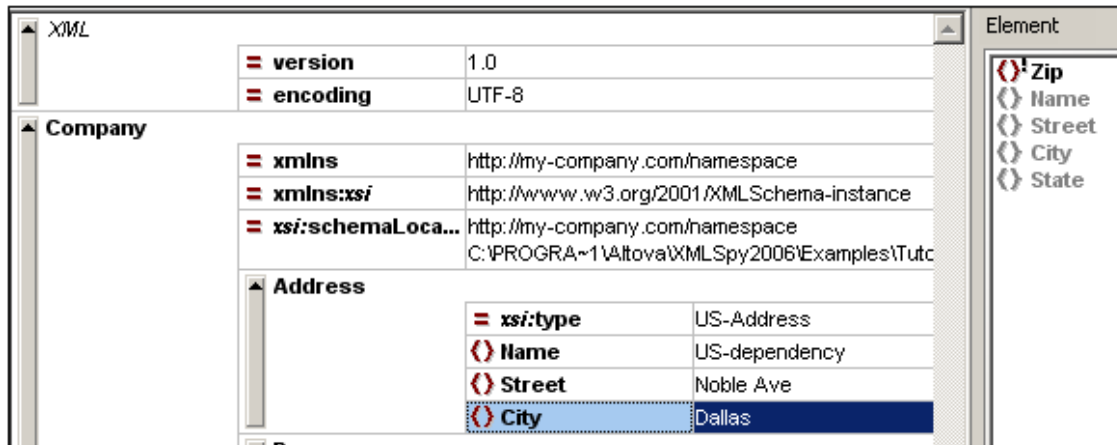
Ein XML-Dokument ist gemäß einem Schema gültig, wenn es von Struktur und Inhalt mit den Schema-Spezifikationen übereinstimmt.

Um die Gültigkeit Ihres XML-Dokuments zu überprüfen, wählen Sie zuerst die Grid-Ansicht aus und wählen Sie anschließend die Menüoption **XML | Validieren** oder drücken Sie die Taste **F8**

oder klicken Sie auf . Im unteren Bereich des Hauptfensters wird eine Fehlermeldung angezeigt, dass die Datei nicht gültig ist. Nach dem Element `City` in `Address` werden obligatorische Elemente erwartet. Wenn Sie Ihr Schema überprüfen, werden Sie feststellen, dass der ComplexType `US-Address` (dem aufgrund von `xsi:type` attribute dieses `Address`-Element entsprechen muss) ein Content Model hat, in dem auf das Element `City` ein Element `zip` und ein Element `State` folgen muss.

Beheben von Validierungsfehlern

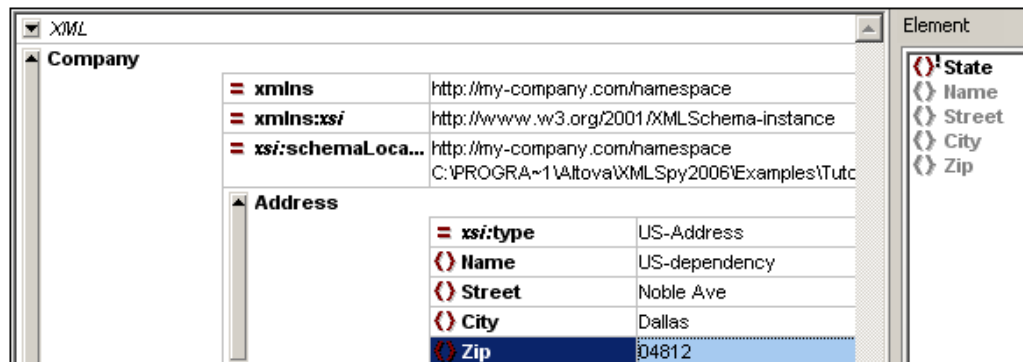
Das Element, welches den Fehler verursacht hat, in diesem Fall das Element `city`, wird markiert angezeigt.



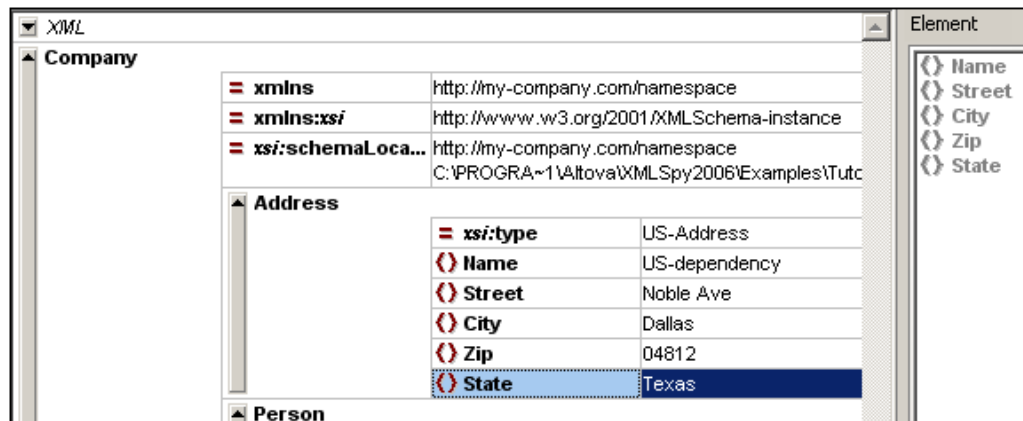
Bitte beachten Sie hierzu auch die Eingabehilfe (oben rechts), bei welcher ein Ausrufungszeichen "!" vor dem Element `zip` anzeigt, dass es sich hierbei um ein **obligatorisches** Element handelt.

Um den Validierungsfehler zu beheben, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie auf das Element `zip` in der Element-Eingabehilfe. Hierdurch wird das Element `zip` unter dem Element `city` eingefügt. (Das Register "Anhängen" ist in der Element-Eingabehilfe aktiv.)
2. Drücken Sie die Tabulatortaste, Stellen Sie sicher, dass sich der Cursor zwischen dem Start- und dem End-Tag des Elements `zip` befindet, geben Sie eine Postleitzahl (Zip) ein - (z.B. 04812) und bestätigen Sie mit der Eingabetaste. Die Element-Eingabehilfe zeigt nun an, dass das Element `state` obligatorisch ist (ein Ausrufezeichen ist dem Element vorangestellt). *Siehe Screenshot unten.*



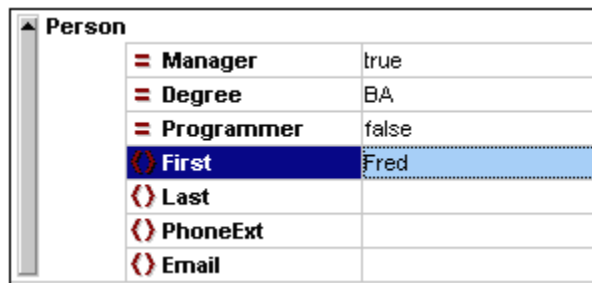
3. Doppelklicken Sie in der Element-Eingabehilfe auf das Element `state`, drücken Sie die Tabulatortaste und geben Sie den Namen des Bundesstaates ein - z.B. `Texas`. Bestätigen Sie mit der Eingabetaste. Die Element-Eingabehilfe enthält nun nur noch ausgegraute Elemente. Hierdurch wird angezeigt, dass alle obligatorischen Child-Elemente von `Address` eingefügt worden sind.



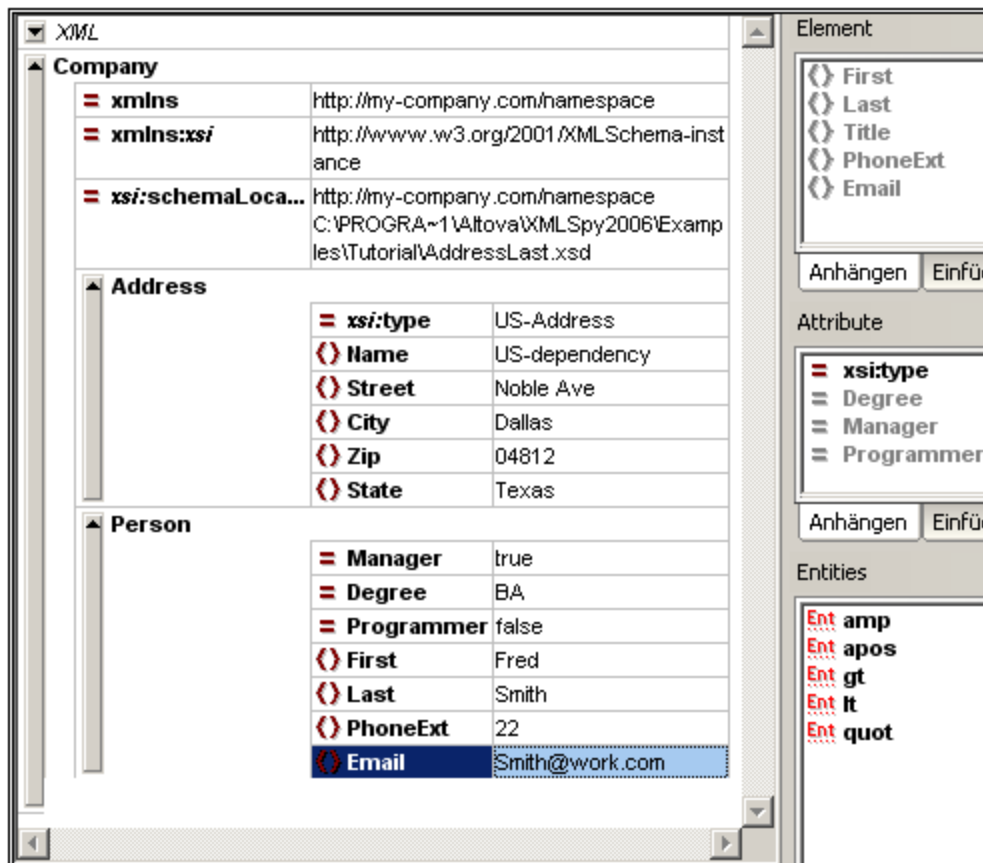
Fertigstellen des Dokuments und erneutes Validieren


Stellen Sie nun das Dokument fertig (geben Sie die Daten für das Element `Person` ein), bevor Sie es erneut validieren. Gehen Sie folgendermaßen vor:

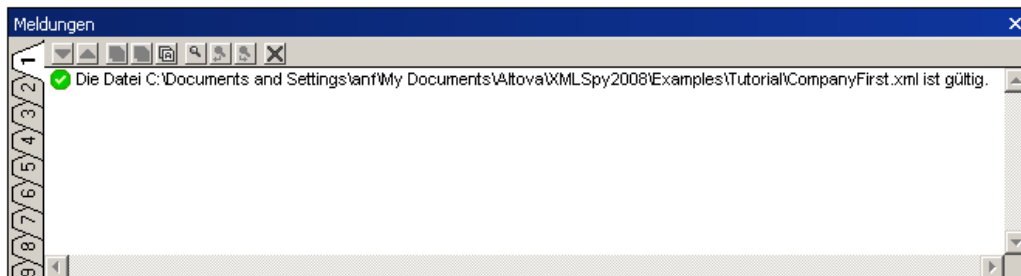
1. Klicken Sie auf das Wertefeld des Elements `First` und geben Sie den Vornamen der Person ein - z.B. `Fred`, bestätigen Sie mit der Eingabetaste.



2. Geben Sie nun auf dieselbe Art Daten für alle Child-Elemente, also `Last`, `PhoneExt` und `Email` ein. Beachten Sie, dass der Wert von `PhoneExt` eine Ganzzahl mit einem Maximalwert von 99 sein muss (da dies der gültige Wertebereich ist, den Sie im Schema definiert haben). Ihr XML-Dokument sollte in der Grid-Ansicht nun so aussehen:



3. Klicken Sie auf die Schaltfläche , um das Dokument neuerlich zu prüfen. Diesmal erhalten Sie die Nachricht "Diese Datei ist gültig". Das XML-Dokument ist nun gemäß dem Schema gültig.



4. Wählen Sie die Menüoption **Datei | Speichern** und benennen Sie das Dokument - z.B. `CompanyFirst.xml`. Beachten Sie, dass es im Ordner Tutorial bereits eine fertige Datei mit dem Namen `CompanyFirst.xml` gibt, d.h. Sie sollten diese vorher umbenennen.

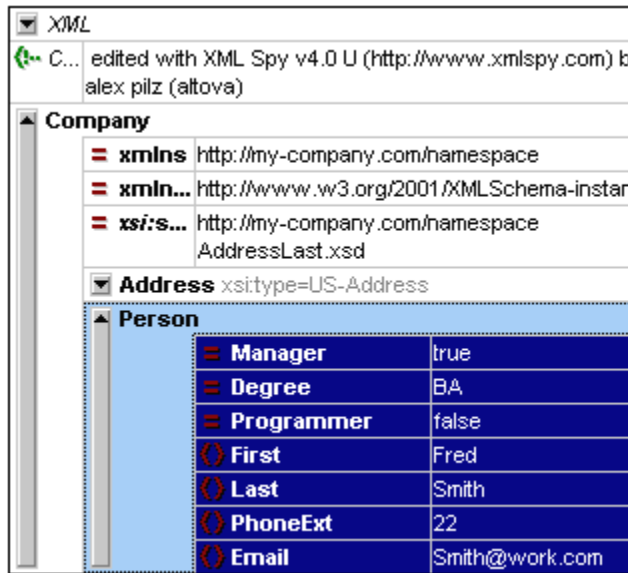
Bitte beachten Sie: Ein XML-Dokument kann auch abgespeichert werden wenn es nicht gültig ist. Beim Speichern werden Sie gefragt, ob Sie das nicht korrekte Dokument speichern wollen. Mit **Save anyway** wird das Dokument in seinem aktuellen, ungültigen Status gespeichert.

5.6 Hinzufügen von Elementen und Attributen

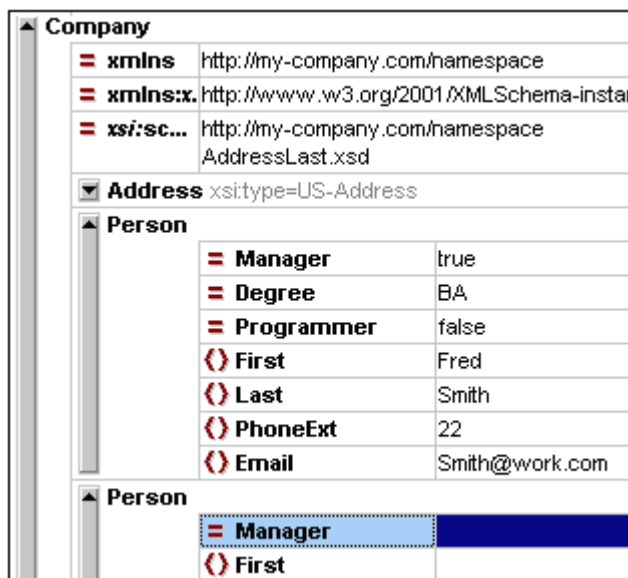
Momentan gibt es im Dokument nur ein `Person`-Element.

So fügen Sie ein neues Element `Person` hinzu:

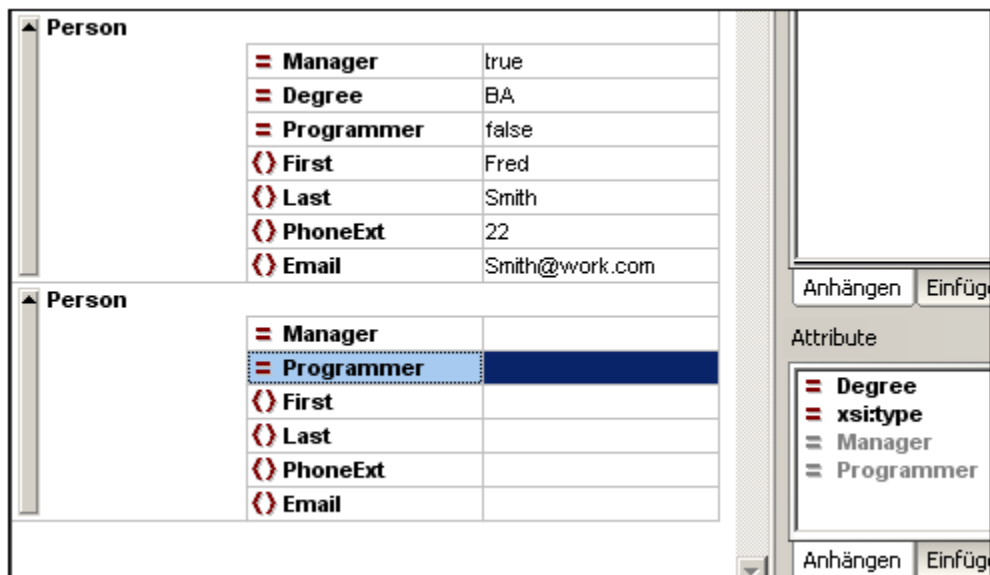
1. Klicken Sie auf die graue Bildlaufleiste links neben dem Element `Address`, um die `Address`-Elemente auszublenden und mehr Platz in der Ansicht zu schaffen.
2. Wählen Sie das gesamte Element `Person`, indem Sie in der Grid-Ansicht auf oder unter den Text des Elements `Person` klicken. Beachten Sie, dass das Element `Person` nun in der Element-Eingabehilfe auf dem Register "Anhängen" verfügbar ist.



3. Doppelklicken Sie in der Element-Eingabehilfe auf das Element `Person`. Ein neues `Person`-Element sowie alle obligatorischen Child-Elemente werden angehängt (siehe Abbildung unten). Beachten Sie, dass das optionale Child-Element `Title` nicht eingefügt wird.



4. Drücken Sie **F12** um in der Grid-Ansicht zum neuen Element `person` zu wechseln.
5. Klicken Sie auf das Attribut `Manager` im neuen Element `person`. Beachten Sie die vorhandenen Attribute in der Attribut-Eingabehilfe. Der Eintrag `Manager` ist ausgegraut, da er bereits verwendet wurde. Werfen Sie auch einen Blick auf das Info-Fenster, in dem nun Informationen über das Attribut `Manager` angezeigt werden.
6. Doppelklicken Sie in der Attribut-Eingabehilfe auf dem Register "Anhängen" auf den Eintrag `Programmer`. Hierdurch wird ein leeres `Programmer`-Attribut hinter dem `Manager`-Attribut eingefügt.




Das Attribut `Programmer` ist nun in der Attribut-Eingabehilfe ausgegraut.

Sie könnten in dieser Ansicht einen Inhalt für das Element `person` eingeben. Wechseln wir jedoch nun in die Datenbank/Tabellen-Ansicht der Grid-Ansicht, da sich diese besser zum Editieren einer Struktur mit mehreren Instanzen wie z.B. bei `person` eignet.

5.7 Editieren in der Datenbank/Tabellen-Ansicht

Die Grid-Ansicht enthält eine spezielle Ansicht namens Datenbank/Tabellen-Ansicht (in weiterer Folge Tabellenansicht genannt), die sich zum Bearbeiten von Elementen mit mehreren Instanzen eignet. Einzelne Elementtypen können als Tabelle angezeigt werden. Wenn ein Elementtyp als Tabelle angezeigt wird, werden dessen Children (Attribute und Elemente) als Spalten dargestellt, wobei die einzelnen Instanzen die Zeilen der Tabelle bilden.

Um einen Elementtyp als Tabelle anzuzeigen, wählen Sie eine beliebige Instanz dieses Elementtyps aus und klicken Sie in der Symbolleiste auf das Symbol "Als Tabelle anzeigen"  (**XML | Tabelle | Als Tabelle anzeigen**). Elementtypen einer untergeordneten Ebene, die mehrere Instanzen haben, werden ebenfalls in Tabellenform angezeigt. Die Tabellenansicht steht in der Enhanced Grid-Ansicht zur Verfügung und dient zum Bearbeiten jeder Art von XML-Datei (XML, XSD, XSL, usw.).

Vorteile der Tabellenansicht

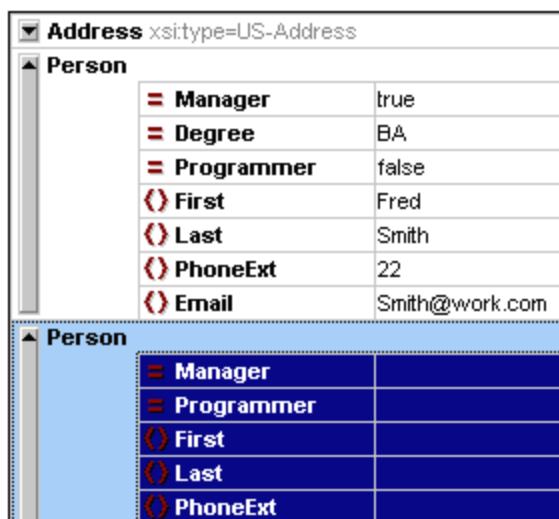
Die Tabellenansicht bietet die folgenden Vorteile:

- Sie können Spalten mit Drag & Drop untereinander austauschen. Im XML-Dokument selbst wird dadurch die relative Position von Child-Elementen oder Attributen für alle Element-Instanzen, die den Zeilen der Tabelle entsprechen, geändert.
- Sortieren von Daten nach Spalten mittels der Funktion **XML | Tabelle | Aufsteigend Sortieren** oder **Absteigend sortieren**.
- Einfügen oder Anhängen von Zeilen mittels der Funktion **XML | Tabelle | Zeile Einfügen**
- Kopieren und Einfügen von **strukturierten Daten** von und in Produkte von Drittanbietern
- Auch die intelligenten Eingabehilfen stehen in der Tabellenansicht zur Verfügung.


Anzeigen eines Elementtyps als Tabelle

So zeigen Sie das Element `Person` in Tabellenform an:

1. Wählen Sie in der Grid-Ansicht eines der `Person`-Elemente aus, indem Sie auf oder in die Nähe des `Person`-Texts klicken.





Person	
Manager	true
Degree	BA
Programmer	false
First	Fred
Last	Smith
PhoneExt	22
Email	Smith@work.com

2. Wählen Sie die Menüoption **XML | Tabelle | Als Tabelle anzeigen** oder klicken Sie auf das Symbol . Beide `Person`-Elemente werden in Form einer einzigen Tabellen

angezeigt. Die Element- und Attributnamen werden in der Kopfzeile der Tabelle, die einzelnen Elementinstanzen in den Tabellenzeilen angezeigt.

	Manager	Degree	Programmer
1	true	BA	false
2			

- Wählen Sie die Menüoption **Ansicht | Optimale Breite** oder klicken Sie auf das Symbol , um die Spaltenbreite zu optimieren.

Bitte beachten Sie: Sie können für einzelne Elementtypen die Tabellenansicht ein- oder ausblenden. Wählen Sie dazu die Tabelle aus (klicken Sie auf den Elementnamen in der Tabelle) und klicken Sie anschließend auf das Symbol "Als Tabelle anzeigen" . Beachten Sie jedoch, dass Child-Elemente, die als Tabellen angezeigt wurden, weiterhin als Tabellen angezeigt werden.

Eingabe von Inhalt in der Tabellenansicht

Um Inhalt für das zweite Person-Element einzugeben, doppelklicken Sie in jede der Tabellenzellen in der zweiten Zeile und geben Sie Daten ein. Beachten Sie jedoch, dass `PhoneExt` eine Ganzzahl bis 99 sein muss, damit die Datei gültig ist. Die intelligenten Editierfunktionen sind auch innerhalb der Tabellenzellen aktiv, sodass Sie Optionen aus Dropdown-Listen auswählen können, falls Optionen verfügbar sind (Boolscher Inhalt und die Enumerationen für das Attribut `Degree`).

	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	true	BA	false	Fred	Smith	22	Smith@work.com
2	false	MA	true	Alfred	Aldrich	33	Aldrich@work

Bitte beachten Sie: Die Eingabehilfen sind auch für die als Tabelle dargestellten Elemente und Attribute aktiv. Wenn Sie z.B. in der Element-Eingabehilfe auf den Eintrag `person` doppelklicken, wird eine neue Tabellenzeile hinzugefügt (D.h. es wird eine neue Instanz zum Element `person` hinzugefügt).

Kopieren von XML Daten aus und in Drittprodukte

In XMLSpy können Kalkulationstabellendaten aus und in Produkte anderer Anbieter schnell und einfach kopiert werden. Die kopierten Daten können in XMLSpy sowie in Produkten anderer Anbieter in nativem Format der Anwendung verwendet werden. In diesem Abschnitt erfahren Sie, wie man Daten in und aus einem Excel-Datenblatt kopiert. Gehen Sie dazu folgendermaßen vor:

- Klicken Sie auf die Zeilennummerierung `1`, halten Sie die **Strg**-Taste gedrückt und klicken Sie auf die Zeilennummerierung `2`. Hierdurch werden beide Zeilen markiert.

Address xsi:type=US-Address						
Person (2)						
	= Manager	= Degree	= Programmer	() First	() Last	
1	true	BA	false	Fred	Smith	
2	false	MA	true	Alfred	Aldrich	

- Wählen Sie die Menüoption **Bearbeiten | Als strukturierten Text kopieren**. Der Befehl kopiert die Elemente in die Zwischenablage, und zwar genauso, wie sie am Bildschirm erscheinen.
- Öffnen Sie Excel und fügen Sie die XML Daten (mit **Strg+V**) in ein Excel-Datenblatt ein.

A	B	C	D	E	F	G	H
TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com	
FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work.com	

- Fügen Sie eine neue Zeile im Excel-Arbeitsblatt ein und geben Sie eine **dreistellige Zahl** im Element `PhoneExt` ein (z.B. 444).

A	B	C	D	E	F	G	H
TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com	
FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work.com	
TRUE	PhD	FALSE	Colin	Coletti	444	Coletti@work.com	

- Markieren Sie die Daten in Excel, wählen Sie **Bearbeiten | Kopieren**, um die Daten in die Zwischenablage zu kopieren
- Wechseln Sie wieder zu XMLSpy.
- Klicken Sie in XMLSpy in die oberste linke Zelle der Tabelle und wählen Sie **Bearbeiten | Einfügen**.

Address xsi:type=US-Address						
Person (3)						
	= Manager	= Degree	= Programmer	() First	() Last	() Pho
1	TRUE	BA	FALSE	Fred	Smith	22
2	FALSE	MA	TRUE	Alfred	Aldrich	33
3	TRUE	PhD	FALSE	Colin	Coletti	444

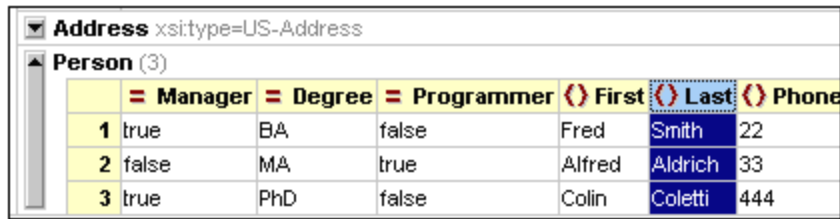
- Die aktualisierten Daten wurden in die Tabelle übernommen.
- Ändern Sie mittels der Menüoption **Suchen | Ersetzen** (Funktionstaste **Strg+H**) die Booleschen Werte `TRUE/FALSE` in `true/false` (in Kleinbuchstaben).

Sortieren der Tabellen nach Spalteninhalt


In der Tabellenansicht können Sie Ihre Daten je nach Spalte in aufsteigender oder absteigender Reihenfolge sortieren. Wir möchten nun die Tabelle `Person` alphabetisch nach Nachnamen sortieren.

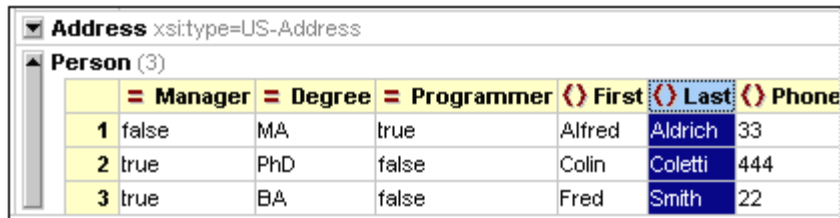
So sortieren Sie eine Tabelle nach dem Inhalt einer Spalte:

- Klicken Sie auf die Kopfzeile in der Spalte **Last**.



	= Manager	= Degree	= Programmer	First	Last	Phone
1	true	BA	false	Fred	Smith	22
2	false	MA	true	Alfred	Aldrich	33
3	true	PhD	false	Colin	Coletti	444

2. Wählen Sie die Menüoption **XML | Tabelle | Aufsteigend sortieren** oder klicken Sie auf das Symbol . Die Spalte wird unter Einbeziehung der gesamten Tabelle alphabetisch sortiert. Die Spalte bleibt hierbei markiert.



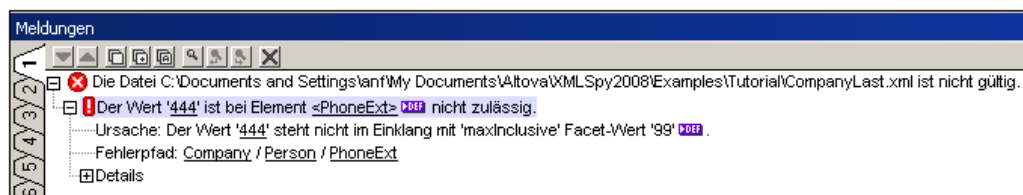
	= Manager	= Degree	= Programmer	First	Last	Phone
1	false	MA	true	Alfred	Aldrich	33
2	true	PhD	false	Colin	Coletti	444
3	true	BA	false	Fred	Smith	22

Die Tabelle wird nicht nur in der Ansicht sortiert, sondern die Daten werden auch im zugrundeliegenden XML-Dokument sortiert, d.h. die `Person`-Elemente werden nun alphabetisch nach dem Inhalt der Spalte `Last` sortiert. (Klicken Sie auf das Register "Text", um die Änderungen in der Textansicht zu sehen).

3. Wählen Sie die Menüoption **XML | Validieren** oder drücken Sie die Funktionstaste **F8**. Es wird eine Fehlermeldung angezeigt, dass der Wert '444' bei einem `PhoneExt`-Elements nicht zulässig ist (*siehe Abbildung*). Das fehlerverursachende `PhoneExt`-Element erscheint markiert.

Erweitern Sie "Details", um zu sehen, dass `PhoneExt` nicht gültig ist, weil es nicht kleiner oder gleich dem Maximalwert 99 ist.

Bitte beachten Sie: Sie können auf die Links in der Fehlermeldung klicken, um zu der Stelle in der XML-Datei zu gelangen, an der der Fehler gefunden wurde.




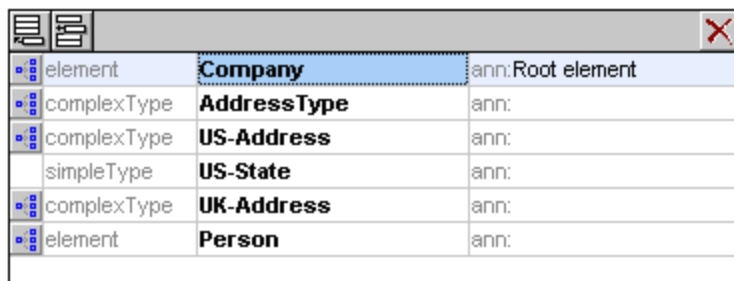
Da der für Durchwahlnummern festgelegte Wertebereich diese Durchwahl nicht mehr enthält, müssen wir das XML-Schema so ändern, dass diese Nummer gültig ist. Dies wird im nächsten Abschnitt beschrieben.

5.8 Ändern des Schemas


Wir haben nun gesehen, dass zweistellige Telefondurchwahlnummern nicht ausreichend sind. Wir werden daher den Wertebereich gültiger Nummern auf drei Stellen erweitern. Sie können das zugrunde liegende XML-Schema öffnen und ändern, ohne Ihr XML-Dokument schließen zu müssen.

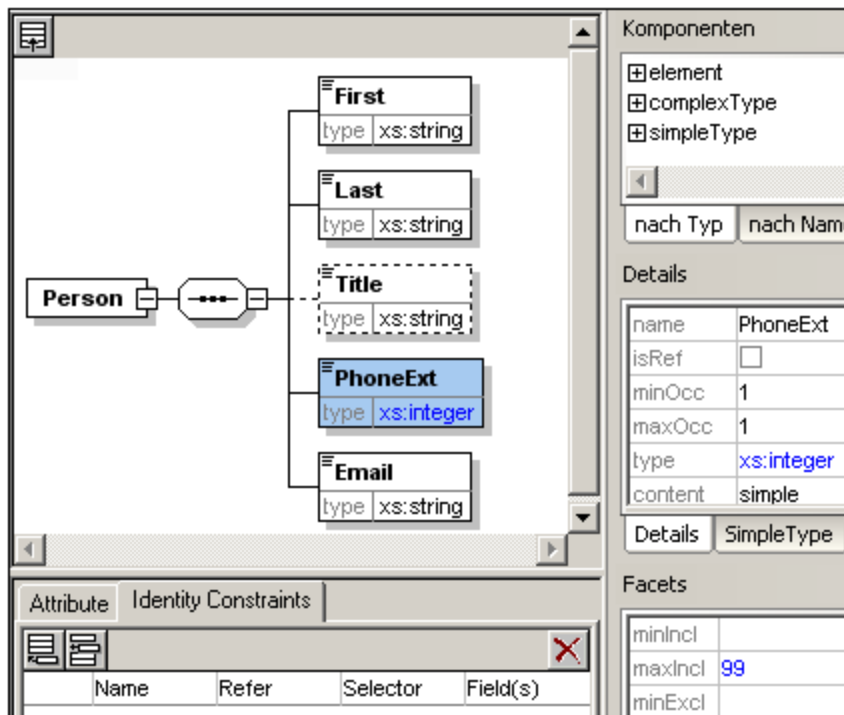
Gehen Sie folgendermaßen vor:

1. Wählen Sie die Menüoption **DTD/Schema | Gehe zu Definition** oder klicken Sie auf das Symbol . Das assoziierte Schemadokument (in diesem Fall `AddressLast.xsd`) wird geöffnet. Wechseln Sie in die Schema/WSDL-Ansicht (siehe Abbildung unten).



Category	Name	Annotation
element	Company	ann:Root element
complexType	AddressType	ann:
complexType	US-Address	ann:
simpleType	US-State	ann:
complexType	UK-Address	ann:
element	Person	ann:

2. Klicken Sie auf das Symbol "Diagramm anzeigen"  des globalen Elementes `Person` und klicken anschließend auf das Element `PhoneExt`. Die Facet-Daten werden auf dem Register "Facets" angezeigt.



The diagram shows the `Person` element containing `First`, `Last`, `Title`, `PhoneExt`, and `Email` elements. The `PhoneExt` element is highlighted, and its facets are displayed in the right-hand panel.

Komponenten

- element
- complexType
- simpleType

nach Typ | nach Name

Details

name	PhoneExt
isRef	<input type="checkbox"/>
minOcc	1
maxOcc	1
type	xs:integer
content	simple

Details | SimpleType

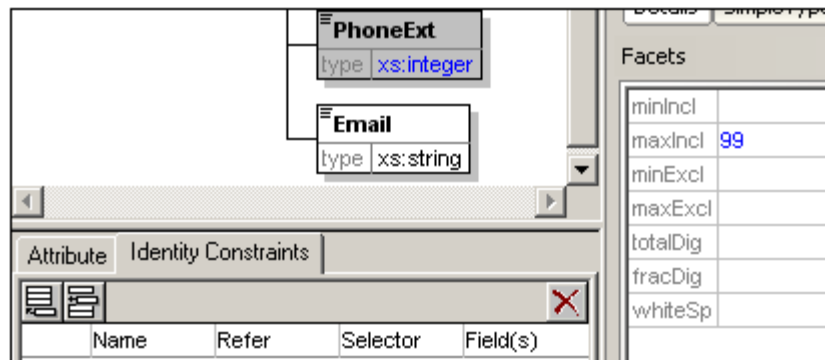
Facets


minIncl	
maxIncl	99
minExcl	

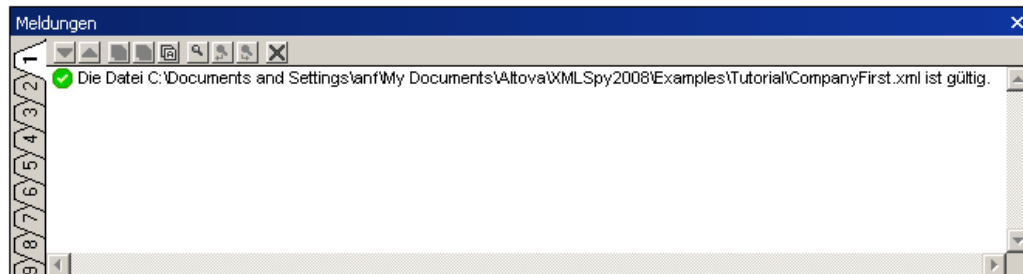
Attribute | Identity Constraints

Name	Refer	Selector	Field(s)

3. Doppelklicken Sie auf dem Register "Facets" auf das Wertefeld `maxIncl`, geben Sie 999 als Maximalwert ein und drücken Sie die Eingabetaste.



4. Speichern Sie das Schema-Dokument.
5. Mit **Strg+TAB** kommen Sie zurück zum XML-Dokument.
6. Klicken Sie auf die Schaltfläche , um das XML-Dokument erneut zu validieren.



Die Meldung "Diese Datei ist gültig" wird angezeigt. Das XML-Dokument ist nun mit der geänderten Schemadefinition konform.

7. Wählen Sie die Menüoption **Datei | Speichern unter...** und benennen Sie die Datei `CompanyLast.xml`. (Denken Sie daran, das Originaldokument `CompanyLast.xml` umzubenennen z.B. in `CompanyLast_orig.xml`).

Bitte beachten Sie: Das Originaldokument `CompanyLast.xml` ist im Verzeichnis `Tutorial` abgelegt.

6 XSLT-Transformationen

Ihr Ziel in diesem Abschnitt

Erstellen einer HTML-Datei von einer XML-Datei mittels eines XSLT Stylesheet zum Transformieren der XML-Datei. Beachten Sie, dass die XML-Datei bei einer "Transformation" nicht geändert wird. Es wird nur eine neue Ausgabedatei erzeugt. Das Wort "Transformation" ist eine Konvention.

Methode:

Gehen Sie beim Transformieren folgendermaßen vor:

- Weisen Sie dem XML-Dokument eine vordefinierte XSL-Datei, nämlich `company.xsl` zu.
- Führen Sie die Transformation mit Hilfe eines der beiden in XMLSpy integrierten Altova XSLT-Prozessoren durch (*siehe Hinweis unten*).

In diesem Abschnitt verwendete Befehle

Die folgenden XMLSpy-Befehle werden in diesem Abschnitt verwendet:



XSL/XQuery | XSL zuweisen weist dem aktiven XML-Dokument eine XSL-Datei zu.



XSL/XQuery | Gehe zu XSL öffnet die vom aktiven XML-Dokument referenzierte XSL-Datei.



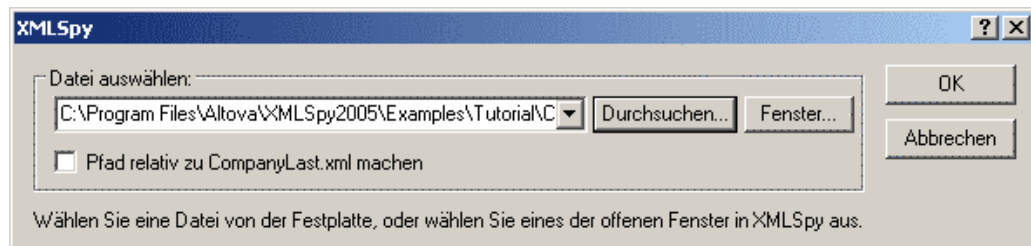
XSL/XQuery | XSL-Transformation (F10) oder die Symbolleistenschaltfläche transformiert das aktive XML-Dokument mit Hilfe des diesem zugewiesenen XSL Stylesheet. Wenn keine XSL-Datei zugewiesen wurde, werden Sie beim Aufrufen dieses Befehls aufgefordert, eine zuzuweisen.

Bitte beachten Sie: XMLSpy verfügt über zwei integrierte XSLT-Prozessoren, den Altova XSLT 1.0-Prozessor und den Altova XSLT 2.0-Prozessor. Der Altova XSLT 1.0-Prozessor dient zum Transformieren von XSLT 1.0 Stylesheets, während der Altova XSLT 2.0-Prozessor für die Verarbeitung von XSLT 2.0 Stylesheets verwendet wird. Der richtige Prozessor wird von XMLSpy automatisch auf Basis des Versionsattributs im Element `xsl:stylesheet` oder `xsl:transform` ausgewählt. In diesem Tutorial verwenden wir für die Transformation XSLT 1.0 Stylesheets. Bei Aufruf des Befehls **XSL-Transformation** wird bei diesen Stylesheets automatisch der Altova XSLT 1.0-Prozessor verwendet.

6.1 Zuweisen einer XSLT-Datei

So weisen Sie der Datei `CompanyLast.xml` eine XSLT-Datei zu:

1. Klicken Sie im Hauptfenster auf das Register `CompanyLast.xml`, um es aktiv zu setzen und wechseln Sie in die Textansicht.
2. Wählen Sie die Menüoption **XSL/XQuery | XSL zuweisen**.
3. Klicken Sie auf die Durchsuchen-Schaltfläche und wählen Sie im Ordner `Tutorial` die Datei `Company.xml` aus. Sie können im Dialogfeld die Option "Pfad relativ zu `CompanyLast.xml` machen", aktivieren, um den Pfad zur XSL-Datei (im XML-Dokument) relativ zu machen.



4. Klicken Sie auf OK um dem XML-Dokument die XSL-Datei zuzuweisen.
5. Wechseln Sie in die Grid-Ansicht, um die Zuweisung zu sehen (*Abbildung unten*).


XML	
Comment	edited with XML Spy v4.0 U (http://www.xmlspy.com) by alex pilz (altova)
xml-stylesheet	type="text/xsl" href="C:\Program Files\Altova\XML Spy Suite\Examples\Tutorial\Company.xml"
Company	
xmlns	http://my-company.com/hamespace
xmlns:x	http://www.w3.org/2001/XMLSchema
xsi:sc...	http://my-company.com/hamespace AddressLast.xsd
Address	
xsi:type	US-Address

Im XML-Dokument wird eine `XML-styleSheet` Verarbeitungsanweisung eingefügt, die die XSL-Datei referenziert. Wenn Sie das Kontrollkästchen "Pfad relativ zu `CompanyLast.xml` machen" aktiviert haben, ist der Pfad relativ; andernfalls ist er absolut (wie in der obigen Abbildung).

6.2 Transformieren der XML-Datei

Um das XML-Dokument mit Hilfe der zugewiesenen XSL-Datei zu transformieren, gehen Sie folgendermaßen vor:

1. Stellen Sie sicher, dass die XML-Datei das aktive Dokument ist.
2. Wählen Sie die Menüoption **XSL/XQuery | XSL-Transformation (F10)** oder klicken Sie

auf das Symbol , um die Transformation mittels des in dem XML-Dokument referenzierten XSL Stylesheet zu starten. (Da die Datei `Company.xml` ein XSLT 1.0-Dokument ist, wird für die Transformation automatisch der integrierte Altova XSLT 1.0-Prozessor ausgewählt). Das Ausgabedokument wird in der Browser-Ansicht angezeigt. Es hat den Namen `XSL Output.html`. (Wenn die HTML-Ausgabedatei nicht generiert wird, stellen Sie sicher, dass die Standarddateierweiterung im Dialogfeld "Optionen" (**Extras | Optionen**) auf dem Register "XSL" auf `.html` gesetzt wurde.) Die Firmendaten werden in einem Block links oben angezeigt, während die Personendaten unterhalb davon in Tabellenform dargestellt werden.



Your Company

Name: US dependency
Street: Noble Ave
City: Dallas
State: Texas
Zip: 04812

First	Last	Ext.	E-Mail	Manager	Degree
Alfred	Aldrich	33	Aldrich@work.com	false	MA
Colin	Coletti	444	Coletti@work.com	true	Ph.D
Fred	Smith	22	Smith@work.com	true	BA

Text | Browser

CompanyLast.xml | AddressLast.xsd | XSL Output.html

Bitte beachten Sie: Falls in der Ausgabedatei nur eine Tabellenkopfeile aber keine Tabellendaten zu sehen sind, stellen Sie bitte sicher, dass Sie den Target Namespace für Ihr Schema wie am Beginn des Tutorials unter [Definieren von Namespaces](#) beschrieben, definiert haben. Der Namespace muss in allen drei Dateien (Schema, XML und XSL) **identisch** sein.

6.3 Ändern der XSL-Datei

Sie können die Ausgabe der Datei ändern, indem Sie die zugrunde liegende XSL-Datei bearbeiten. In diesem Fall soll die Hintergrundfarbe der Tabelle von "Lime (Hellgrün)" in "Yellow (Gelb)" geändert werden.

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf das Register `CompanyLast.xml` um es aktiv zu setzen und stellen Sie sicher, dass Sie sich in der Grid-Ansicht befinden.
2. Wählen Sie die Menüoption **XSL/XQuery | Gehe zu XSL**.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:my="http://my-company.com/namespace">

<xsl:template match="/">
  <html>
    <head> <title>Your company</title></head>
    <body>
      <h1><center>Your Company</center></h1>
      <xsl:apply-templates select="//my:Address"/>
      <table border="1" bgcolor="lime">
        <thead align="center">
          <td><strong>First</strong></td>
          <td><strong>Last</strong></td>
          <td><strong>Ext.</strong></td>
```

Mit diesem Befehl wird die Datei `Company.XSL` geöffnet, die in der XML-Datei referenziert wird.

3. Gehen Sie zur Zeile `<table border="1" bgcolor="lime">` und ändern Sie den Eintrag von `bgcolor="lime"` in `bgcolor="yellow"`.

```
<h1><center>Your Company</center></h1>
<xsl:apply-templates select="//my:Address"/>
<table border="1" bgcolor="yellow">
  <thead align="center">
    <td><strong>First</strong></td>
    <td><strong>Last</strong></td>
```

4. Wählen Sie die Menüoption **Datei | Speichern**, um die Änderungen an der XSL-Datei zu speichern.
5. Klicken Sie auf das Register `CompanyLast.xml` um die XML-Datei aktiv zu setzen, und wählen Sie **XSL/XQuery | XSL-Transformation** oder drücken Sie die Funktionstaste **F10**. In der Browser-Ansicht von XMLSpy erscheint eine neue Datei namens `XSL_Output.html`. Die Hintergrundfarbe der Tabelle ist gelb.

Your Company

Name: US dependency
Street: Noble Ave
City: Dallas
State: Texas
Zip: 04812

First	Last	Ext.	E-Mail	Manager	Degree
Alfred	Aldrich	33	Aldrich@work.com	false	MA
Colin	Coletti	444	Coletti@work.com	true	Ph.D
Fred	Smith	22	Smith@work.com	true	BA

6. Wählen Sie die Menüoption **Datei | Speichern** und benennen Sie die Datei `Company.html`.

7 Projektverwaltung

In diesem Abschnitt werden Sie mit den Projektverwaltungsfunktionen von XMLSpy vertraut gemacht. Nachdem Sie die Vorteile der Gliederung von XML-Dateien in Projekten kennengelernt haben, werden wir die soeben erstellten Dateien in einem einfachen Projekt strukturieren.

7.1 Vorteile von Projekten

Im Folgenden sind die Vorteile der Strukturierung von XML-Dateien in Form von Projekten aufgelistet.

- Dateien und URLs können durch gemeinsame Erweiterungen oder andere Kriterien in Ordner gruppiert werden.
- Batchprozesse können bestimmten Ordnern oder dem ganzen Projekt zugewiesen werden.
- DTD / XML-Schemas können bestimmten Ordnern zugewiesen werden, um eine sofortige Validierung der darin befindlichen Dateien zu ermöglichen.
- XSL-Dateien können bestimmten Ordnern zugewiesen werden, um eine sofortige Transformation der darin befindlichen Dateien zu erlauben.
- Es können Zielordner für die XSL-Transformationsdateien definiert werden.

Alle Einstellungen können unter **Projekt | Projekteigenschaften...** definiert werden. Im nächsten Abschnitt werden Sie über das Menü "Projekt" ein Projekt erstellen.

Des Weiteren stehen die folgenden Projektfunktionen zur Verfügung:

- XML-Dateien können unter Versionskontrolle gestellt werden (**Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen...**). Weitere Informationen erhalten Sie im Abschnitt Versionskontrolle der Online-Hilfe.
- Persönliche, netzinterne und Webordner können in Projekte eingefügt werden, wodurch die Batchvalidierung ermöglicht wird.

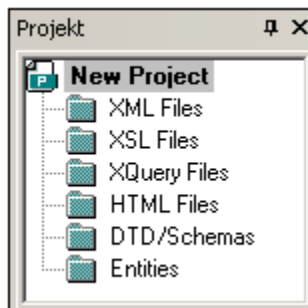
7.2 Erstellen eines Projekts

Im Hauptfenster ist nun eine Reihe von Tutorial-Dateien offen. Sie können diese Dateien in einem Tutorial-Projekt gliedern. Zuerst müssen Sie ein neues Projekt erstellen und anschließend die Tutorial-Dateien zu den jeweiligen Unterordnern des Projekts hinzufügen.

Erstellen eines einfachen Projekts

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

1. Wählen Sie die Menüoption **Projekt | Neu**. Im Projektfenster wird ein Projektordner mit dem Namen `New Project` erstellt. Das neue Projekt enthält leere Ordner für typische XML-Dateikategorien in einem Projekt (siehe Abbildung unten).



2. Klicken Sie auf das Register `CompanyLast.xml` um diese Datei im Hauptfenster aktiv zu setzen.
3. Wählen Sie die Menüoption **Projekt | Aktive und verwandte Dateien hinzufügen**. Es werden zwei Dateien zum Projekt hinzugefügt: `CompanyLast.xml` und `AddressLast.xsd`. Beachten Sie, dass durch Verarbeitungsanweisungen referenzierte Dateien (wie z.B. XSLT-Dateien) nicht als verwandte Dateien gelten.
4. Wählen Sie die Menüoption **Projekt | Projekt speichern** und speichern Sie das Projekt unter dem Namen `Tutorial`.

Hinzufügen von Dateien zu einem Projekt

Sie können auch andere Dateien zum Projekt hinzufügen. Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie auf eine beliebige offene XML-Datei (mit der Erweiterung `.xml`) mit Ausnahme von `CompanyLast.xml`, um diese XML-Datei zur aktiven zu machen. (Wenn keine andere XML-Datei geöffnet ist, öffnen Sie eine oder erstellen Sie eine neue XML-Datei).
2. Wählen Sie die Menüoption **Projekt | Aktive Datei hinzufügen**. Die XML-Datei wird anhand ihres Dateityps (`.xml`) in den Ordner XML Files hinzugefügt.
3. Fügen Sie nun auf dieselbe Art eine HTML-Datei und eine XSD-Datei (z.B. `Company.html` und `AddressFirst.xsd`) zum Projekt hinzu. Diese Dateien werden zum Ordner HTML Files bzw. DTD/Schemas hinzugefügt.
4. Speichern Sie das Projekt, entweder durch Auswahl der Menüoption **Projekt | Speichern** oder durch Auswahl einer beliebigen Datei oder eines Ordners im Projektfenster und Klicken auf das Symbol "Speichern" in der Symbolleiste (oder **Datei | Speichern**).

Bitte beachten Sie: Alternativ dazu können Sie mit der rechten Maustaste auf einen Projektordner klicken und den Befehl "Aktive Datei hinzufügen" auswählen, um die aktive Datei zu diesem bestimmten Ordner hinzuzufügen.

Andere nützliche Befehle

Hier sind einige weitere gebräuchliche Projektbefehle aufgelistet:

- Um einen neuen Ordner zu einem Projekt hinzuzufügen, wählen Sie die Menüoption **Projekt | Projektordner in Projekt einfügen** und fügen Sie den Namen des Projektordners ein.
- Um einen Ordner aus einem Projekt zu löschen, rechtsklicken Sie auf den Ordner und wählen Sie im Kontextmenü den Befehl **Löschen**.
Um eine Datei aus einem Projekt zu löschen, wählen Sie die Datei aus und drücken Sie die **Entf**-Taste.

8 **Fertig!**

Herzlichen Glückwunsch!

Wir hoffen, dass Ihnen diese Anleitung geholfen hat, um die Grundzüge von XMLSpy kennen zu lernen. Weitere Informationen finden Sie in der kontextsensitiven Online-Hilfe. Oder drucken Sie die PDF-Version des Tutorials aus, die sich im Anwendungsordner von XMLSpy befindet (tutorial.pdf).

Index

A

Attribute,

- in der Content Model-Ansicht ein- und ausblenden, 32
- in Schemadefinitionen, 32

B

Beispieldateien,

- Tutorial, 1

C

ComplexType,

- Definition erweitern, 22

ComplexTypes,

- in Schema-Definitionen, 22

Content Model,

- Attribute ein- und ausblenden, 32
- einfaches Content Model erstellen, 9

Content Model-Ansicht, 4

D

Database/Table View,

- how to use, 59

Details-Eingabehilfe, 9

Dokumentieren,

- Schema, 39

Drag & Drop,

- Elemente hinzufügen, 14

E

Eingabehilfe,

- Details, 9

Eingabehilfen, 2

- in Grid-Ansicht, 57

Elemente,

- Inhalt einschränken, 18
- optional machen, 18

Elementtyp,

- in XML-Dokument festlegen, 46

Enhanced Grid-Ansicht,

- siehe Grid-Ansicht, 48

Enumerationen,

- für Attribute definieren, 32

Erstellen,

- Neues XML Dokument, 44

F

Fenster,

- Übersicht, 2

G

Globale Elemente,

- in XML-Schemas verwenden, 30

Grid View,

- and Table View, 59

Grid-Ansicht,

- Attribut / Element anhängen, 57
- Dateneingaben, 48
- Eingabehilfen verwenden, 57

H

Hauptfenster, 2

I

Identity Constraint,

- in der Content Model-Ansicht ein- und ausblenden, 32

Info,

- Fenster, 2

Infofenster, 2

Instanzen,

Anzahl, 9

K

Komponenten-Definitionen,

wiederverwenden, 22

Kompositor,

für Sequences, 9

N

Namespaces,

in Schemas, 7

Navigieren,

Kürzel in Schema, 36

Neues XML-Dokument,

erstellen, 44

O

Optionale Elemente,

erstellen, 18

P

Projekte in XMLSpy,

erstellen, 72

Vorteile, 71

Projektfenster, 2

Projektverwaltung in XMLSpy, 70

S

Schema,

Dokumentation, 39

siehe auch XML-Schema, 3

Schema-Ansicht,

konfigurieren, 16

Schema-Übersicht, 4

Sequence-Kompositor,

verwenden, 9

SimpleTypes,

in Schema-Definitionen, 22

T

Table View,

how to use, 59

Textansicht,

Bearbeiten in, 49

Tutorial,

Beispieldateien, 1

Ziele, 1

type,

Erweiterung im XML-Dokument, 46

U

Übersicht, 2

V

Validieren,

XML-Dokument, 53

Vorlagen,

Ordner, 1

W

Wohlgeformtheitsprüfung,

bei XML-Dokument, 53

X

XML,

Dokument, 43

Neues Dokument, 44

Symbole, 43

XML-Dokument,

in Textansicht bearbeiten, 49

XML-Schema,

Ansicht konfigurieren, 16

Erstellen eines einfachen Schemas, 3

Komponenten hinzufügen, 9

Namespaces definieren, 7

Navigation in der Design-Ansicht, 36

neues erstellen, 4

Tutorial, 3

während der Bearbeitung des XML-Dokuments ändern, 63

xsi:type,

Verwendung, 46

XSLT,

in XMLSpy ändern, 68

XSL-Transformation,

siehe XSLT, 65

XSLT-Transformation,

in XMLSpy, 67

Tutorial, 65

XSLT-Datei zuweisen, 66