Tutorial

© 1998-2001 Altova GmbH & Altova, Inc.

ALTOVA
www.altova.com

XML Spy IDE 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.

Permission is hereby granted to print the electronic form of this document (PDF file) for the sole purpose of using it to work with the XML Spy software obtained under separate license agreement.

Printed: 24 May 2002

© 2002 Altova

Table of Contents

Part I XML SPY Tutorial 1

1 XML Spy Overview 2

2 Creating a schema from scratch 3

Creating a new Schema file 4

Adding elements to a schema 7

Adding elements with drag and drop 10

Configuring the schema view 11

Completing the basic schema 12

3 Making schema components reusable 15

Globals, extending simple and complex types 16

References, attributes and enumerations 22

Navigation shortcuts in schema documents 25

4 Generating Schema documentation 28

5 Creating an XML document 32

Creating and completing a new XML file 33

Editing in Text- and Enhanced Grid view 36

Validating and entering data 40

Manipulating data - Entry helpers 44

Database/Table view 47

6 XSL Transformation 53

Transforming XML to HTML 54

7 Importing and exporting database data 57

Exporting XML data to external databases 58

Importing database data 60

8 Creating a database schema 63

Converting a database to a schema 64

9 Creating a project 68

Adding files to a project 69

10 That's it ! 71

Index 72

1 XML SPY Tutorial

This tutorial gives a short overview of XML, and takes you through several tasks which provide an overview of how to use XML Spy to its fullest.

You will learn how to:

- Create a **simple schema** from scratch
- **Generalize** the schema using simple and complex types
- Create schema **documentation**
- Create an **XML document** based on the schema file
- Copy data XML data to a **third party product** (Excel) and reinsert it in XML Spy
- **Validate** the XML document against its schema
- **Update** Schema settings while editing the XML document
- **Transform** the XML document into HTML using XSLT, and see the result in the Browser view
- **Import** and **export** database data to and from XML Spy
- Create a **schema** from a MS Access database
- Create an XML Spy **project** to organize all your XML documents

XML Spy installation and configuration
This tutorial assumes that you have successfully installed XML Spy on your computer and received a free evaluation key-code, or are a registered user of XML Spy.

The evaluation version of XML Spy is fully functional but time-limited to 30 days. You can request a regular license from our secure web server or through any one of our resellers.

Tutorial example files
The tutorial files are available in the **...\XML Spy Suite\Examples\Tutorial** folder.

Tutorial example files:

AddressFirst.xsd
AddressLast.xsd
CompanyFirst.xml
CompanyLast.xml
Company.xsl
Company.html
Company.mdb
DB2schema.mdb
DB2schema.xsd
Person-import.xml

The **Examples folder** contains various XML files for you to experiment with, while the **Tutorial** folder contains all the files used in this tutorial.

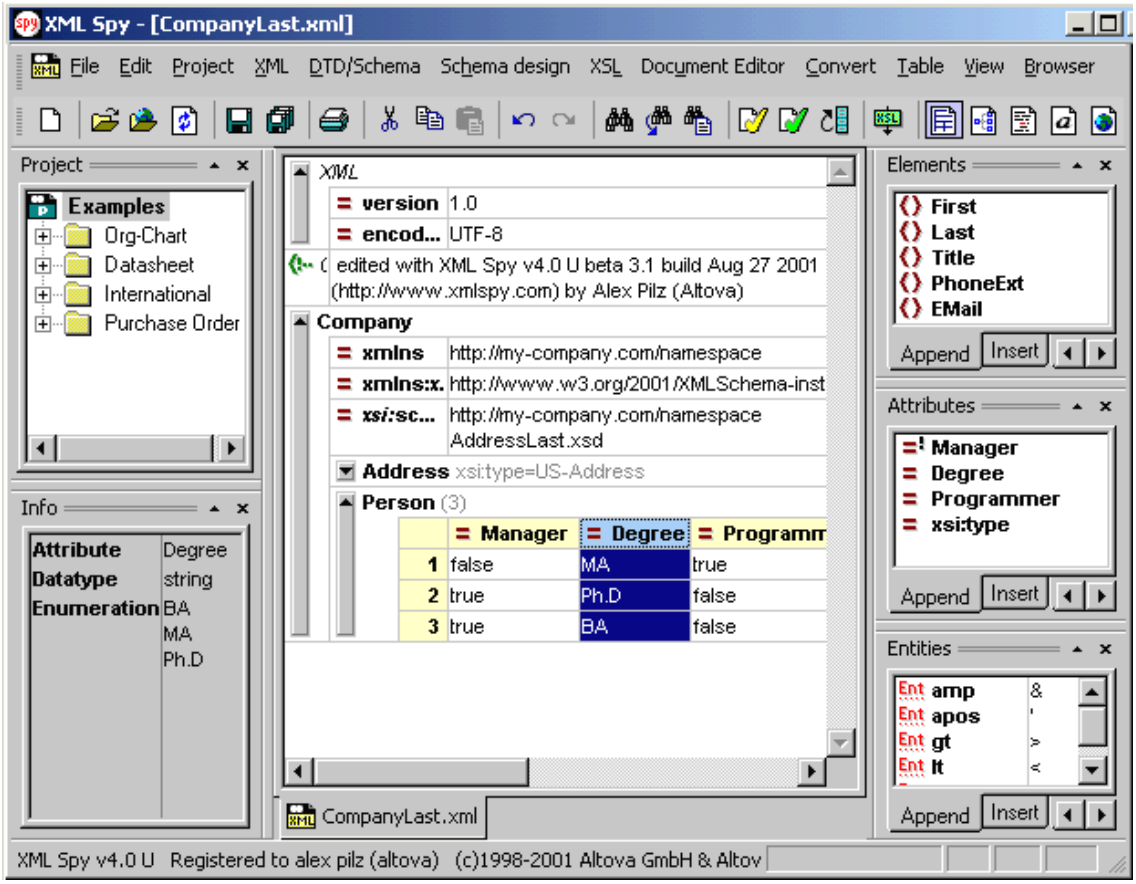
The **Template folder** contains all the XML template files that are used whenever you select the menu option **File | New**. These files supply the necessary data (namespaces and XML declarations) for you to start working with the respective XML document immediately.

© 2002 Altova

XML Spy Overview

XML Spy provides several windows that show various aspects of your XML document:

- The left area consists of the **Project** and **Info** windows.
- The central area, called **Main** window, is where you edit and view all types of XML documents.
  You can choose from different views: Enhanced Grid view, Schema view, Text view, Document Editor view or Browser view. The Enhanced Grid view incorporates a special view, called the Database/Table view, which collapses recurring XML data into table form.
- The right area contains the three **Entry helper** windows which allow you to insert or append: elements, attributes, and entities.



Creating a schema from scratch

A Schema describes what one or more XML documents can look like, and defines:

- The elements the document contains, and the order in which they appear
- The element content, and element attributes if any

The purpose of a schema is to allow machine validation of document structure. Instead of using the syntax of XML 1.0 DTD declarations, schema definitions use XML element syntax. A correct XML schema definition is, therefore, a well-formed XML document.

Goal of this section:
The goal of this section is to **create a simple schema** describing a company and its employees. The company is to consist of an **address** and an unlimited number of **persons**.

This will be achieved by:

- **Adding** elements to the schema
- Defining element **sequences**
- Adding **sub-elements** to an element (child elements)
- Creating elements using **drag and drop**
- **Configuring** the schema view
- Making an element **optional**
- Defining an element **facet**

Functions (and their icons) in this section:

File | New, creates a new type of XML file.

Schema design | Display diagram, the component icon displays the content model of the active global component.

Schema design | Display all globals, takes you back to the schema overview.

TAB Takes you to the next field and automatically opens a drop-down list if one exists.

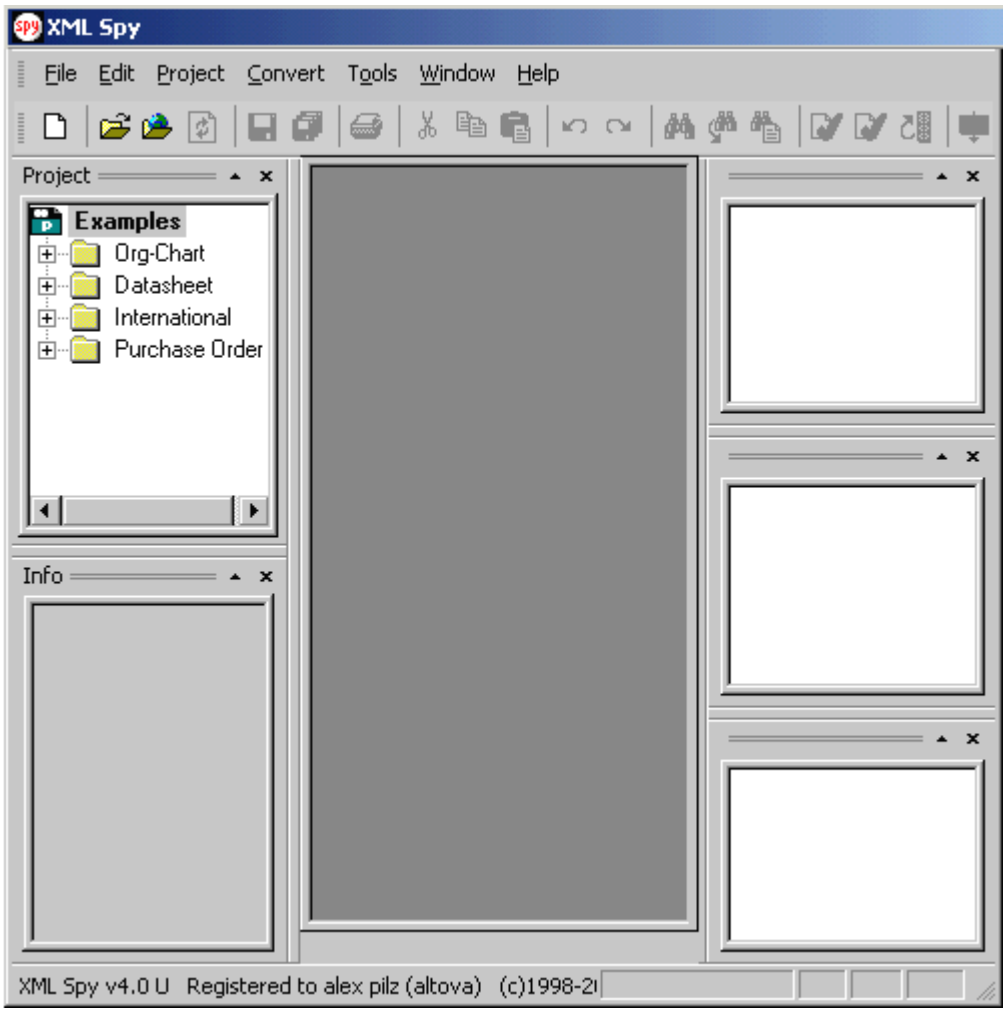
CTRL + Drag&Drop, enables you to copy existing elements.

Append icon, allows you to append an element to the schema or add a new line in the "View config." dialog (**Schema design | View config.**)

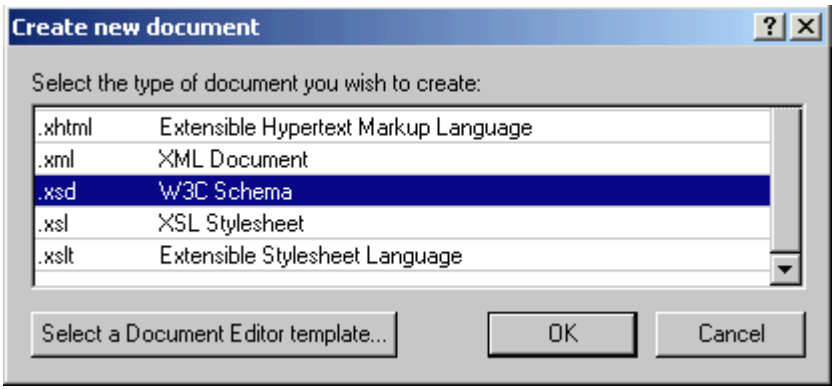
Creating a new Schema file

To create a new schema file:

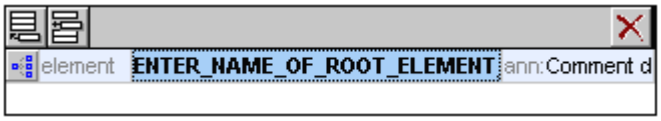
1. Start XML Spy by double clicking on the XML Spy icon.
   You are presented with an empty environment. There are no XML documents in the main window.



2. Select the menu option **File | New** and select the **.xsd W3C Schema** entry from the dialog and confirm with OK.

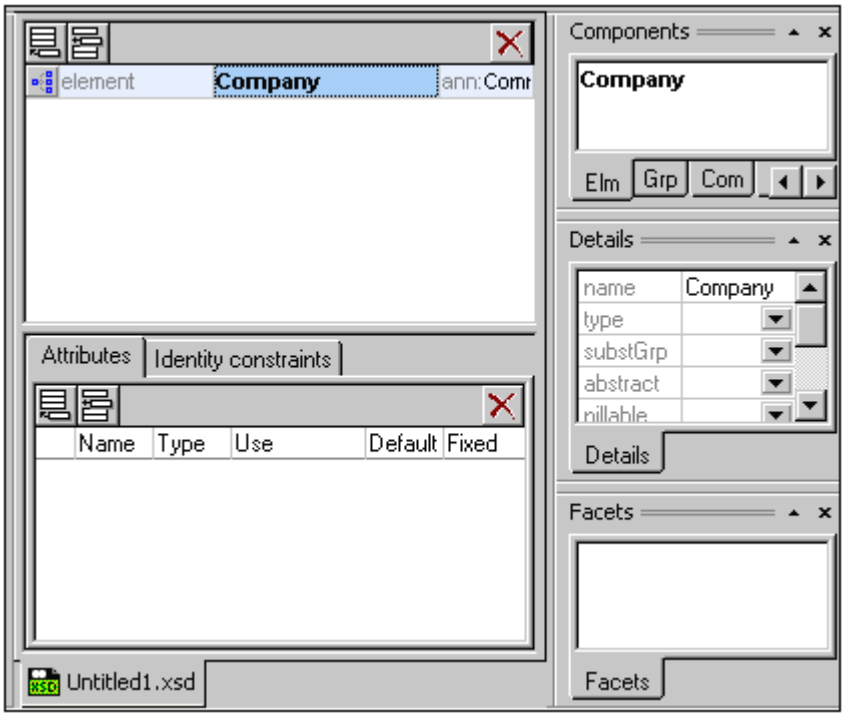


An empty schema file appears in the main window. You are prompted to enter the name of the root element.



3. Click in the highlighted field and enter "Company", confirm with Enter.
   Company is now the "root" element of this schema and is automatically a "global element" as well.

   This view is the **Schema overview** and displays the **global components** in the top window and the **attributes** of the currently selected component, in the lower one.

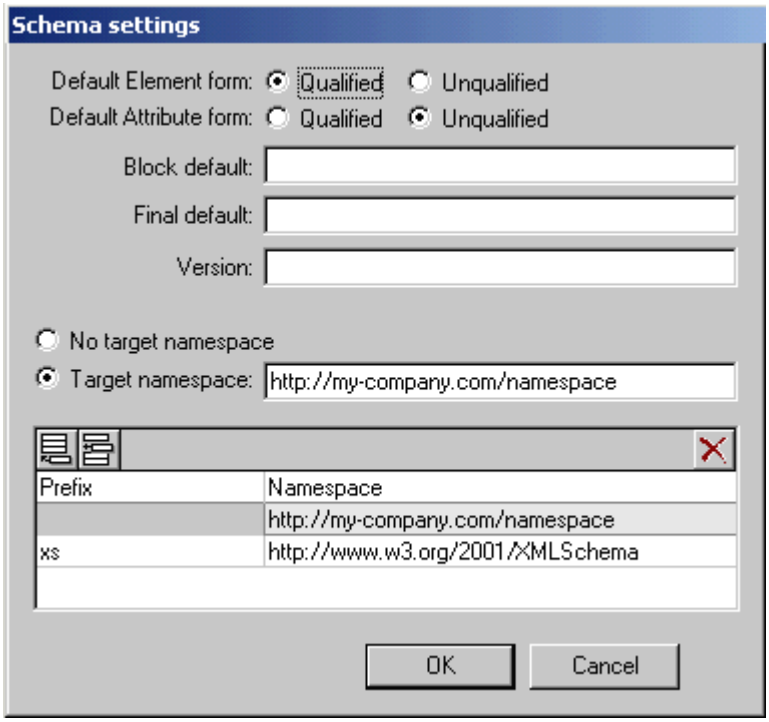


   The top entry helper window, the **Component Navigator**, displays Company in the "Elm" tab. The entries in these tabs can be used to navigate your schema by double clicking on them.
4. Click the menu option **File | Save as**, and name your schema (**AddressFirst** for

example).

Defining your own namespace:

1. Select the menu option **Schema Design | Schema settings.**
2. Click the Target namespace radio button, and enter **http://my-company.com/namespace.**

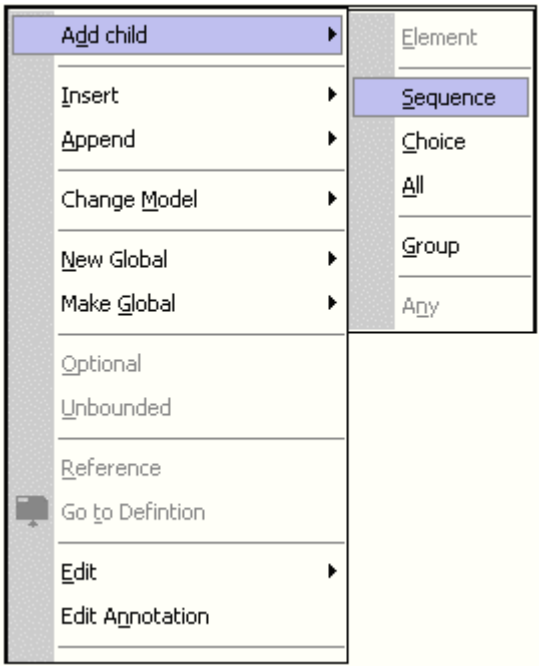


3. Confirm with the OK button.

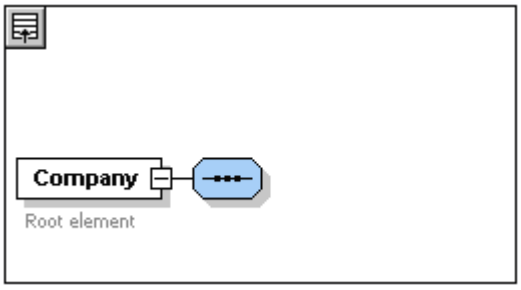
Adding elements to a schema

To add elements to a schema:

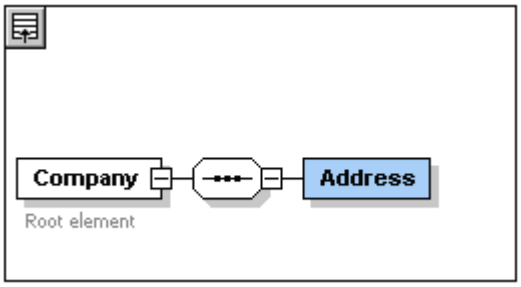
1. Click the component icon next to the **Company** element, in the main window, to display the content model (or double click on the Company entry in the Component Navigator).
   The text below the company element is annotation text. Double click the text if you want to edit it. (shortened to "Root element" here.)
2. Right click the Company element to open the context menu, and select **Add Child | Sequence**.



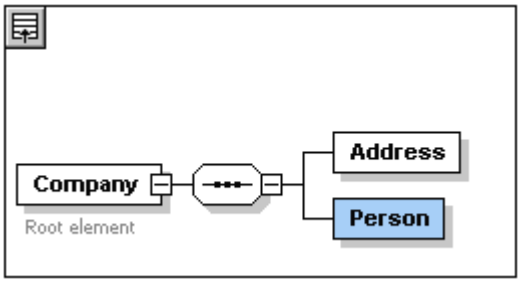
This inserts the **Sequence compositor**, and defines that the following elements must appear in the same sequence (in the XML document).



3. Right click the Sequence compositor and select **Add Child | Element**.
4. Enter "Address" as the name of the element, and confirm with the Enter key.

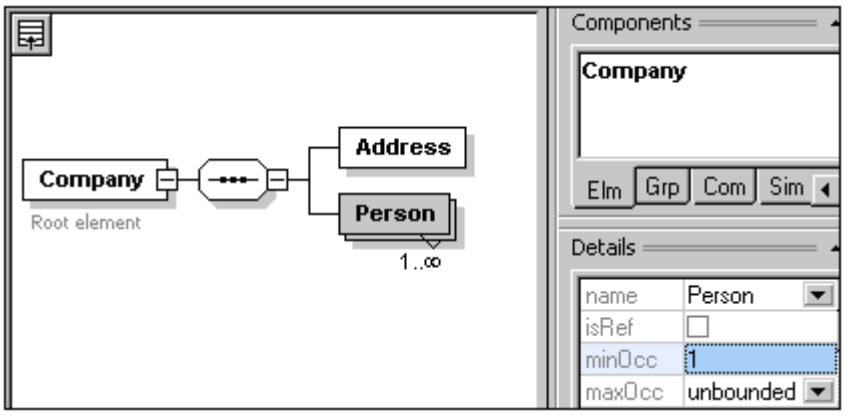


5. Right click the Sequence compositor again, select **Add Child | Element**, and enter "Person" as the name of the element.



We have now defined a schema which allows for one address and one person per company. As this is too restrictive, we want to make sure that we can include as many persons per company as necessary.

6. Right click the Person element, and select **Unbounded** from the context menu. The Person element changes at this point, showing the range in which it can occur, in this case 1 to infinity.



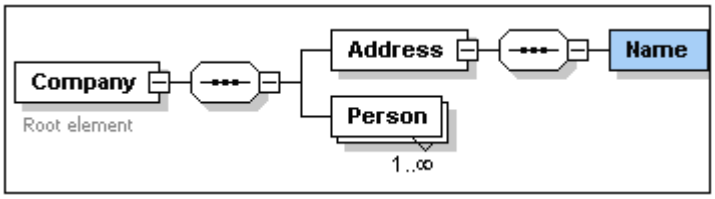
Please note:

You can also edit the **minOcc** and **maxOcc** fields in the Details entry helper directly.

We will now add the sub-elements which define the address structure.

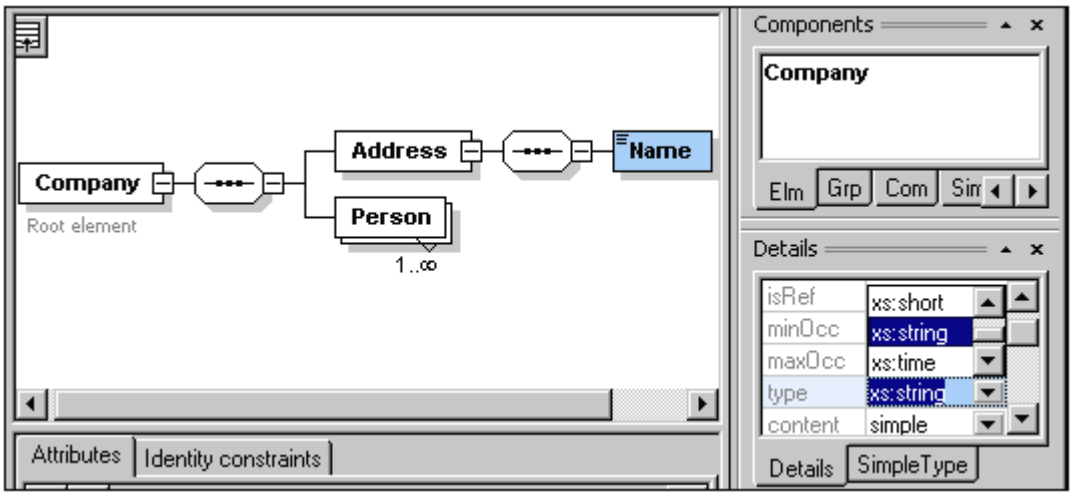
To add sub-elements to an element:

1. Right click the **Address** element to open the context menu, and select **Add Child | Sequence**.
2. Right click the **Sequence** compositor, and select **Add Child | Element**. Enter "Name" as the element name.



Defining element parameters:
At this point we want to define that the Name element is to occur only once, and contain only textual data.

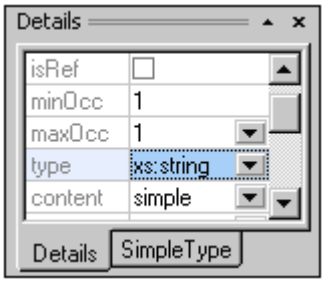
1. Click the Name element, if not currently selected.
2. Click on the **type** combo box of the middle entry helper, and select the entry **xs:string** from the drop down list.



This entry helper is called "**Details**" in the Schema view, and provides information on the currently selected element. **All data can be edited directly in the Details window!**

An icon appears in the top left of the element Name, indicating that this element contains text.

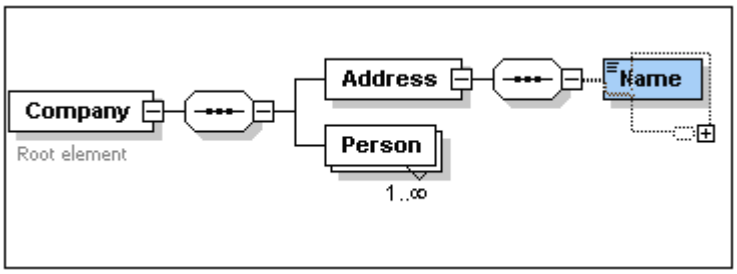
Both "minOcc" and "maxOcc" fields contain 1, showing that there is only one occurrence of this element (this is the default setting when creating a new element).



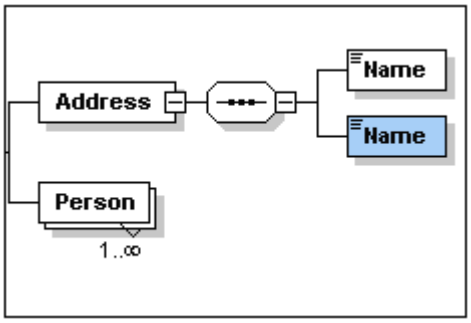
Adding elements with drag and drop

To add elements using drag and drop:
There is a quicker method of adding new elements to a schema, which avoids multiple menu commands:

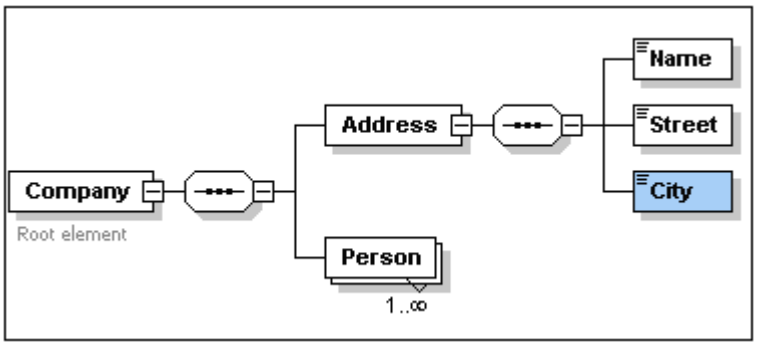
1. Click the Name element, hold down the **CTRL** key, and **drag** "slightly" with the mouse. A marquee with a small "plus" icon appears, showing that you are about to copy the element.



2. Release the mouse button to create the new element. If the new element appears somewhere else, just drag it near to the Name element and drop it there.
This method creates an element of the same type, with the same settings as the one copied.



3. Type "Street" to change the element name.
4. Use the same method to create a third element, "City".
The content model should now look like this:

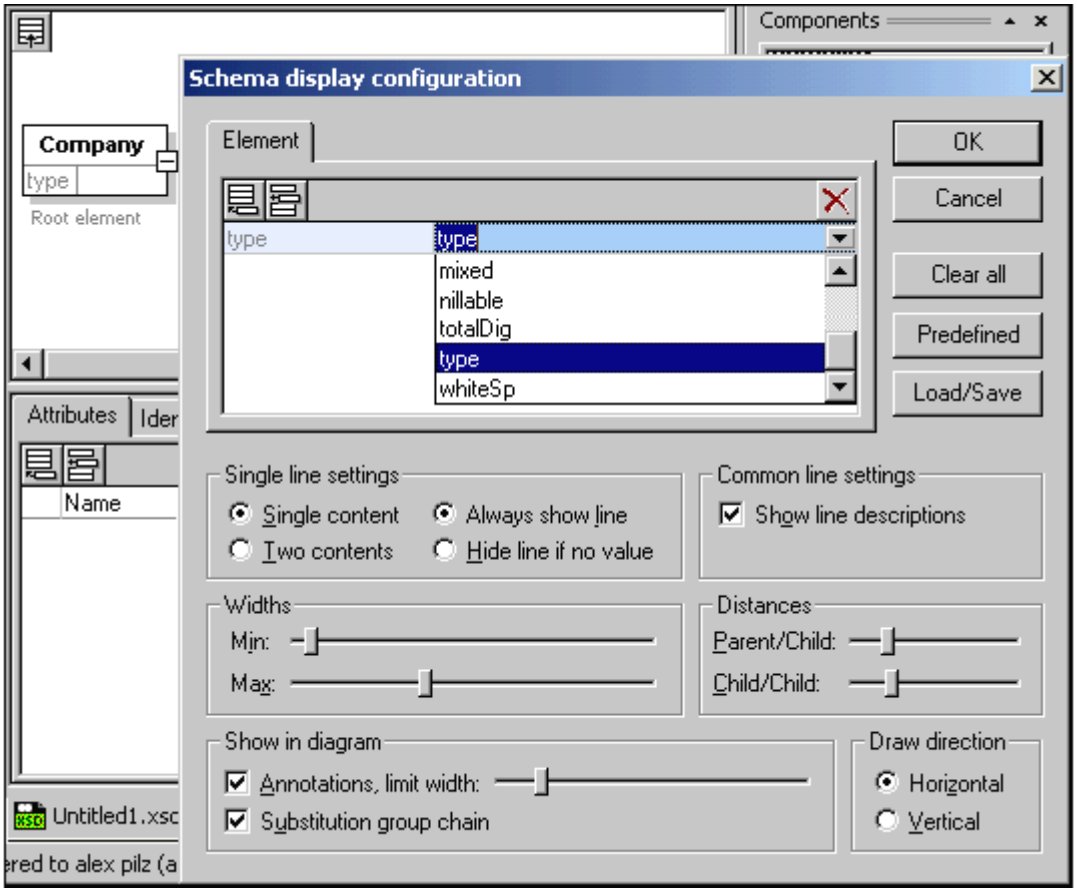


Configuring the schema view

Configuring the Schema view enables you to see specific settings in the content model, and also edit them directly.

To configure the schema view:

1. Select the menu option **Schema Design | View config**.
   A dialog opens at the bottom right of XML Spy, giving you room to see your selections appear in the content model immediately.
2. Click the **Append** icon to add a line, and use the combo box to select "**type**" from the drop-down list (or double click in the line and enter "type").



   The delete button deletes a line from the dialog.
   The dialog remains open, and each element symbol has been enlarged by a "type" field.
   The type field displays the element "type".
3. Click **OK** to confirm the changes.

Please note:
The settings you define here apply to the schema documentation output as well the printer output.

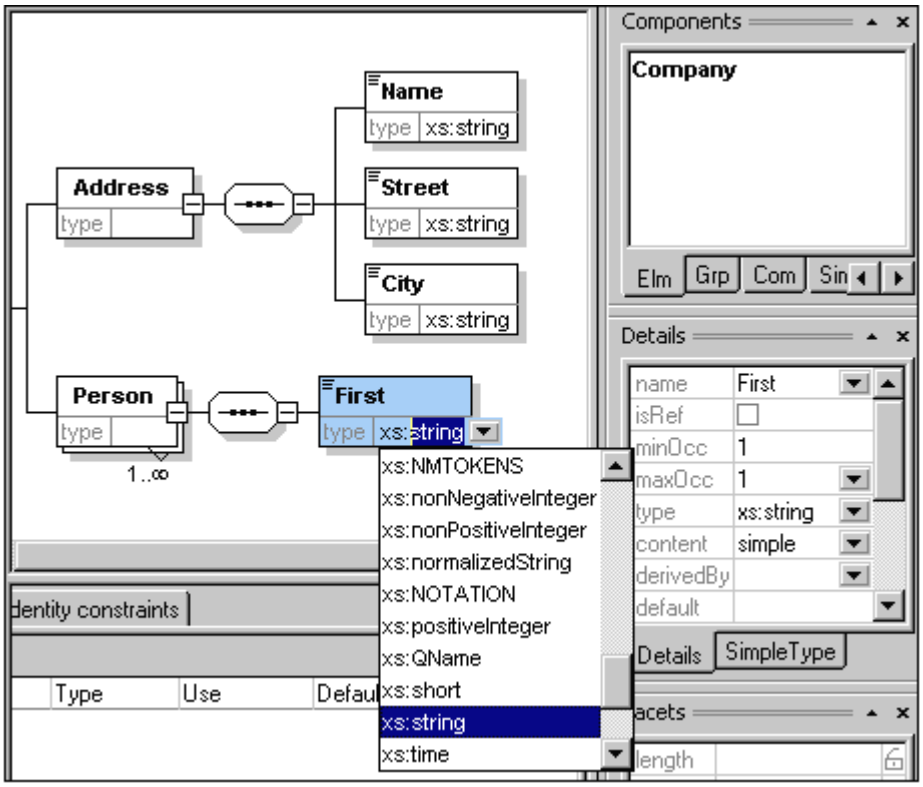
Completing the basic schema

At this point we want to add those sub elements to the Person element, which make up the personal data. All these elements will be **simple types** (with **simple content** models).

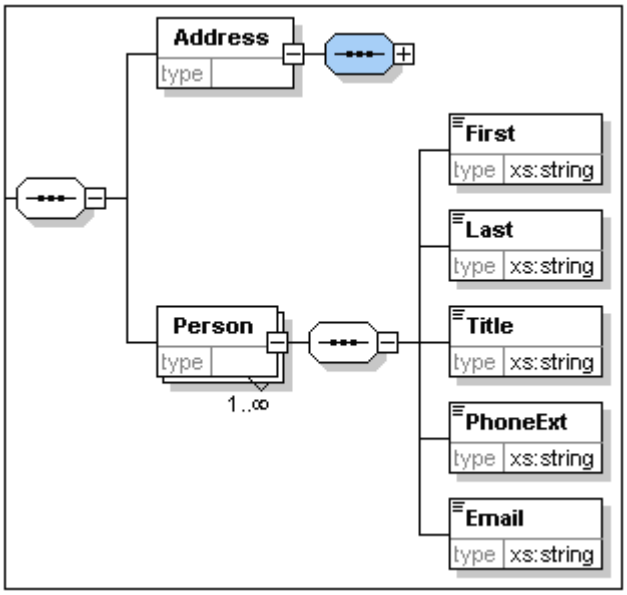
Person sub-elements: First, Last, Title, PhoneExt, and Email.

Requirements:
Title element: should be **optional**
PhoneExt: should be an **integer** and limited to **2 digits**

1. Right click the Person element to open the context menu, and select **Add Child | Sequence**. This inserts the Sequence compositor.
2. Right click the **Sequence** compositor, and select **Add Child | Element**.
3. Enter "**First**" as the name of the element, and hit the "**Tab**" key. This automatically places you in the **type** field.



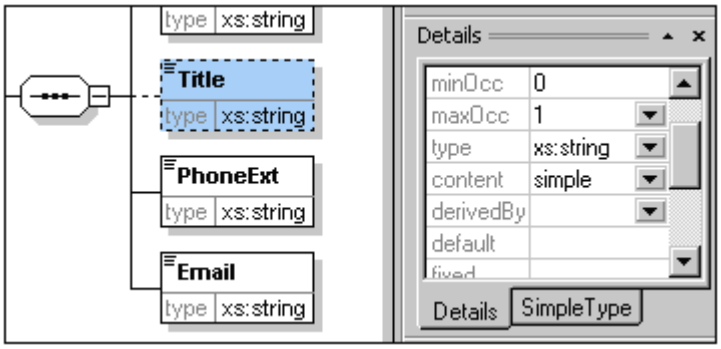
4. Select (or enter) the **xs:string** entry from the drop down list.
5. Use the drag and drop method to create **four more elements**, and name them: Last, Title, PhoneExt, and Email respectively.



Please note:
You can select multiple elements by holding down the CTRL key, and clicking each one.

To make an element optional:

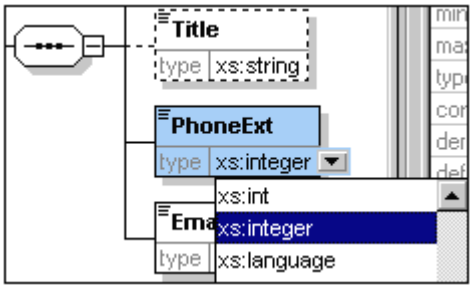
1. Right click the **Title** element, and select **Optional** from the context menu.
   The solid element frame changes to a dashed one; this is the visual display that an element is optional.



The "Details" fields have also been updated **minOcc=0** and **maxOcc=1**.

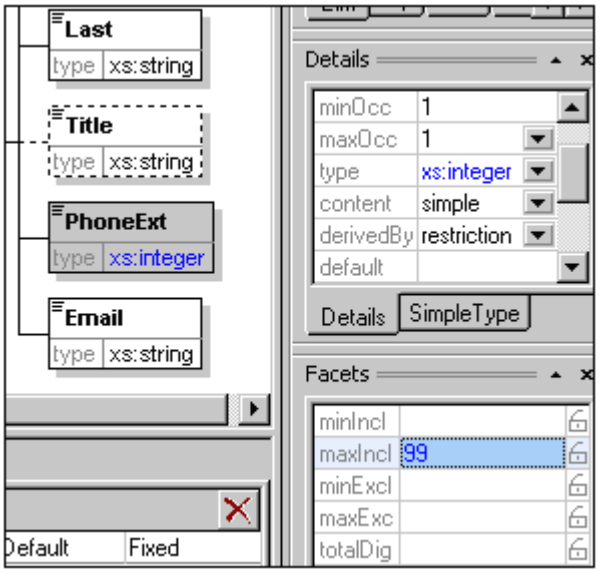
To limit the content of an element (Facets):

1. Double click in the **type** field of the PhoneExt **element**, and select (or enter) the **xs:integer** entry from the drop down list.



The items in the Facets tab (in the lowest entry helper) change at this point.

2. Double click in the "**maxIncl**" field of the Facets tab (in the lowest entry helper) and enter **99**, confirm with Enter.



This defines that all phone extensions up to, and including 99, are valid.

3. Select the menu option **File | Save** to save the changes to the schema.

Please note:

- Selecting a predefined simple type "text" (i.e. xs:string, xs:date etc.) for an element, automatically changes the content model to: content = simple, in the Details entry helper.
- Adding a compositor to an element (selection, choice or all), automatically changes the content model to: content = complex, in the Details entry helper.
- This schema is available as '**AddressFirst**' in the ..\Tutorial folder.

Making schema components reusable

Goal of this section:
To create generic **schema** components which can be reused by other elements.

This will be achieved by:

- Creating a global **AddressType** component, which will be the basis for specific country addresses (a complex type)
- Creating two specific address templates for UK-, and US Adresses by **extending** the global address element (extend the complex type)
- Creating a global US-State element, by **restriction** (simpleType)
- Creating a global person element by **reference**
- Defining person **attributes** that supply information about the persons position in the company
- **Limiting** the **attribute contents** to a predefined set of attribute values (enumeration)

Functions (and their icons) in this section:

Schema design | Display all globals, takes you back to the schema overview.

Append icon, allows you to append an element, attribute or enumeration to a schema.

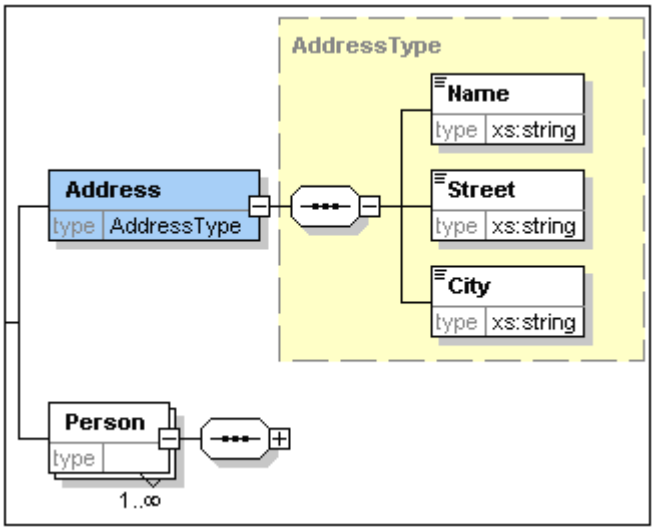
Schema design | Display diagram, the component icon displays the content model of the active global component in the schema overview.

Globals, extending simple and complex types

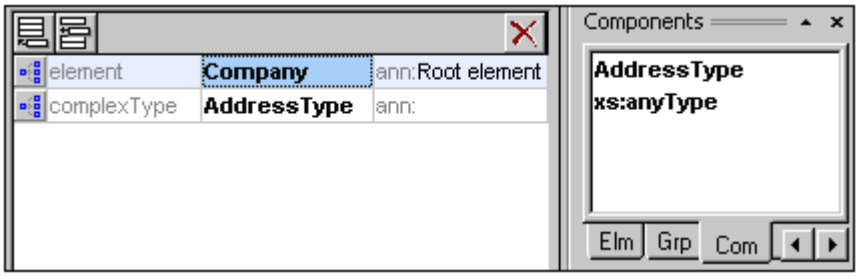
Having defined an element, you may then realize that you want to reuse it somewhere else in your schema. In XML Spy this is achieved by creating a **global component**.

To create a global component:

1. Right click the Address element, and select **Make Global | Complex type**.
   The Address elements appear in a yellow box.

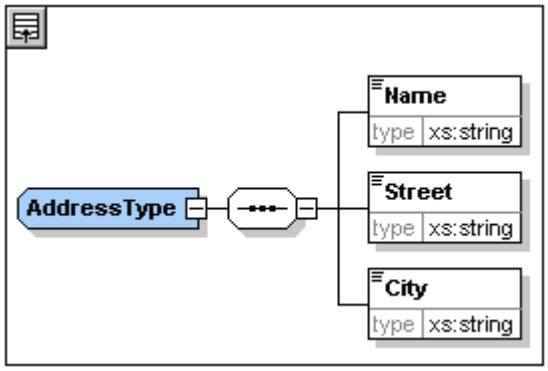


2. Click on the "Display all Globals" icon.
   The schema overview now displays two global components: the Company element and the complexType "AddressType".



   Click on the **Com**(plex) tab of the **Component Navigator** to see that AddressType is also visible there.

3. Click on the AddressType component icon , to see the content model.



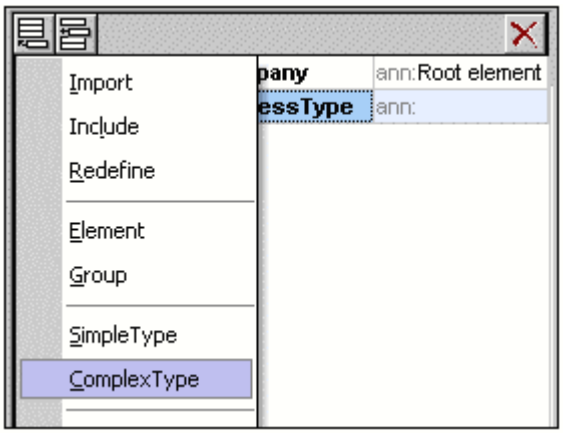
4. Click the "Display all Globals" icon to return to the schema overview.

Extending a "complex type" definition

We now want to use the global AddressType component, to create two kinds of country specific addresses. For this purpose we will define a **new** complex type **based** on the AddressType component.

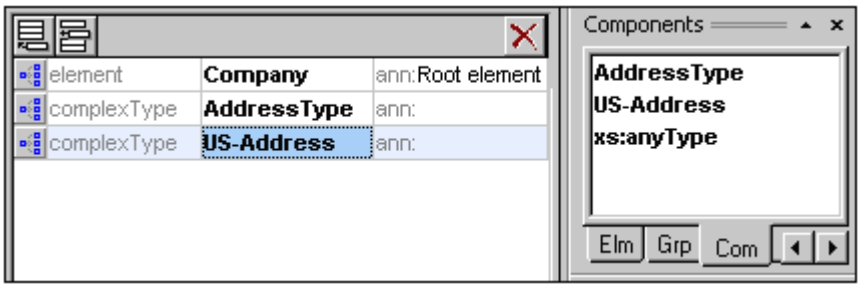
To extend a "complex type" definition:

1. Switch to the schema overview, if not already visible (Display all globals).
2. Click the **Append** icon, at the top left of the component window.
3. Select **ComplexType** from the context menu.



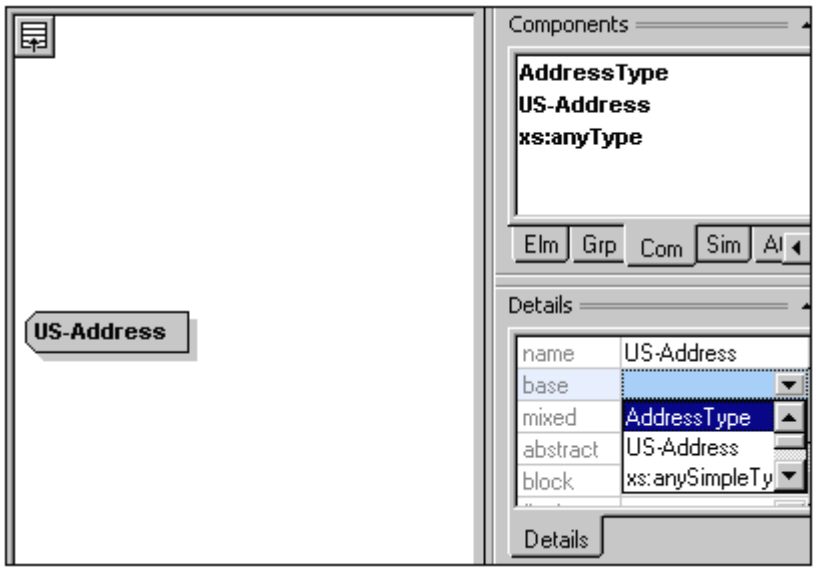
A new line appears in the component list, and the cursor is set for you to enter the component name.

4. Enter "US-Address" and confirm with Enter. (If you forget to enter the hyphen character "-", the element name will appear in **red,** signalling an illegal character.)

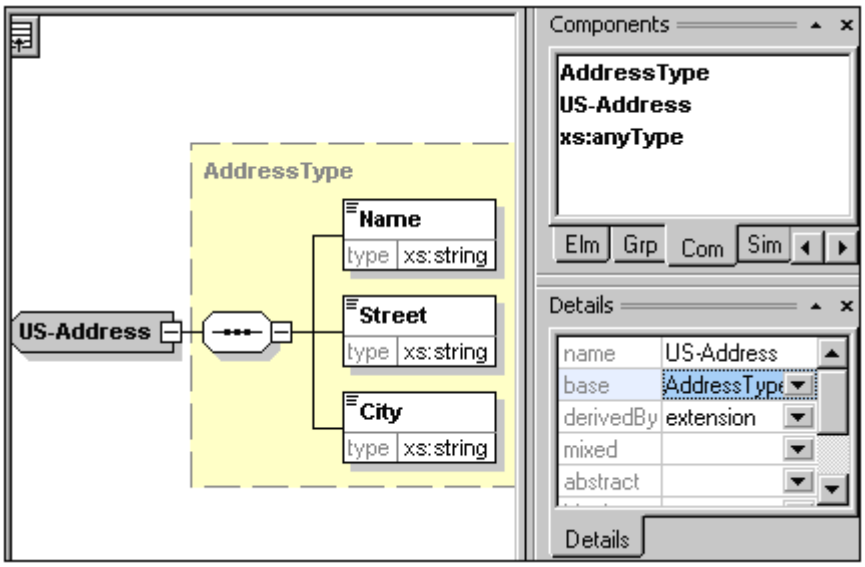


5. Click the **US-Address** component icon to see the content model.

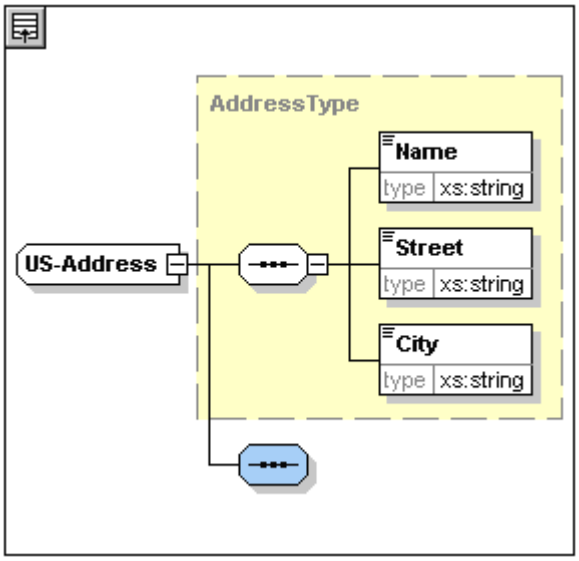
6. Click the "**base**" combo box in the Details entry helper, and select the "AddressType" entry.



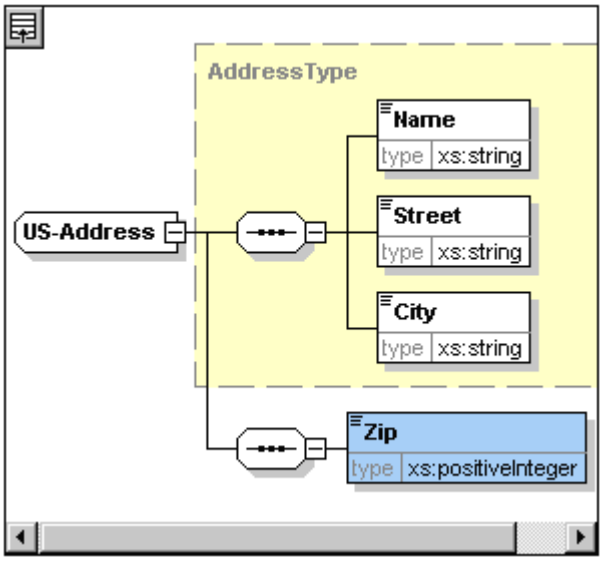
The content model view changes immediately and displays the previously defined generic address.



7. Right click the US-Address **element**, and select **Add Child | Sequence**.
   A new sequence compositor is displayed **outside** of the AddressType box. This is a visual indication that this is an **extension** to the element.



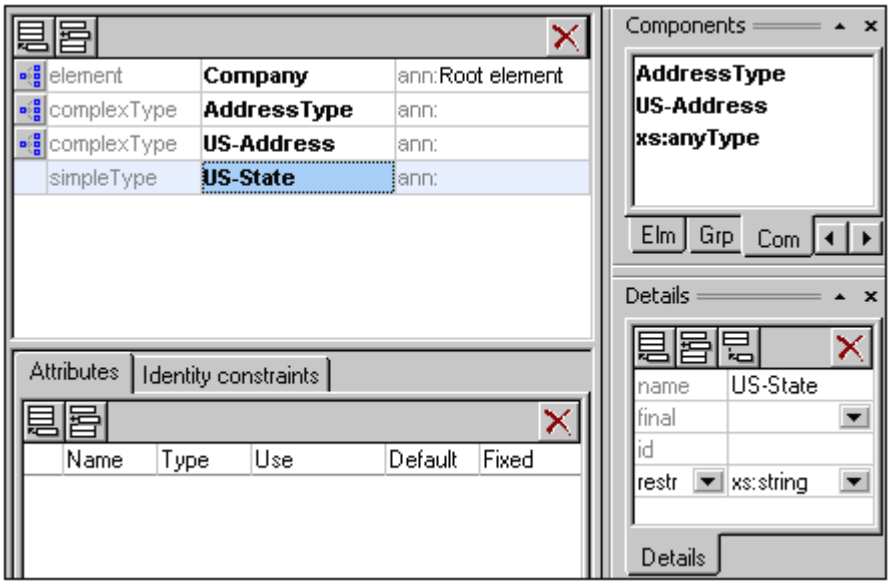
8. Right click the new **sequence** compositor, and select **Add Child | Element**.
9. Name the element "Zip", and hit the "**Tab**" button.
10. Select (or enter) **xs:positiveInteger** from the "type" field combo box, and confirm with Enter.



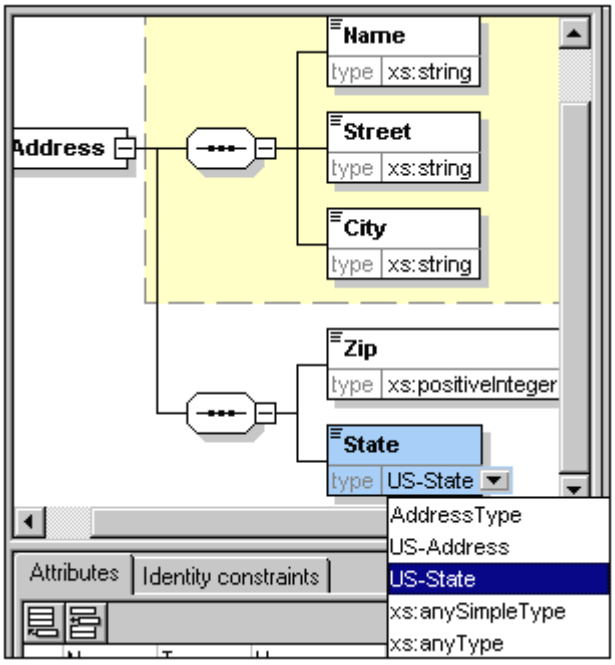
Creating reusable "simple type" elements
Simple type elements can also be made generic. In this case we want to make the State element reusable, so that an abbreviated version could also be included in address labels at a later time (GA for Georgia, for example).

To create reusable "simple type" elements:

1. Switch to the Schema overview (Display all Globals).
2. Click the **append** icon, select SimpleType, and enter "US-State" as the element name (Enter to confirm).
3. Select **xs:string** in the "**restr**." value field of the Details entry helper.
   This completes the definition. This element can now be used in the US-Address definition.



4. Click the US-Address component icon , then right click the lower sequence compositor and select **Add Child | Element**.
5. Enter "State" for the element name, and hit the "Tab" key.
6. Select (or enter) "US-State" from the "type" combo box (click Enter to confirm).

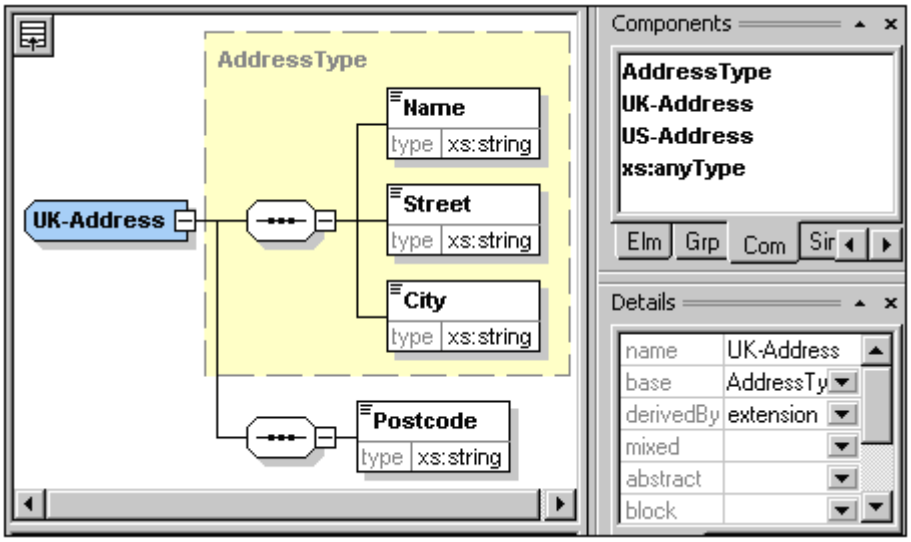


Please note:
Global simple types can only be created from the schema overview.

Creating the second Address template
Using the method described above, define the global complex type "**UK-Address**".

1. Create the global **complex type** "UK-Address", with the **base=**"AddressType"
2. Add a new Postcode element to the content model of UK-Address.

Your UK-Address content model should finally look like this:

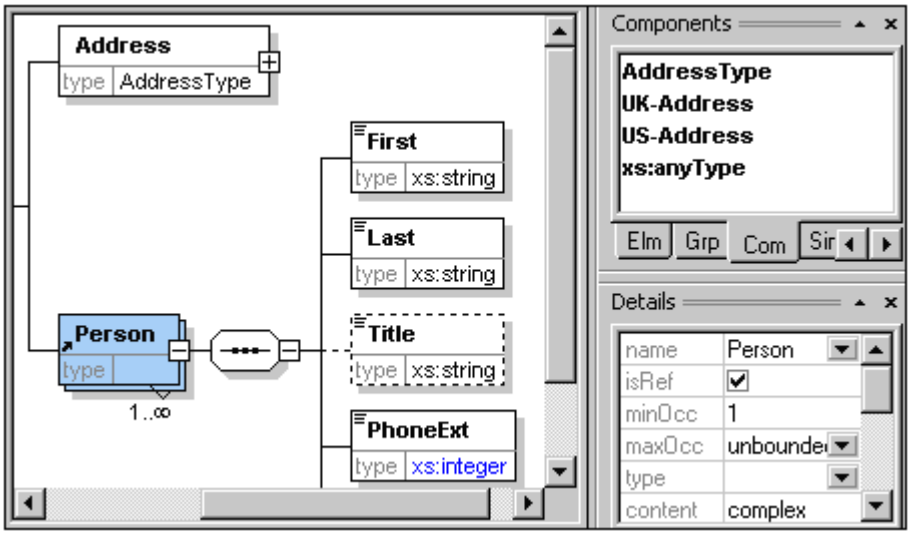


References, attributes and enumerations

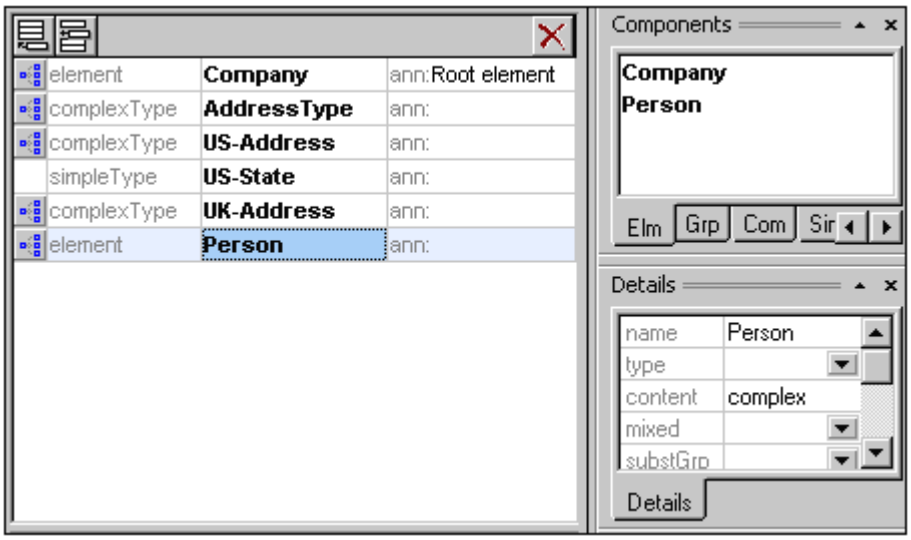
To finish off the schema definition we will make the Person element global, define specific element attributes and limit the attribute selection.

To create a reference:

1. Switch to the Schema overview (Display all Globals).
2. Click on the component icon of the **Company** element.
3. Right click the Person element, and select **Make Global | Element**.
   A small "link" icon appears in the Person element, showing that this element now references the globally declared "Person" element. The "**isRef**" field in the Details entry helper is set active.



4. Click the "Display all Globals" icon to return to the schema overview.
   The Person element is now also visible in the component list, as well as in the "**Elm**" tab of the Component navigator. Click the Elm tab to see the global elements.

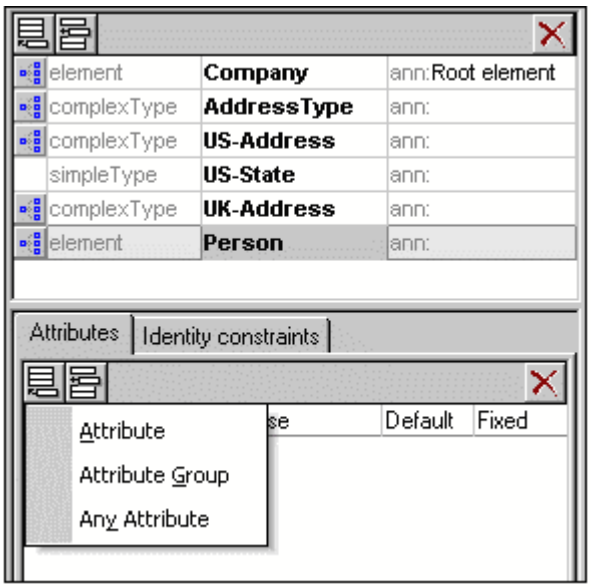


Please note:

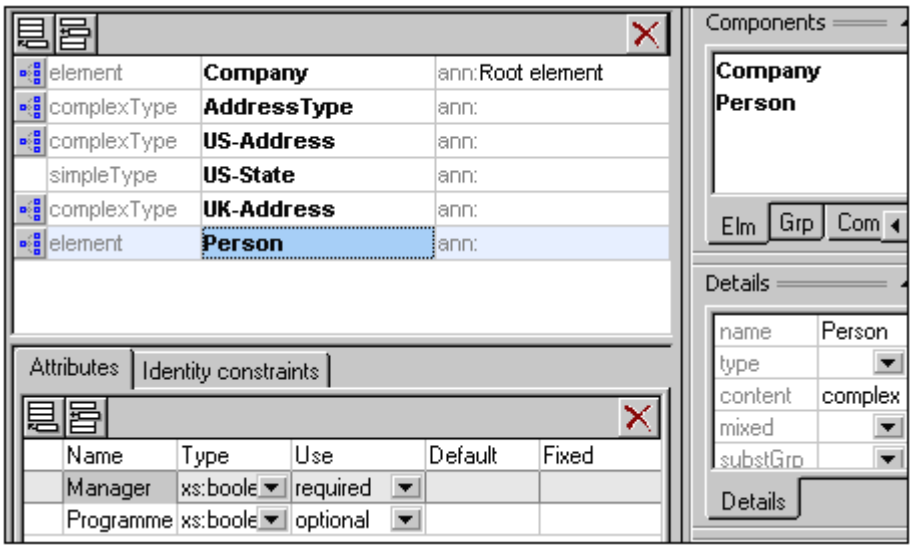
Global declarations do not describe where an element is to be used in an XML document, they only describe what it contains. Global definitions have to be referenced from within a complex type, or another element, to determine their position in the XML document.

To define Element attributes:

1. Click the Person element to make it active.
2. Click the Append icon, in the top left of the **attribute** tab (the lower window of the schema overview), and select the "Attribute" entry.



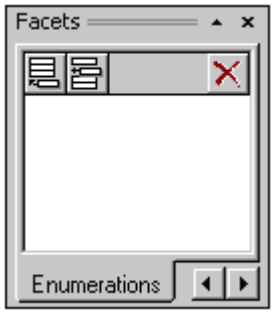
3. Enter "**Manager**" as the attribute name in Name field.
4. Use the **Type** combo box to select "xs:boolean".
5. Use the **Use** combo box to select "required".



6. Use the same method to:
   Add a "Programmer" attribute in the Name field (type="xs:boolean), and set its Use to "optional".

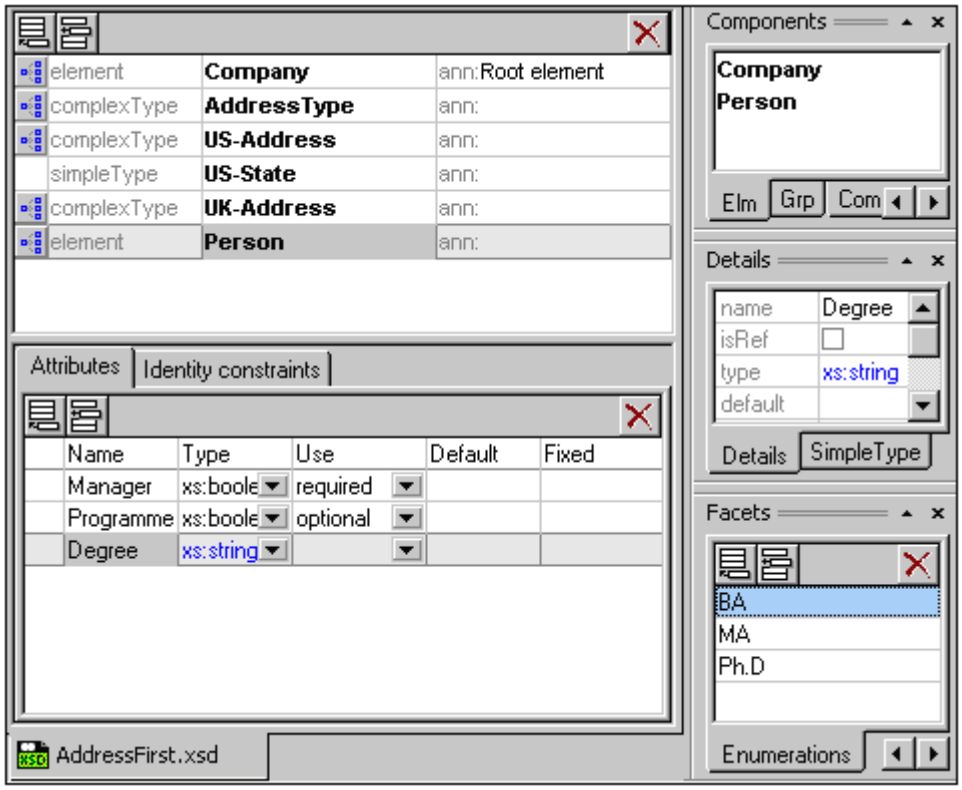
To limit the contents of an attribute (Enumerations):

1. Click the **Append** icon in the top left of the attribute window, and select the "**attribute**" entry.
2. Enter "**Degree**" as the attribute name, and select "xs:string" as the attribute type.
3. Click the **Enumerations** tab of the Facets entry helper.



4. Click the Append icon of the Enumerations tab and enter "BA", confirm with Enter.
5. Use the same method to add two more items to the enumerations list ("MA" and "Ph.D").

The finished schema should look like this:



6. Select the menu command **File | Save**, and save the file as **AddressLast.xsd**.

Please note:

This schema is available as '**AddressLast.xsd**' in the Tutorial folder.

Navigation shortcuts in schema documents

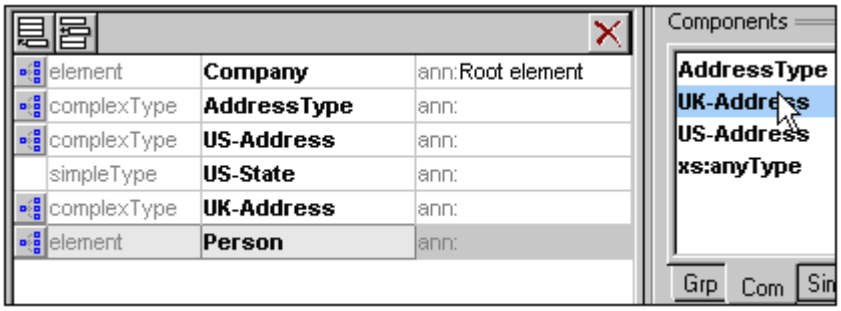
This section is designed to show you how you can navigate the Schema view efficiently.

Displaying the content model of any element:

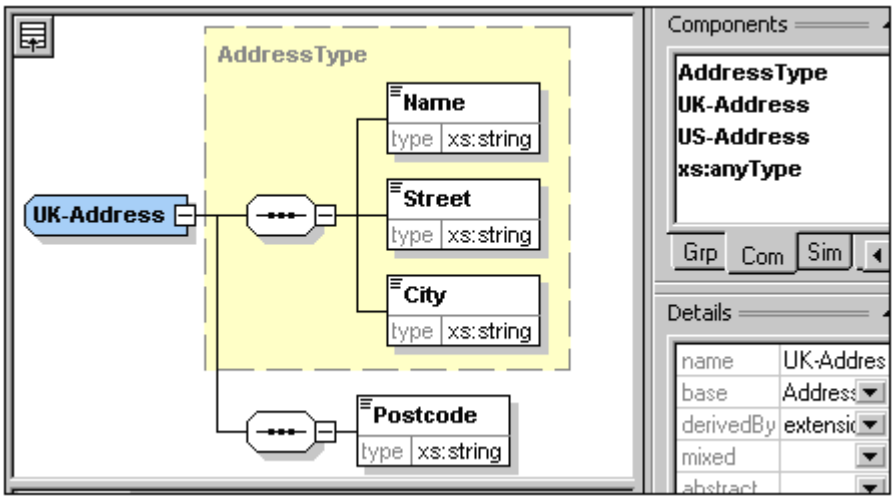
- Select the **element type** you want to see by clicking the specific Component navigator **tab** e.g. Com(plex).

  Elm=global elements, Grp=element group, Com=Complex type, Sim=Simple type, Att=Attribute, AGrp=Attribute group. The Component navigator entries are independent of the content model currently visible in the main window.

- **Double click** the element name in the Com tab e.g. **UK-Address**.



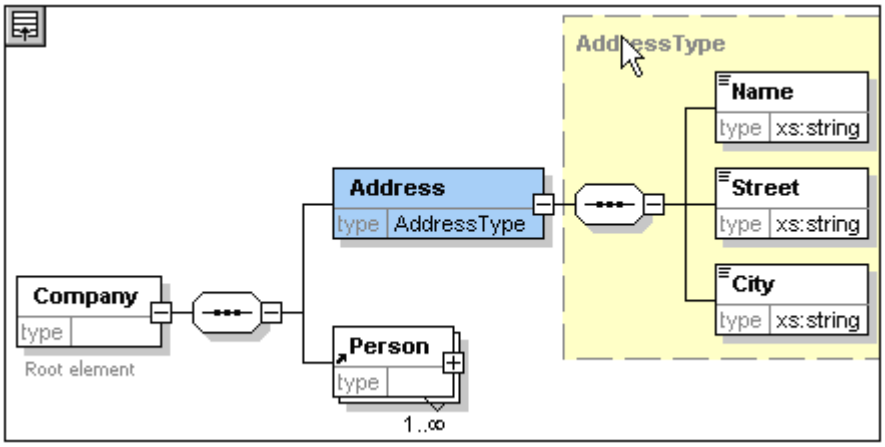
  The content model of the UK-Address element is displayed. The specific settings are shown in the Details tab.



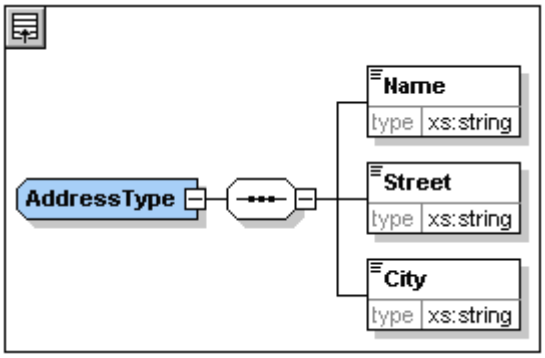
Go to "ElementType" definition:

E.g. While viewing the **Company** content model:

- Double clicking the *AddressType* **text** in the yellow box, takes you to the AddressType definition.



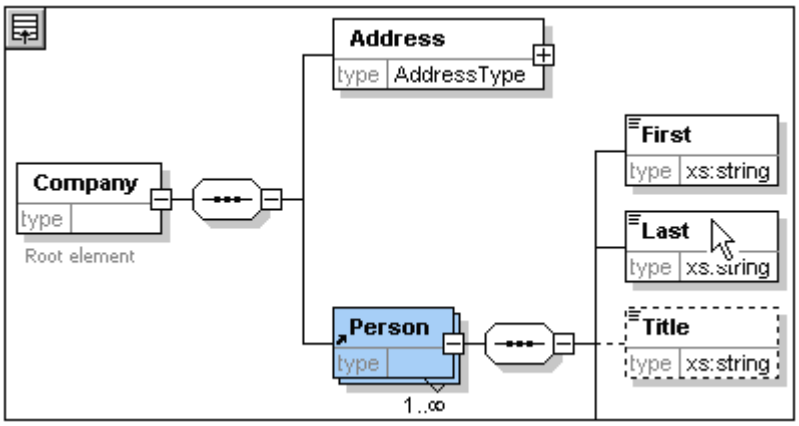
The AddressType definition:



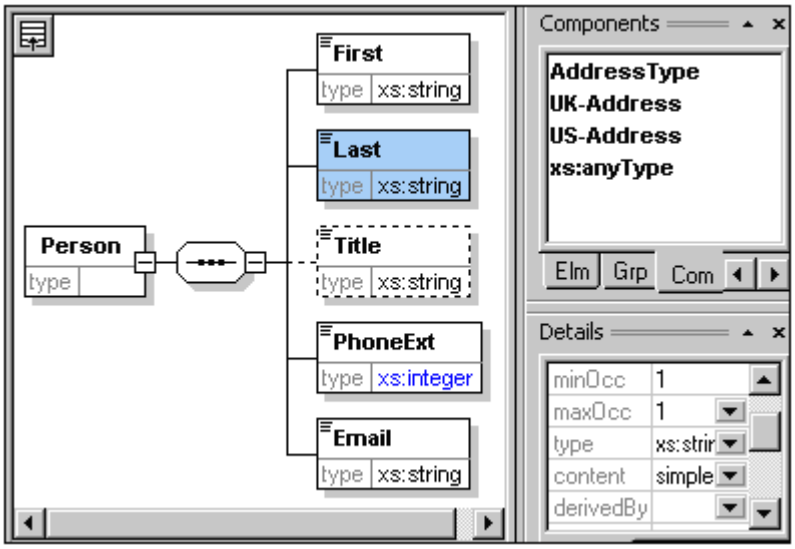
Go to element definition:

E.g. While viewing the **Company** content model:

- Press and hold down the **CTRL** keyboard key, and
- **Double click** on any element definition you want to see (here, the element **Last**).



The element Last, which is a sub-element of the Person element, is displayed. The specific settings are shown in the Details tab.


First
type xs:string
Last
type xs:string
Person
type
Title
type xs:string
PhoneExt
type xs:integer
Email
type xs:string
Components
AddressType
UK-Address
US-Address
xs:anyType
Elm
Grp
Com
Details
minOcc 1
maxOcc 1
type xs:strin
content simple
derivedBy

Generating Schema documentation

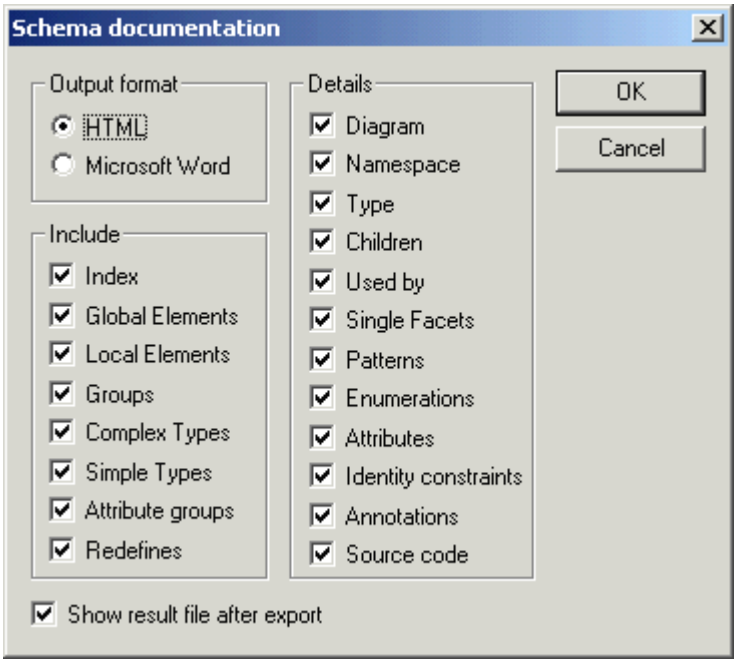
Goal of this section:
To generate detailed documentation on our current schema and select the specific elements we want to include in it.

You can generate an HTML or Word document, whereby related schema elements (child elements, complex types etc.) are hyperlinked, enabling you to navigate from element to element.

To generate Microsoft Word documentation, you have to have Microsoft Word installed on your computer (network).

To create schema documentation (of the AddressLast schema):

1. Select the menu option **Schema design | Generate documentation**.
2. Select the **Output format**, HTML or Word, and confirm with OK.
3. Select the folder and enter the name of HTML file you want to output, in the Save as... dialog, then click the Save button.



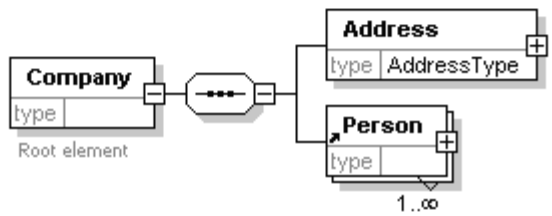
If you select HTML, the HTML document appears in the Browser view of XML Spy. Selecting Microsoft Word creates and displays the Word document.

Schema **AddressLast.xsd**

schema location: **C:\Program Files\Altova\XML Spy Suite\Examples\Tutorial\AddressLast.xsd**
targetNamespace: **http://my-company.com/namespace**

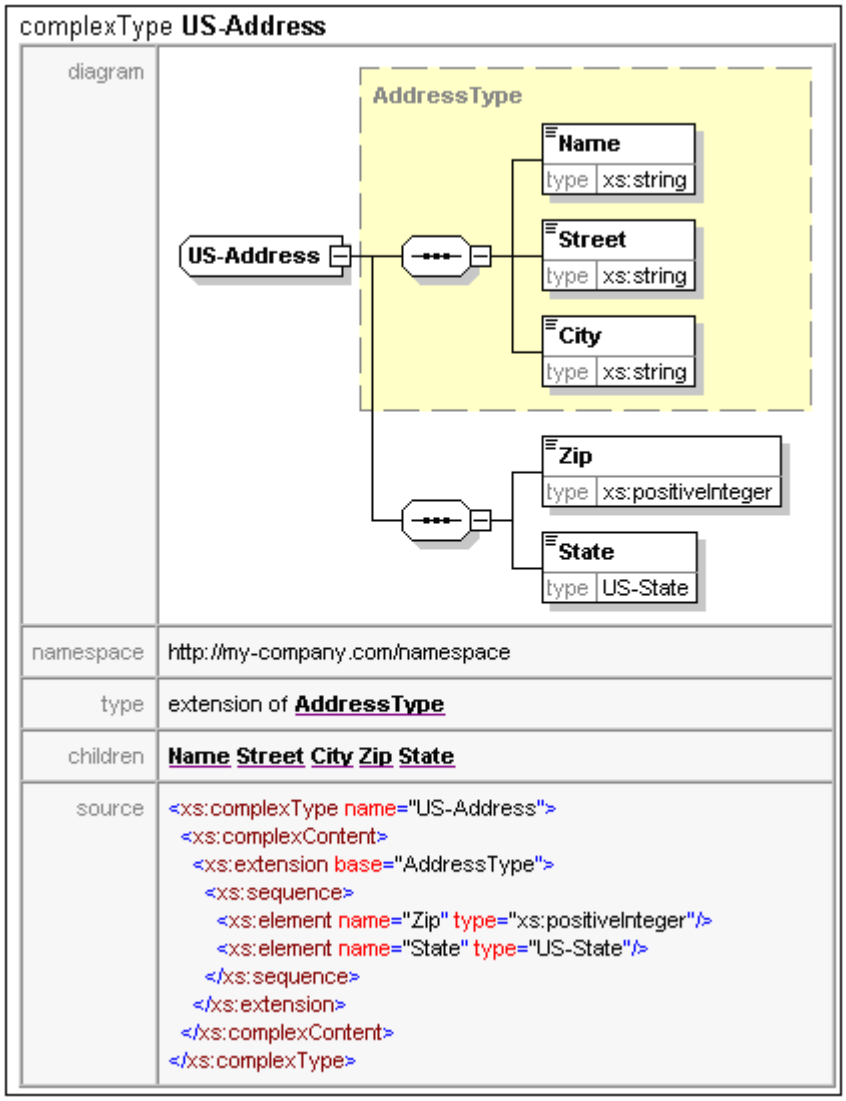
Elements	Complex types	Simple types
Company	**AddressType**	**US-State**
Person	**UK-Address**	
	US-Address	

element **Company**

diagram	
namespace	http://my-company.com/namespace
children	**Address Person**
annotation	documentation Root element

source	`<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>`

The diagram above, shows the **first page** of the schema documentation in HTML form. If components from other schemas have been included, then those schemas are also documented.



The diagram above, shows how ComplexTypes are documented.

element **US-Address/Zip**

diagram	Zip type xs:positiveInteger
namespace	http://my-company.com/namespace
type	**xs:positiveInteger**
source	<xs:element name="Zip" type="xs:positiveInteger"/>

element **US-Address/State**

diagram	State type US-State
namespace	http://my-company.com/namespace
type	**US-State**
source	<xs:element name="State" type="US-State"/>

simpleType **US-State**

namespace	http://my-company.com/namespace
type	**xs:string**
used by	element **US-Address/State**
source	<xs:simpleType name="US-State"> <xs:restriction base="xs:string"/> </xs:simpleType>

The diagram above, shows how elements and simpleTypes are documented.

Creating an XML document

Goal of this section:
To create a new XML document and use the various XML Spy views and intelligent editing capabilities, to rapidly enter and validate data.

This will be achieved by:

- Creating a new XML document based on the **AddressLast** schema
- Making **elementType** definitions available to an XML document
- Adding elements using **intelligent entry helpers** in the Text and Enhanced Grid view
- Copying **XML data to Excel**, adding new data there, and pasting it back to XML Spy (Enhanced Grid-, and Database/Table view)
- **Sorting** table data by Last name in the **Database/Table** view
- **Validating** the XML document
- **Updating** the **schema definition** to allow for three digit phone extensions

Functions (and their icons) in this section:

File | New, creates a new type of XML file.

View | Text View, switches to the text view.

View | Enhanced Grid view, switches to the Enhanced Grid view.

XML | Table | Display as Table, displays recurring elements and attributes in table form. Hotkey: **F9** switches between the Table and Enhanced Grid view.

expand, displays sub elements, in the Enhanced Grid view.

Checks for well-formedness. Hotkey: **F7**.

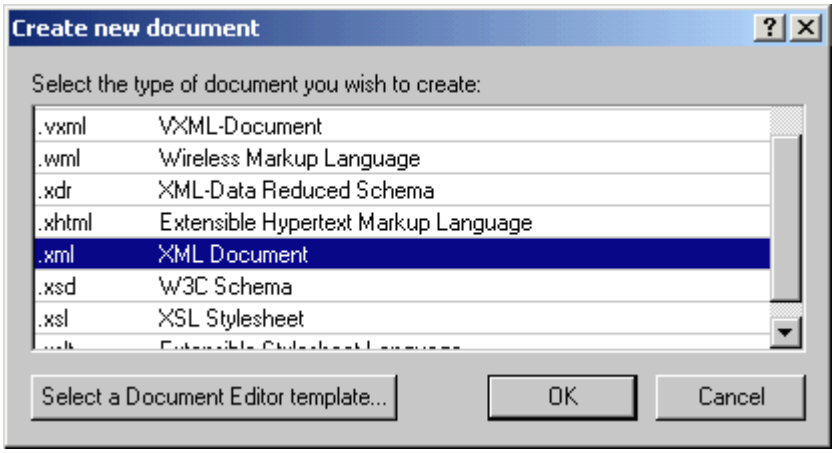
Validate the XML file against the associated DTD or Schema. Hotkey **F8**.

Opens the associated DTD or Schema file.

Creating and completing a new XML file

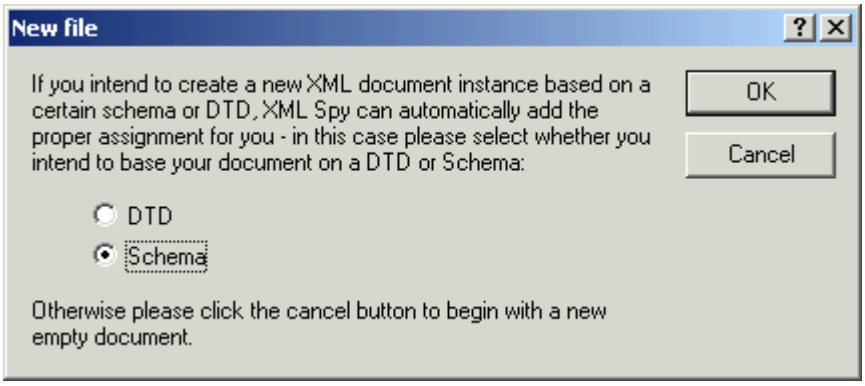
To create a new XML document:

1. Select the menu option **File | New**, and select the **.xml XML Document** entry from the dialog, then confirm with OK.



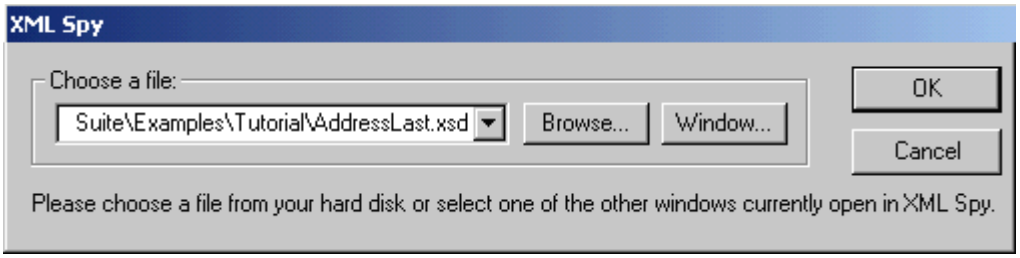
   A prompt appears, asking if you want to base the XML document on a DTD or Schema.

2. Click the **Schema** radio button, and confirm with OK.



   A further dialog appears, asking you to select the schema file your XML document is to be based on.

3. Use the Browse or Window buttons to find the schema file, in our case the **AddressLast** schema, and confirm the selection with OK.

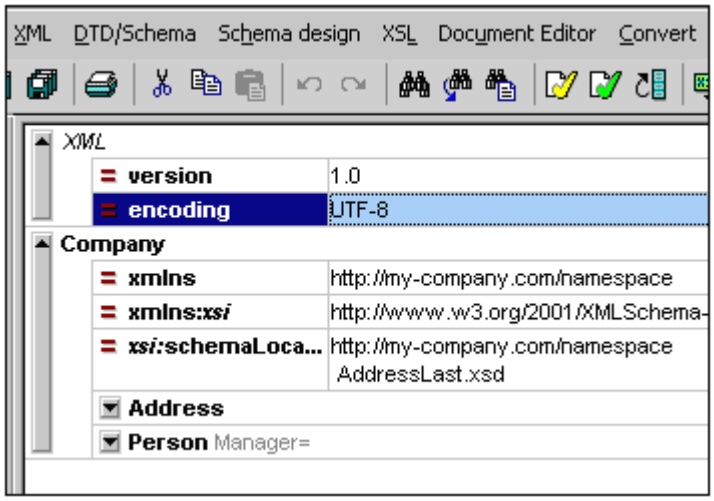


   An XML document containing the main elements defined by the schema, opens in the main window. XML documents are automatically opened in the Enhanced Grid View.

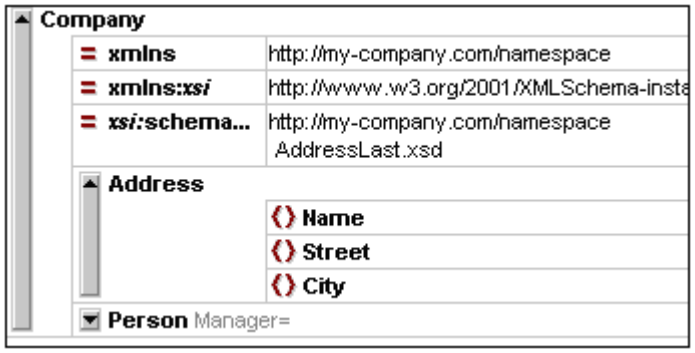
Please note:

XML Spy tries to find the root element of a schema automatically. The "Select a root

element" dialog box is opened, if it is unclear which is the root element. You can then select the root element manually.



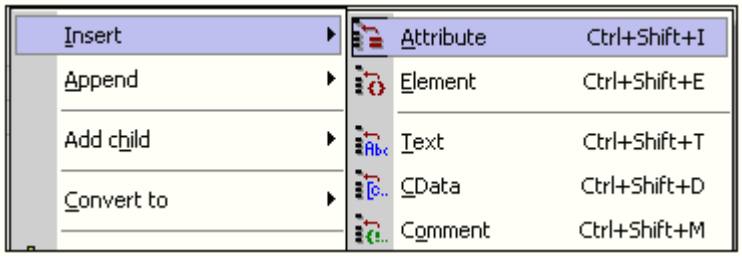
4. Click on any element to deselect the data.
5. Click on the icon next to Address, to view the Address sub-elements.



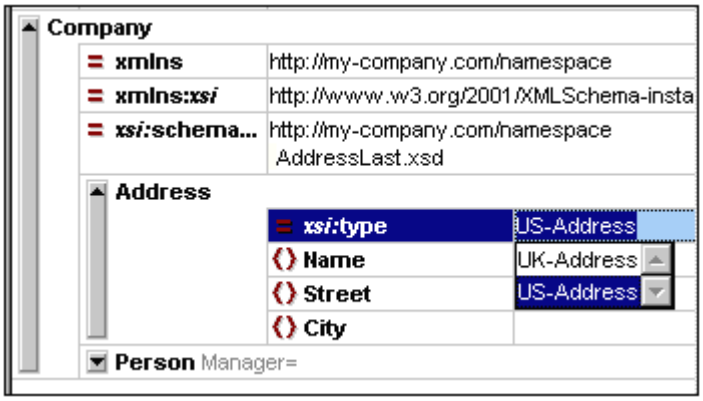
Making elementType definitions available in XML documents

The Address elements we see in the grid view are those that were defined by the global complex type "AddressType". We would, however, like to access the specific AddressTypes we defined: UK- and US-Address.

1. Right click the **Name** element, and select **Insert | Attribute** from the context menu. An attribute field is added to the Address element, and a popup containing **xsi:type** automatically opens.



2. Hit the "Tab" key to move into the next field.
3. Select **US-Address** from the drop-down list, and confirm with Enter.

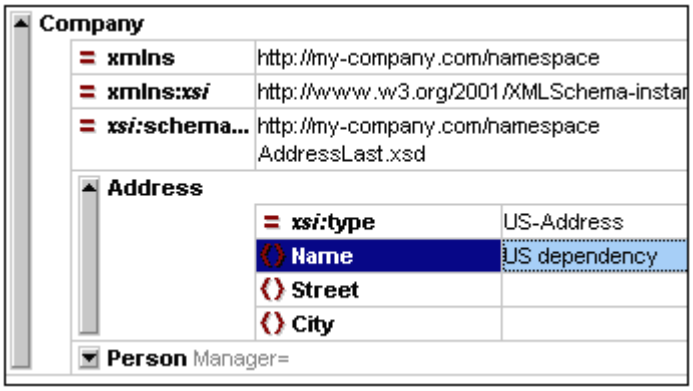


Please note:

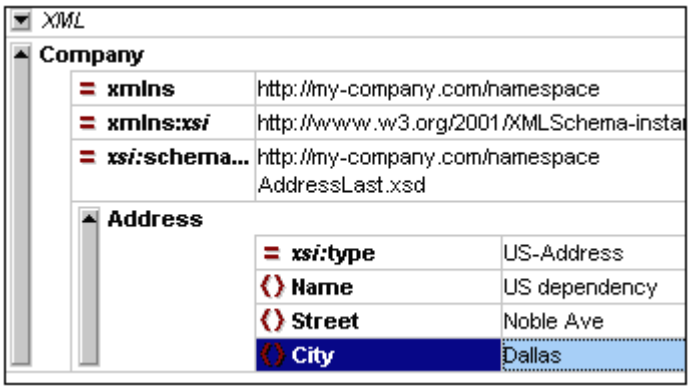
The ***xsi*** prefix allows you to use special XML Schema related commands in your XML document instance. Please see the W3C website at http://www.w3.org/TR/2001/REC-xmlschema-0-20010502 for more information.

Entering (and deleting) data

1. Double click in the **Name** value field (or use the arrow keys) and enter "US dependency", confirm with Enter.



2. Use the same method to enter a **Street** and **City** name (e.g. Noble Ave. and Dallas).
3. Click the **Person** element, and hit the "**Del**" key to delete it (we will add it again in a few moments in the Text view!).
4. Click on any Address element to deselect the elements.
   Your XML document should look like this:



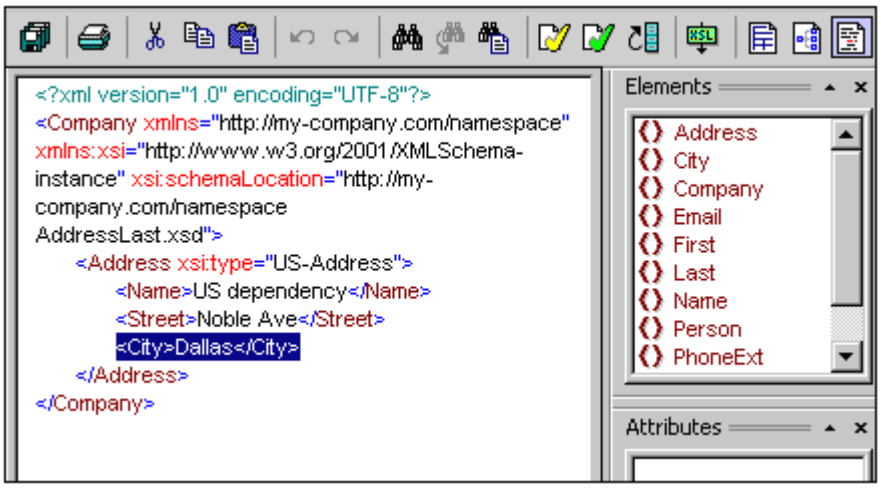
Editing in Text- and Enhanced Grid view

XML Spy Text view

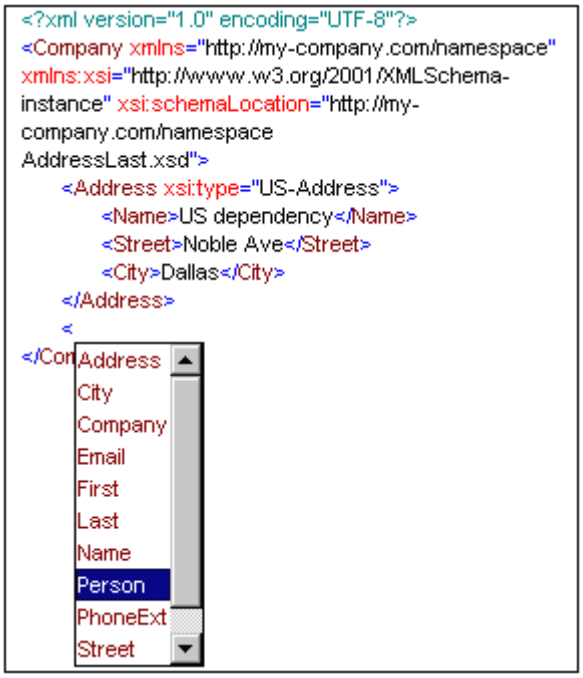
When it comes down to low-level work, the text view of XML Spy is suitable for editing any type of XML files in textual or source code form, and provides **intelligent editing** capabilities if you are working with an XML document based on a DTD or XML Schema.

Viewing and entering data in the Text view

1. Select the menu item **View | Text view**, or click on the Text view icon.
   You now see the XML document in its raw text form (with syntax coloring).



2. Place the text cursor after the </Address> | **end tag**, and hit Enter to add a new line.
3. Enter the "less than" angle bracket **<** at this position.



4. A drop-down list appears; select the **Person** entry.

The element name "Person" as well as the attribute "Manager", are inserted.

```
    </Address>
    <Person Manager="|"
</Company>
```

5. Enter the letter "**t**" and hit Enter.

```
    </Address>
    <Person Manager="t"
</Company>
```

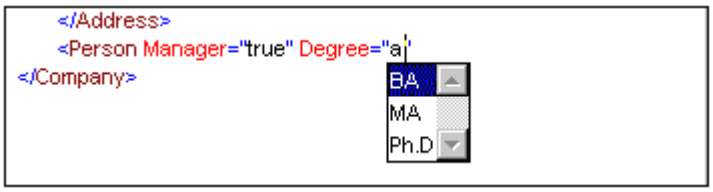
This opens a drop down list where "true" is highlighted. Enter, inserts the value (true) at the cursor positon.

6. Move the cursor to the end of the line (End key), and hit the space bar.
This opens the drop-down list again. There are now fewer entries available in the list; "Manager" is grayed out in the Attribute entry helper.

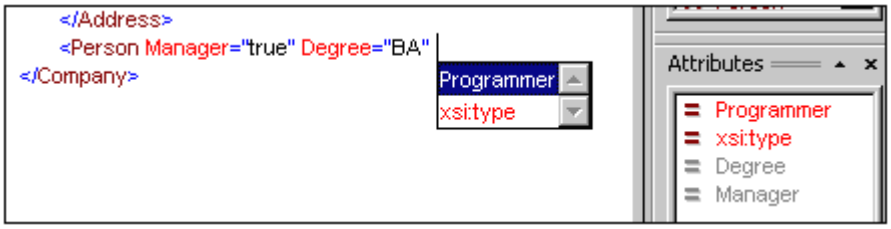
7. Select "Degree" with the Down arrow key, and hit Enter.

```
    </Address>
    <Person Manager="true" Degree="|"
</Company>
```

8. Enter any character. This opens another list box from which you can select one of the predefined enumerations (BA, MA or Ph.D).



9. Select "BA" with the Down arrow key (confirm with OK), move the cursor to the end of the line (End key), and hit the space bar. Manager and Degree are now grayed out in the Attribute entry helper.



10. Select "Programmer" with the Down arrow key, and hit Enter.

```
    </Address>
    <Person Manager="true" Degree="BA" Programmer="f"
</Company>
                                                  false
                                                  true
```

11. Enter a "**f**" character and hit Enter.
12. Move the cursor to the end of the line (End key), and enter the "greater than" angle bracket **>**.

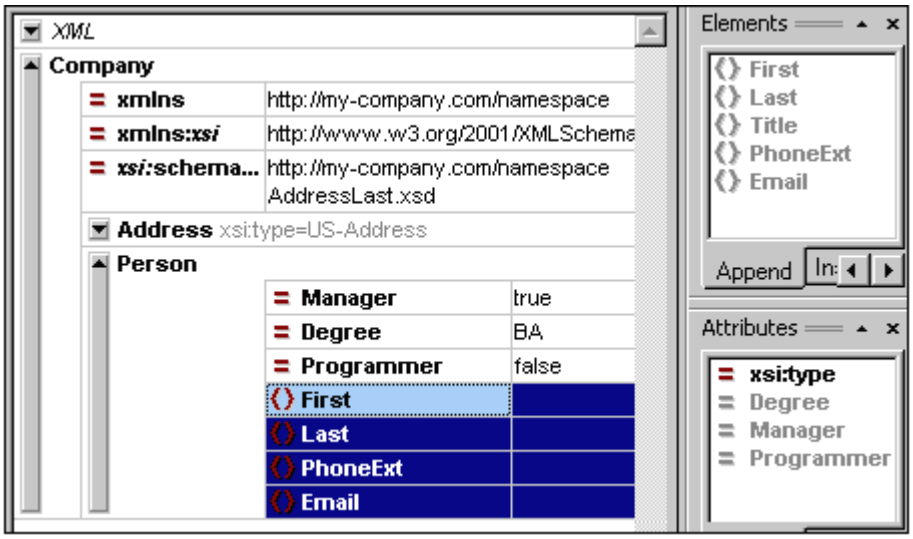
```
<?xml version="1.0" encoding="UTF-8"?>
<Company xmlns="http://my-company.com/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://my-company.com/namespace
AddressLast.xsd">
    <Address xsi:type="US-Address">
        <Name>US dependency</Name>
        <Street>Noble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

XML Spy automatically inserts all the Person element tags. Each element is supplied with start and end tags.

You could now enter the Person data here in the text view, but why do so? The Enhanced Grid view is a lot more comfortable, and contains a special view enabling recurring data to be presented in tabular form: the Database/Table view.

Enhanced Grid view

- Select the menu item **View | Enhanced Grid View,** or click the Enhanced Grid view icon. The Person attributes (and data) entered in the Text view, are also visible in the Enhanced Grid view.

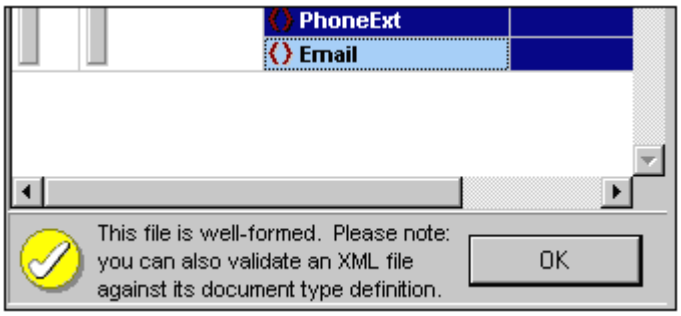

XML
Company
xmlns
http://my-company.com/namespace
xmlns:xsi
http://www.w3.org/2001/XMLSchema
xsi:schema...
http://my-company.com/namespace
AddressLast.xsd
Address xsi:type=US-Address
Person
Manager
true
Degree
BA
Programmer
false
First
Last
PhoneExt
Email
Elements
First
Last
Title
PhoneExt
Email
Append
Attributes
xsi:type
Degree
Manager
Programmer

Validating and entering data

At this point let's check if the document is well-formed and valid, there might still be work to do.

To check for well-formedness:

1. Select the menu option **XML | Check well-formedness** or hit the **F7** key.
   A message appears at the bottom of the main window declaring that the document is well formed. Click OK to confirm and close the message.

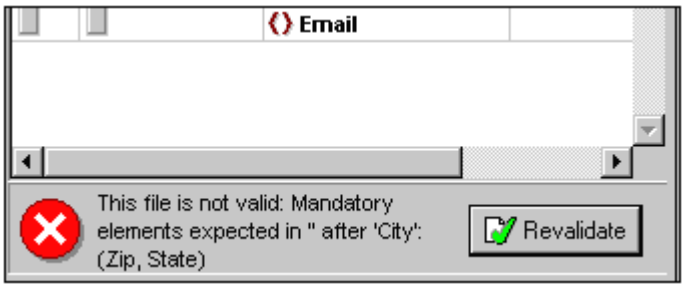


Being well-formed, means that the XML document **syntax** is correct (i.e. there is a root element, each start tag has a corresponding end tag, all elements are nested correctly etc.).

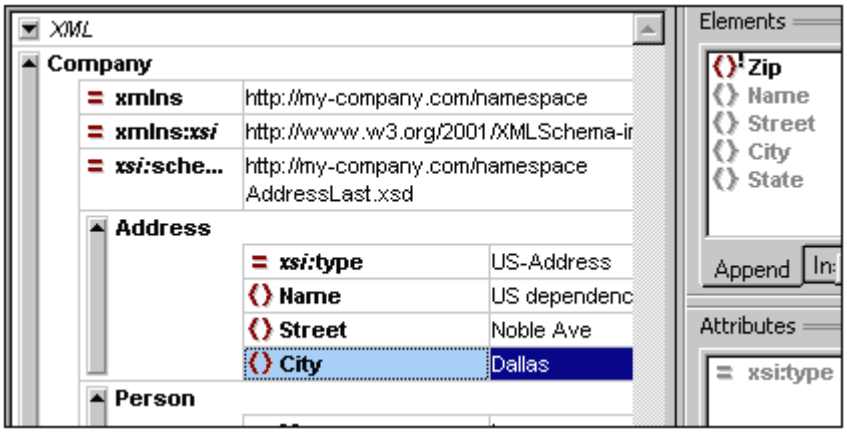
This check does not check against a schema file (or any other external file). Element sequence or element content are not checked either.

To check for validity:

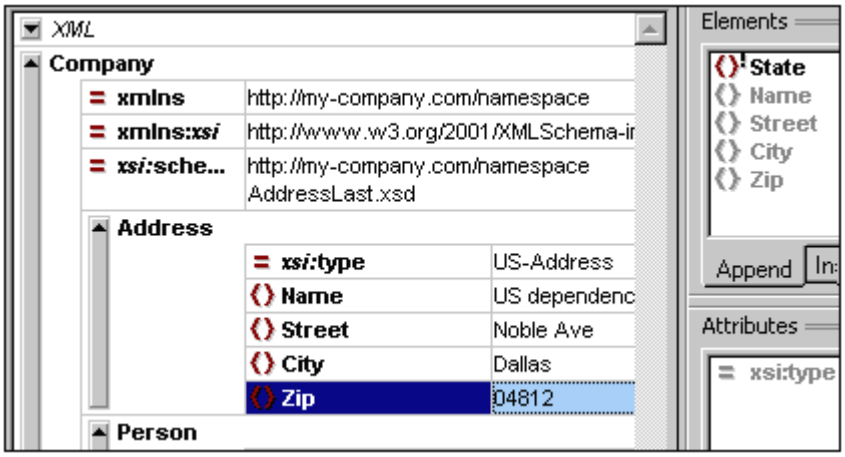
1. Select the menu option **XML | Validate** or hit the **F8** key.
   An **error message** appears: "This file is not valid: Mandatory elements expected after 'City' (Zip, State).
   The error message describes in detail what is currently wrong with our XML document.



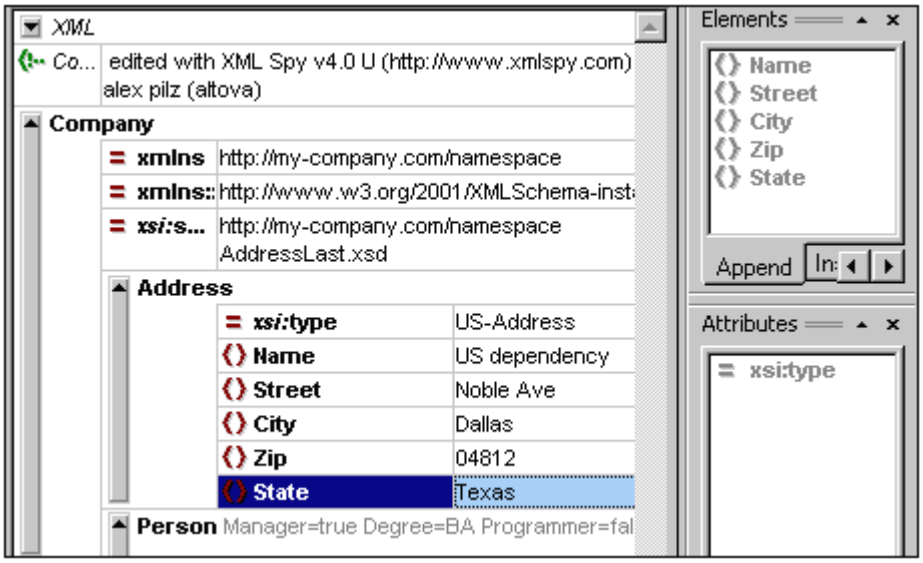
Fixing the invalid document (intelligent help):
At this point the element which is the "cause" of the error message is highlighted (City). Take note of the **Element entry helper** (top right). The Zip element is prefixed by an exclamation mark. This is the symbol for a **mandatory** element, and means that the US-Address element must contain the Zip sub-element.



1. Double click the **!Zip** element in the **Element entry helper**.
   This inserts the Zip element under the City element (Append tab is active by default).
2. Hit the **Tab** key, enter the Zip Code of the State (04812), and confirm with Enter.
   The Element entry helper now contains the **!State** entry, which is also a mandatory element and must also appear with the Zip element.

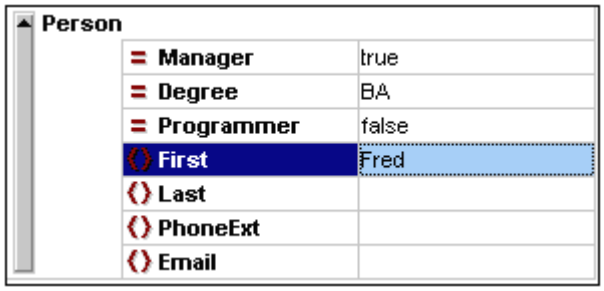


3. Double click the **!State** element, hit the Tab key and enter the name of the state (e.g. Texas), confirm with Enter.
   The Element entry helper now contains only grayed-out elements. This shows that all the required Address sub-elements have been inserted.

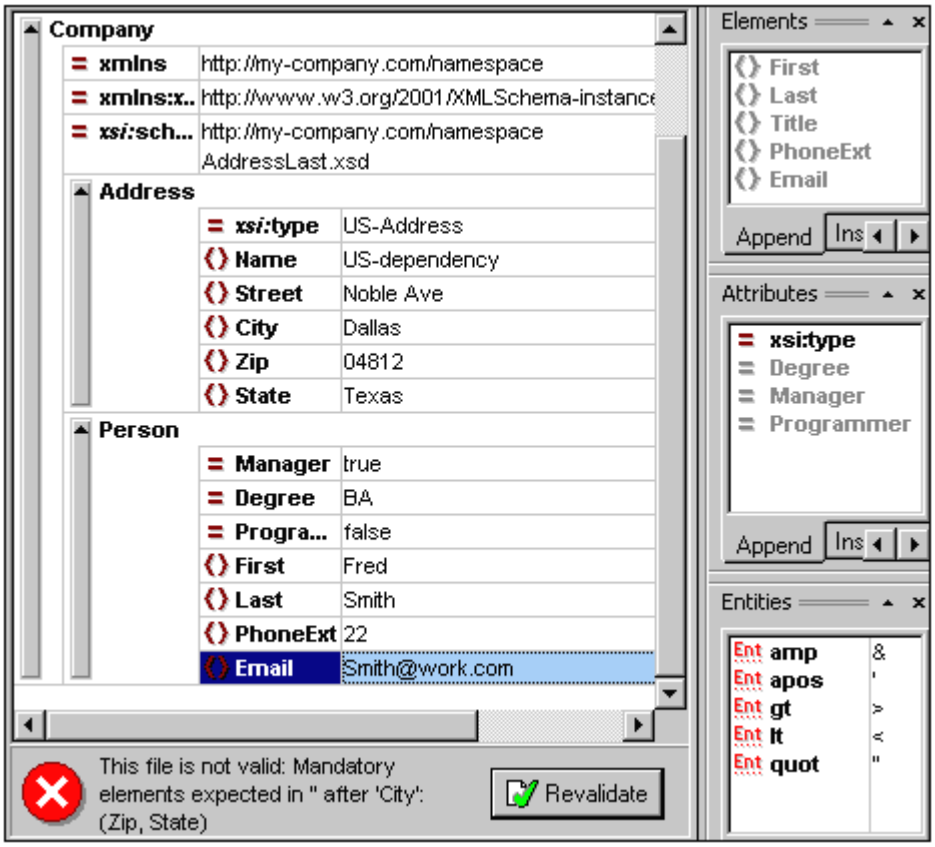


Filling in the rest of the XML document data & revalidating

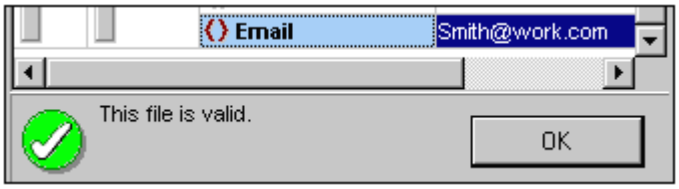
1. Click the empty element content field (right of element name) of the element **First**, enter the persons first name (e.g. Fred), and hit the Enter key.



2. Hit the Down arrow key, and fill in the next field, **Last** (e.g. Smith)
3. Use the same method to enter PhoneExt (e.g. 22) and the persons e-mail address (e.g. smith@work.com). You XML document should look like this:



4. Click the **Revalidate** button to check if the document is valid.
   The "This file is valid" message appears. The XML document is now valid against its schema. Click OK to confirm and close the message.



   Being valid, means that the XML document adheres to the assigned schema i.e. the elements and the sequence they appear in is correct, as well as the element "contents" and their attributes.

5. Select the menu option **File | Save As...** and name the XML document (e.g. **CompanyFirst.xml**)

This XML document is available as 'CompanyFirst.xml' in the Tutorial folder.

Please note:

An XML document does not have to be valid in order to save it. Saving an invalid document causes a prompt to appear which then allows you to select "**Save anyway**", the document is then saved in its current state.

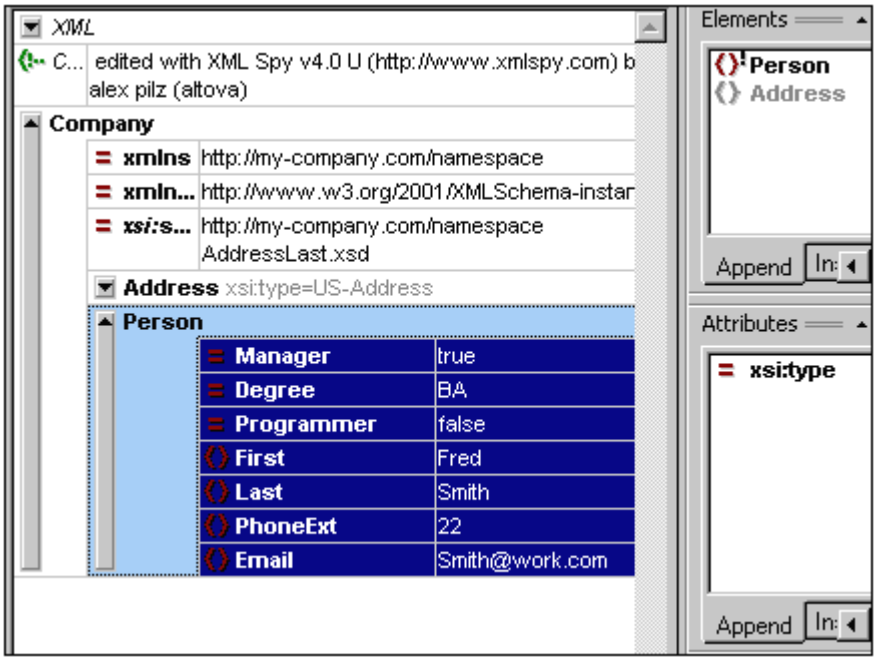
Manipulating data - Entry helpers

At this point we want to enter more person data in our XML document. XML Spy incorporates a special view (within the Enhanced Grid view), which allows you to enter data in tabular form - the **Database/Table view**.

Inserting elements and attributes (intelligent entry help):

We now want to add a new Person element to the document, as well as define specific person attributes.

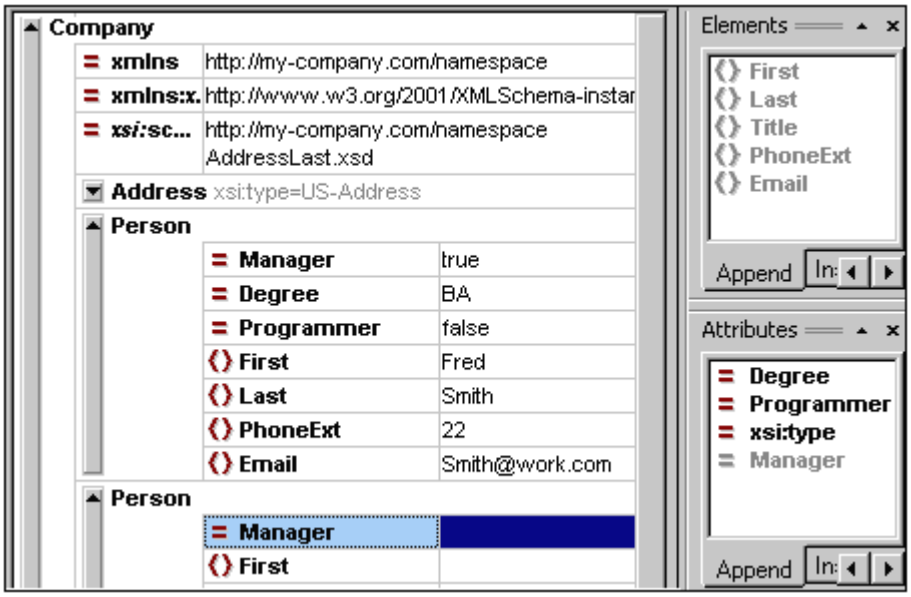
1. Click the gray **side bar** to the left of the Address element, to collapse the Address elements.
2. Click on or below, the "**Person**" element text in the grid view.
   This marks the Person and all its sub-elements. Notice that **!Person** is now available in the Element entry helper.



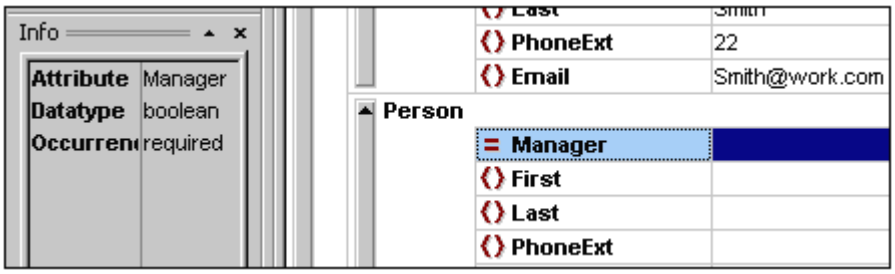
3. Double click the **!Person** element in the Element entry helper (Append tab active).
   A new Person element with all **required** sub-elements is appended.
4. Click on the **Manager** attribute of the new Person element. Notice the attributes available in the **Attribute entry helper**.

   The underlying schema document delivers information to the Entry helpers, specifically the elements or attributes that can be inserted at specific points in the XML document.

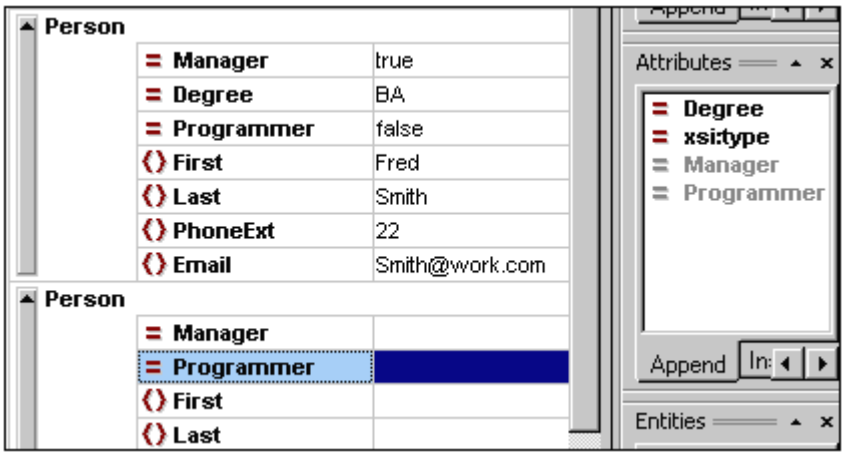
   Clicking the Update Entry Helpers icon , refreshes the Entry helper contents.



The "Manager" attribute has been **grayed out**, as it already exists in the Person element. Looking in the **Info** window, you can see all the Manager attribute information: Datatype=boolean, Occurrence=required (which means mandatory).



5. Double click the **Programmer** attribute in the Attribute entry helper.
   This inserts an empty **Programmer attribute** after the Manager attribute (append tab is active).



The Programmer attribute is now grayed out in the Attribute entry helper.
We could continue entering data here, but let's be more efficient and activate the

Database/Table view.

Database/Table view

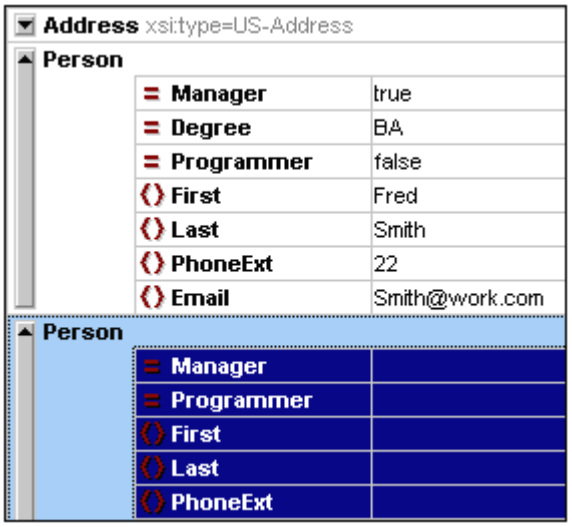
Database/Table view:
The Database /Table view is available wherever the Enhanced Grid view can be activated, and can be used when editing any type of XML file - XML, XSD, XSL etc.

Advantages:

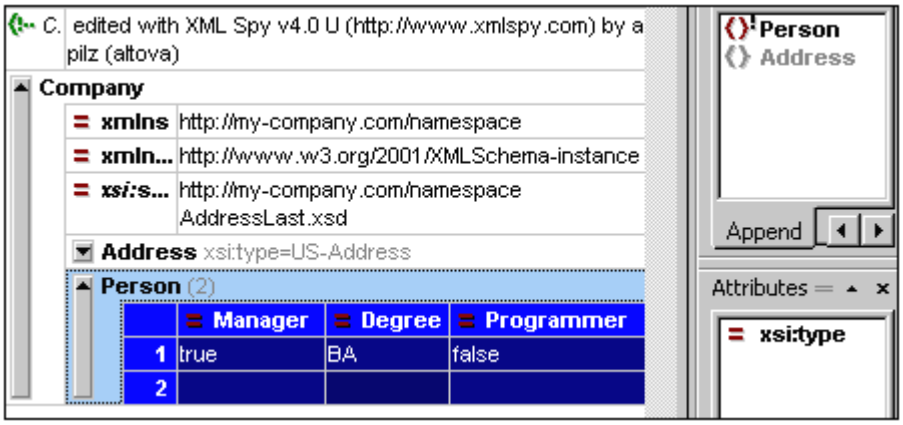
- Drag and drop column headers
- Sort column (table) data using the menu command **XML | Table | Ascending Sort**
- Append (or insert) rows using the menu command **XML | Table | Insert Row**.
- Copy and paste **structured data** to and from third party products
- Intelligent entry help

Activating the Database/Table view:

1. Click on or near, Person element text in the grid view (this marks the person element).



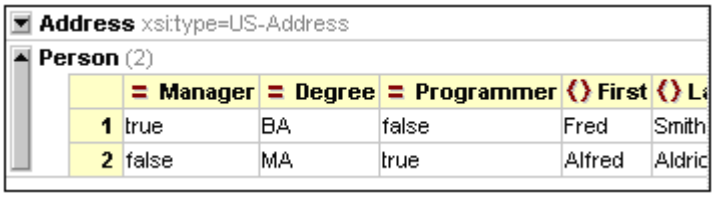
2. Select the menu option **XML | Table | Display as table**, or click the Display as table icon (Hotkey F9).
   The Person elements have now been combined into a single table. The Element and Attribute names are now the column headers, and the element contents (values) are now the rows of the table.
3. Select the menu option **View | Optimal widths**, or click the Optimal widths icon, to optimize the table view.



Please note:

The element and attribute entry helpers also function in the Database/Table view. The element entry helper now displays the mandatory **!Person** element; double clicking it would add a new row to the table.

4. Double click in the "Manager" cell of row 2, and select "false". Use the Tab key to get to the next cells, and select the following data: Degree=MA, Programmer=true, First=Alfred, Last=Aldrich, PhoneExt=33 and EMail=Aldrich@work.



Please note:

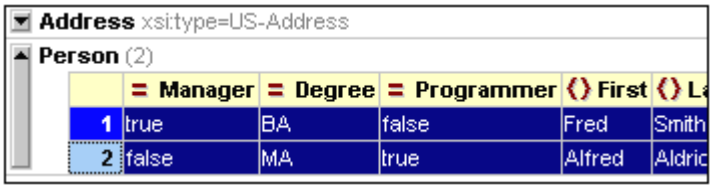
The **F9** key switches between Grid view and Database/Table view, of the currently selected table or recurring element.

Copying XML data to and from third party products

XML Spy allows you to easily copy data to and from third party products. The copied data can be used within XML Spy as well as third-party products, enabling you to transfer XML data to spreadsheet-like applications (e.g. Microsoft Excel).

Copying XML data to and from Excel:

1. Click on the row label 1, hold down the CTRL key and click on row label 2.
   This marks both rows of the table.



2. Select the menu option **Edit | Copy as Structured text**.
   The "Copy as Structured Text" command, copies elements to the clipboard as they appear on screen.
3. Switch to Excel and **paste** the XML data in an Excel worksheet.

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	

4. Enter a new row of data in Excel. Make sure that you enter a three digit number for the PhoneExt element (e.g. 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	
TRUE	Ph.D	FALSE	Colin	Coletti	444	Coletti@work.com	

5. Mark the table data in Excel, select **Edit | Copy**, and switch back to XML Spy.
6. Click in the top left cell of the table data in XML Spy, and select **Edit | Paste**.

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	Ph.D	FALSE	Colin	Coletti	444

The updated table data is now visible in the table.

7. Change the uppercase boolean values, "TRUE/FALSE", to lowercase "true/false" using the menu option **Edit | Replace** (Hotkey CTRL+H).

Sorting data in the Database/Table view

The Database/Table view allows you to sort your XML table data by any column you wish. In this case we want to sort our table by last names.

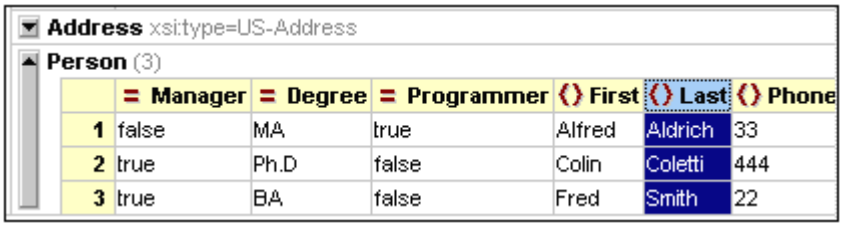
1. Click on the **Last** column header. This marks the whole column.

Address xsi:type=US-Address

Person (3)

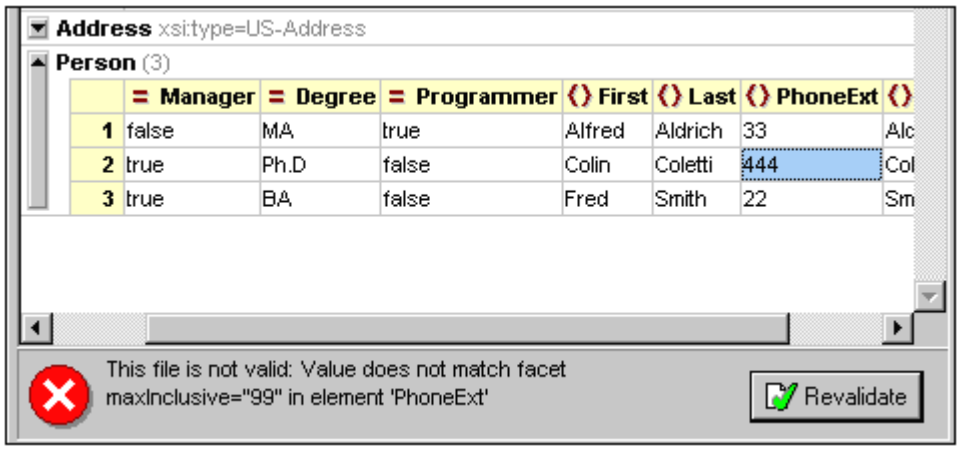
	= Manager	= Degree	= Programmer	{} First	{} Last	{} Phone
1	true	BA	false	Fred	Smith	22
2	false	MA	true	Alfred	Aldrich	33
3	true	Ph.D	false	Colin	Coletti	444

2. Select the menu option **XML | Table | Ascending sort,** or click on the "Ascending sort" icon.
   The column and **whole table** are now been sorted alphabetically (the column remains marked).



This sorting procedure affects your data at source level. (Click the Text view icon if you want to see the changes there.)

3. Select the menu option **XML | Validate**, or hit the **F8** key.
   An **error message** appears: "This file is not valid: Value does not match facet **maxInclusive="99"** in element **'PhoneExt'**.

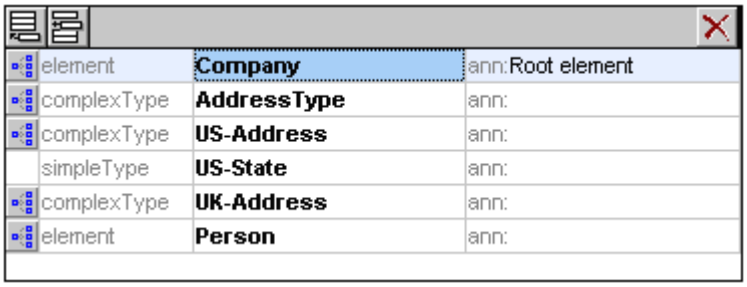


The offending element is automatically marked in the Database/table view. We need more telephone extensions!

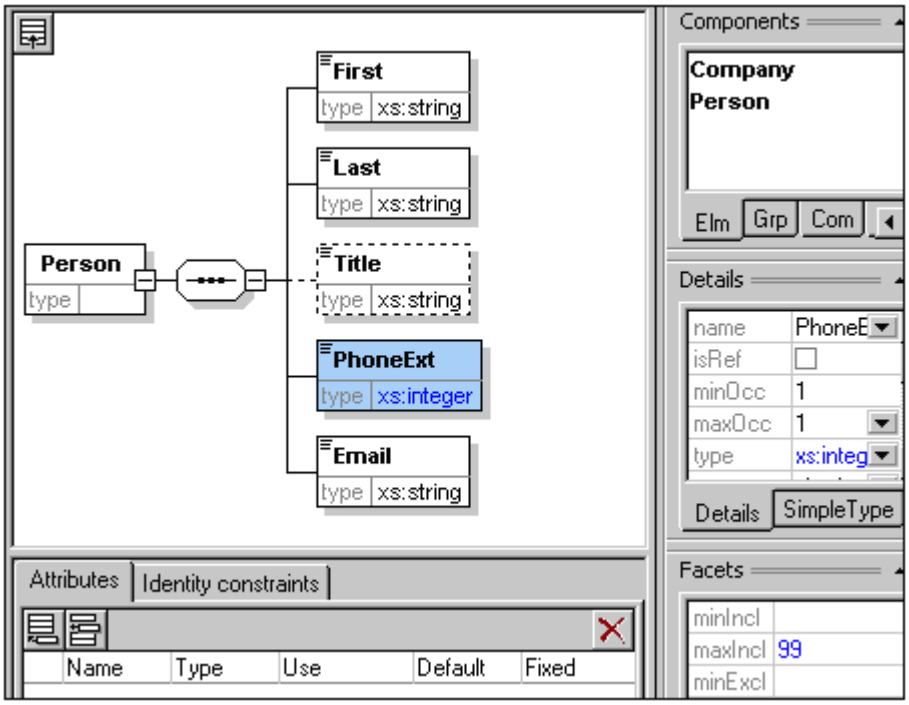
Updating a schema definition

At this point that we realize that two digit phone extensions are definitely not enough, and that we would like to allow for three digits. To do this we have to make a change to the underlying schema document.

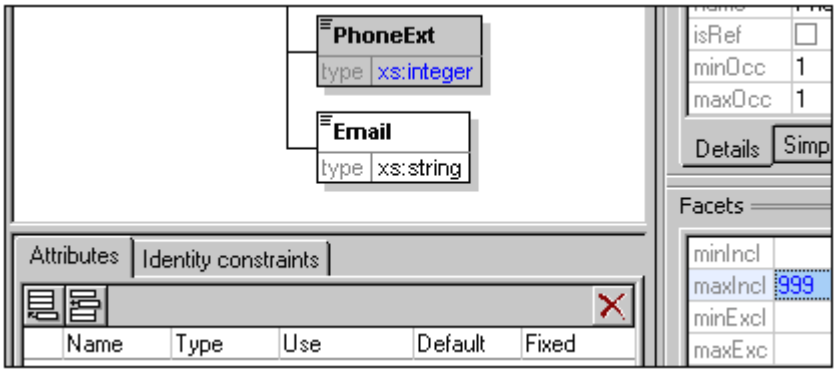
1. Select the menu option **DTD/Schema | Go to definition** or click the "Go to definition" icon.
   The associated schema document, in this case **AddressLast.xsd**, is opened in the Schema overview.



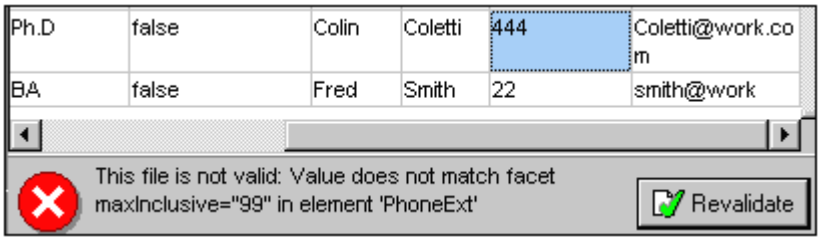
2. Click the "component" icon of the global **Person** element, and then click the PhoneExt element. You can now see the facet data in the facets tab.



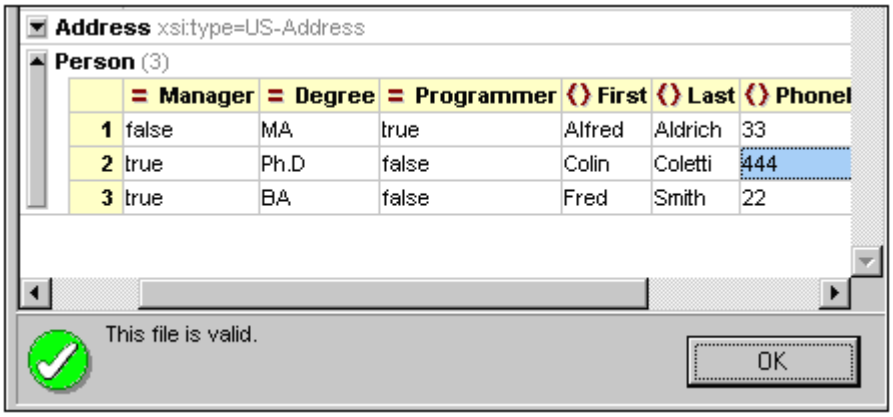
3. Click the "**maxIncl**" cell containing the facet data, enter **999**, and confirm with Enter.



4. Hit CTRL+TAB to switch back to the XML document.
5. Click the "**Revalidate**" button to revalidate the XML document.



The "This file is valid" message appears. The XML document now conforms to the changed schema definition!



6. Select the menu option **File | Save As...** and name the XML document (e.g. **CompanyLast.xml**).
7. Hit CTRL+TAB to switch back to the schema document, and save the schema document.

   The XML document is available as 'CompanyLast.xml' in the Tutorial folder.

XSL Transformation

Goal of this section:
To generate a Company HTML document which can then be posted on the corporate web site.

This will be achieved by:

- **Assigning** a predefined Company.**xsl** file to the XML document
- Using the XSL file to **transform** the XML file into an HTML document

Functions (and their icons) in this section:

XSL | Assign XSL, assigns an XSL file to an XML document.
XSL | Go to XSL, opens the XSL file referenced by the XML document.

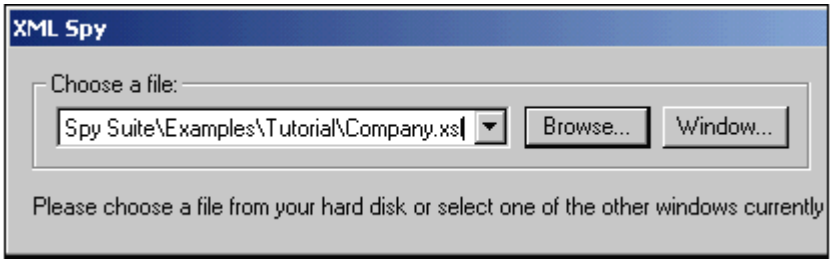
XSL | XSL Transformation, transforms the XML document(s) into the files specified by the XSL Transformation document, in this case into an HTML file. Hotkey: **F10**.

Please note:
If you encounter a problem while generating the HTML file, you only see the table header and no XML data, please download and install the **MSXML Parser 3.0** (649 kB) from the component download center at http://www.xmlspy.com/download_components.html.

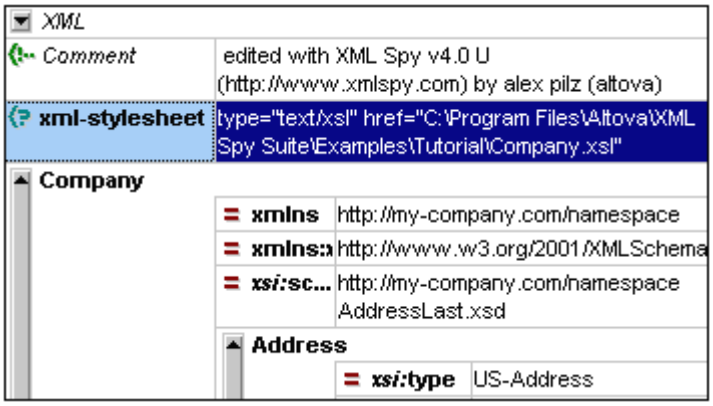
Transforming XML to HTML

To assign an XSL file to the CompanyLast XML file:

1. Click the **CompanyLast.xml** tab on the main window, to make it the active document.
2. Select the menu option **XSL | Assign XSL**.
3. Click the Browse button, select the **Company.xsl** file, and confirm with Open.



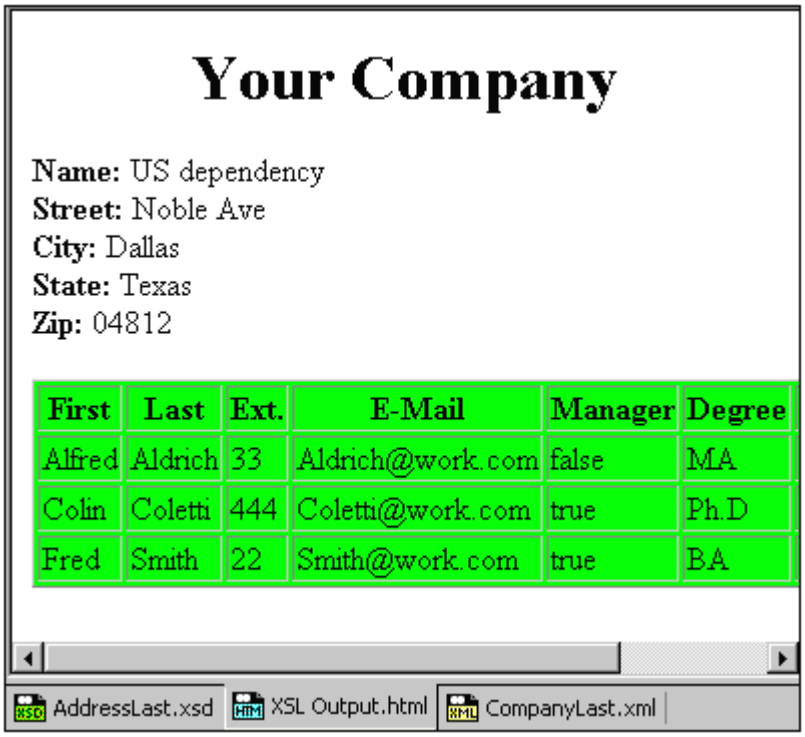
4. Click the OK button to assign the XSL file to the XML document.



A XML-stylesheet reference is placed in the XML document.

To transform the XML document into HTML:

1. Select the menu option **XSL | XSL Transformation** or click the icon (Hotkey: **F10**) This creates a new document in the **Browser view** with the name **XSL Output.html**. It shows the Company data in one block down the left, and the Person data in tabular form below.



Changing the output of the HTML file:
You can change the appearance/output of the HTML file by editing the underlying XSL file. In this case, we want to change the table background color from lime to yellow.

1. Click the **CompanyLast.xml** tab to make it the active document.
2. Select the menu option **XSL | Go to 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>
```

The command opens the Company.XSL file referenced in the XML document.

3. Find the line <table border="1" bgcolor="lime">, and change the entry bgcolor="**lime**" to 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. Select the menu option **File | Save** to retain the changes made to the XSL file.
5. Click the **CompanyLast.xml** tab to make the XML file active, and select **XSL | XSL Transformation,** or hit the **F10** key.

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

A **new** XSL Output.html file appears in the Browser view. The table background now appears in yellow.

6. Select the menu option **File | Save**, and save the document as **Company.html**.

XML Spy Suite and XSLT Designer
The XML Spy product family includes XSLT Designer which automates the generation of XSLT stylesheets for XML to HTML transformation. Please see the XSLT Designer documentation for more information on XSLT Designer. The documentation also includes a tutorial on how to generate a XSLT stylesheet.

Importing and exporting database data

Goal of this section:
To export Person data from our address list to MS Access, and reimport the Person table into XML Spy.

This will be achieved by:

- using the menu option **Convert**, and then selecting the export or import process.

Functions (and their icons) in this section:

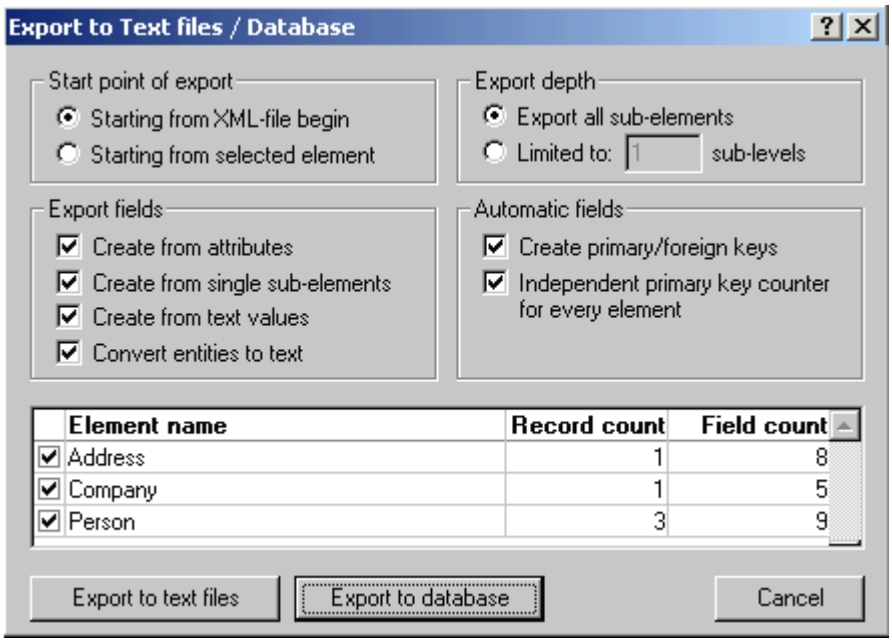
Convert | **Export to Text files / Database,** enables you to export XML data as text or for use in third party databases.

Convert | **Import Database data,** enables you to import database data into XML Spy.

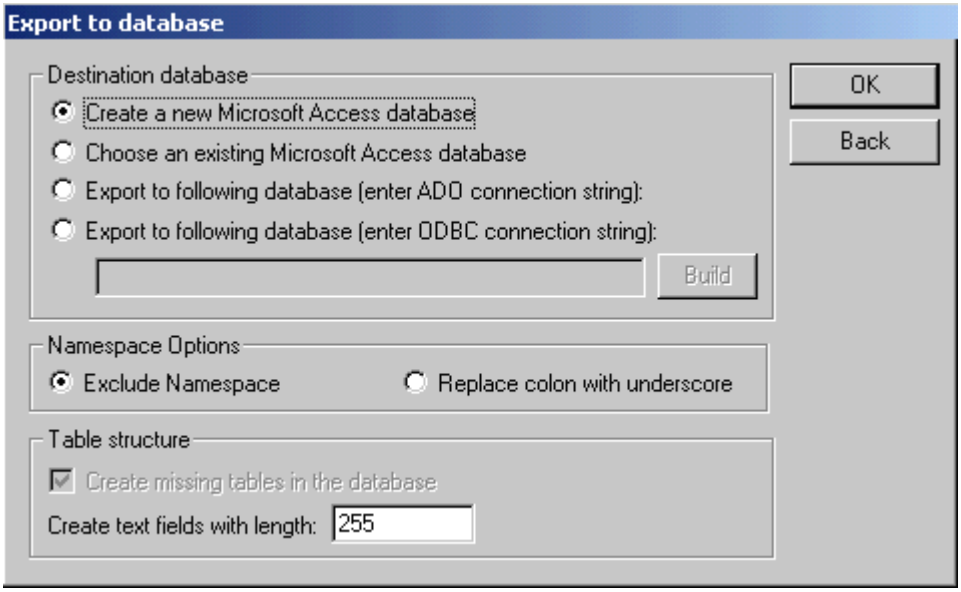
Exporting XML data to external databases

To export data to a database:

1. Click the **CompanyLast.xml** tab on the main window, to make it the active document.
2. Select the menu option **Convert | Export to Text files / Database...**.
   The default settings in this dialog export all elements, attributes, and generate primary and foreign keys.

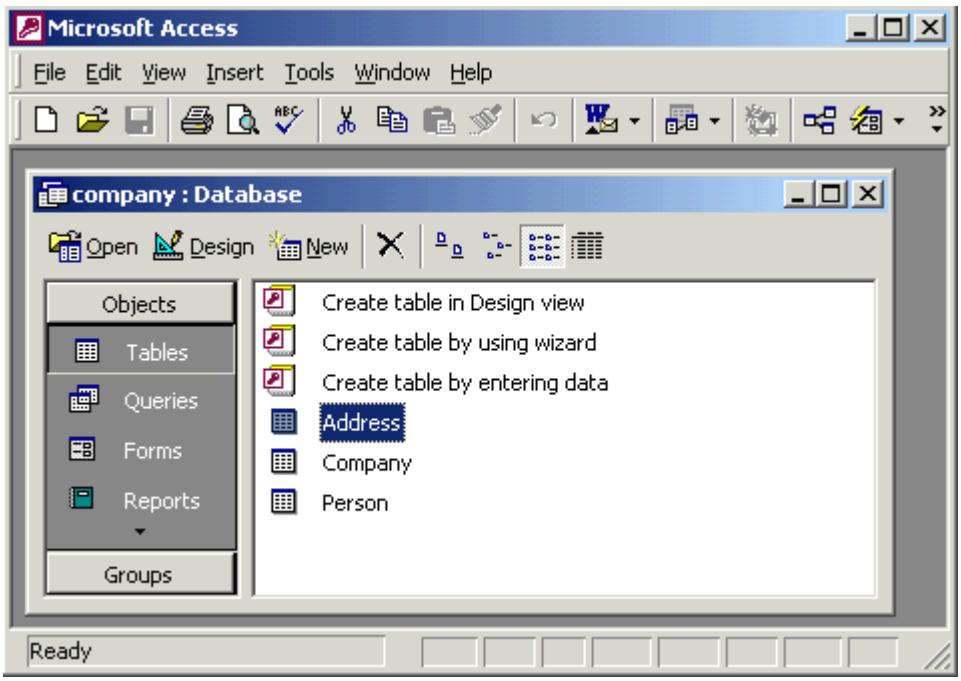


3. Click the **Export to Database** button.

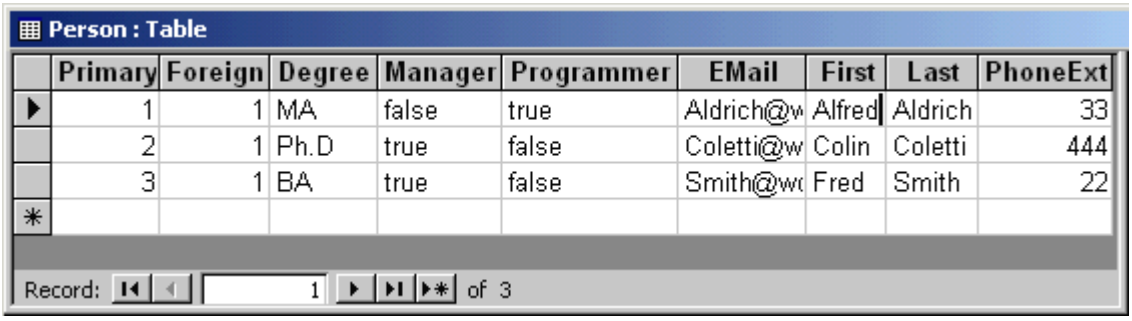


   This dialog allows you to select if you want to create a new Access table, export data to an existing one, or export to a third party database. The namespace option, "Exclude Namespace" is active per default.
4. Click the **Create a new Microsoft Access database** entry, and confirm with OK.

5. Enter the name of the new database in the "Save as..." dialog (e.g. Company.mdb), and confirm by clicking the Save button.
   A progress indicator displays export progress and a message box appears when the process has been completed successfully. Click OK to confirm.
6. Open the **Company.mdb** file you just saved. The export process automatically created a table for each exported element.



7. Double click the **Person** icon to open the Person table.
   The table shows all the Person data from the XML file and includes the "Automatic fields" PrimaryKey and ForeignKey, which can be used to index the database data.



Primary	Foreign	Degree	Manager	Programmer	EMail	First	Last	PhoneExt
1	1	MA	false	true	Aldrich@w	Alfred	Aldrich	33
2	1	Ph.D	true	false	Coletti@w	Colin	Coletti	444
3	1	BA	true	false	Smith@wc	Fred	Smith	22

Please note:

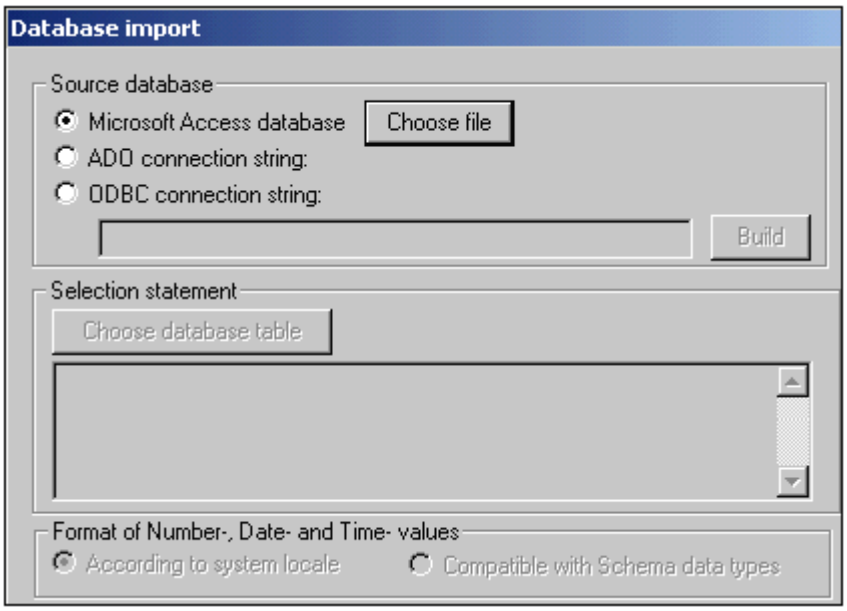
If you select the "Create a new Microsoft Access database" option when **exporting** database data, XML Spy creates a new Access 2000 database!

If you want to export data to an Access 97 database, please create an empty Access 97 database first, and then select the export option "Choose an existing Access database". There are no restrictions when **importing** data from any Access database.

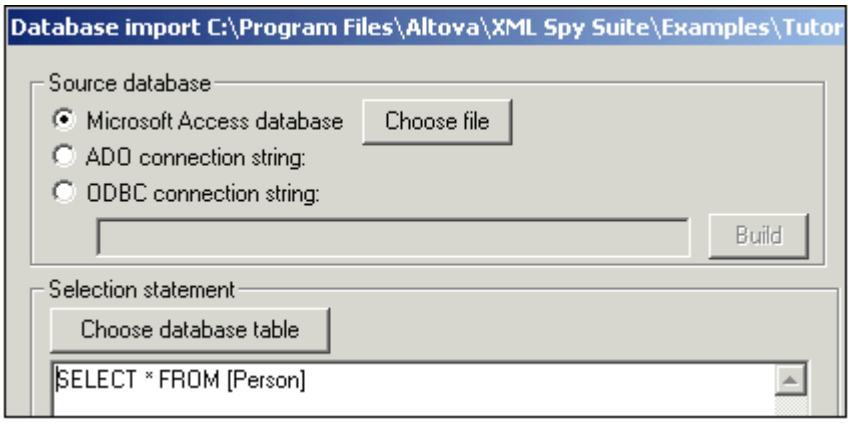
Importing database data

To import data into XML Spy:

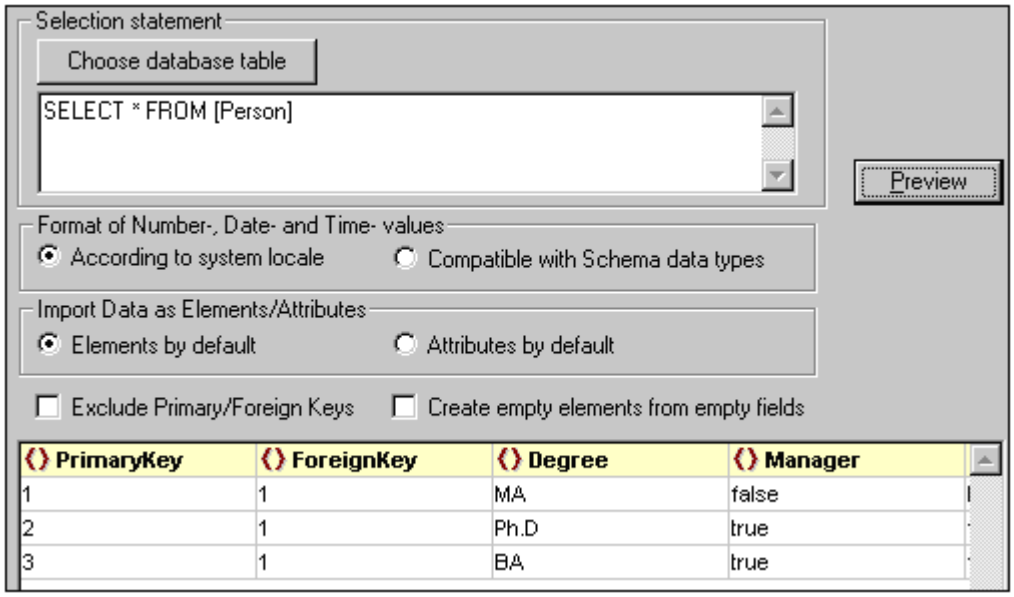
1. Select the menu option **Convert** | **Import Database data...**.



2. Click the **Choose file** button, and select the **Company.mdb** file.
   The cursor is placed in the Selection statement text box.
3. Click the **Choose database table** button, select Person and confirm with OK.
   The *SELECT * from [Person]* statement is now visible in the text box. You can extend the select statement, using standard SQL statements, to further filter the imported data.



4. Click the **Preview** button to see an example of the table data you intend to import. The Preview window displays only those records that fulfill the select criteria.



Importing fields as: attributes, elements, or skip import
The preview window allows you to directly select and define the field data you want to import.

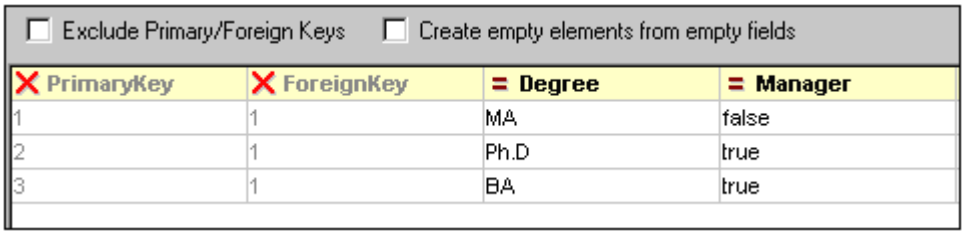
Clicking repeatedly on the **element symbol** to the left of the element name, cycles through the available possibilities:

Define and import this field as an **Element**.

Define and import this field as an **Attribute**.

Skip, do not import this field.

5. Click the **Element** symbol of the PrimaryKey column, until the **skip** symbol appears. Do the same for the ForeignKey column.
6. Click the **Element** symbol of the Degree column, until the **attribute** symbol appears. Do the same for the Manager and Programmer columns.



7. Click the **OK** button to start the import process.
8. Select the menu option **File | Save** and name the XML document (e.g. **Person-Import.xml**)

XML Spy creates an **untitled XML file** containing the Person table data. The **root element** is called Import, and each Person element has become a Row element.

Import

Row (3)

	Degree	Manager	Programmer	EMail	First	Last	Phone
1	MA	false	true	Aldrich@	Alfred	Aldrich	33
2	Ph.D	true	false	Coletti@w	Colin	Coletti	444
3	BA	true	false	Smith@w	Fred	Smith	22

Click on the Text view icon, to get another view of the imported data.

```
<Import>
    <Row Degree="MA" Manager="false" Programmer="true">
        <EMail>Aldrich@work.com</EMail>
        <First>Alfred</First>
        <Last>Aldrich</Last>
        <PhoneExt>33</PhoneExt>
    </Row>
    <Row Degree="Ph.D" Manager="true" Programmer="false">
        <EMail>Coletti@work.com</EMail>
        <First>Colin</First>
        <Last>Coletti</Last>
        <PhoneExt>444</PhoneExt>
    </Row>
    <Row Degree="BA" Manager="true" Programmer="false">
        <EMail>Smith@work.com</EMail>
        <First>Fred</First>
        <Last>Smith</Last>
        <PhoneExt>22</PhoneExt>
    </Row>
</Import>
```

For more information on importing data, please see the following sections in the reference section under **Menus | Convert Menu | Convert How to...**

- Setting up an ADO or ODBC connection
- Importing a table using ODBC
- Importing hierarchical data - ADO Data shaping
- Shape string examples

Creating a database schema

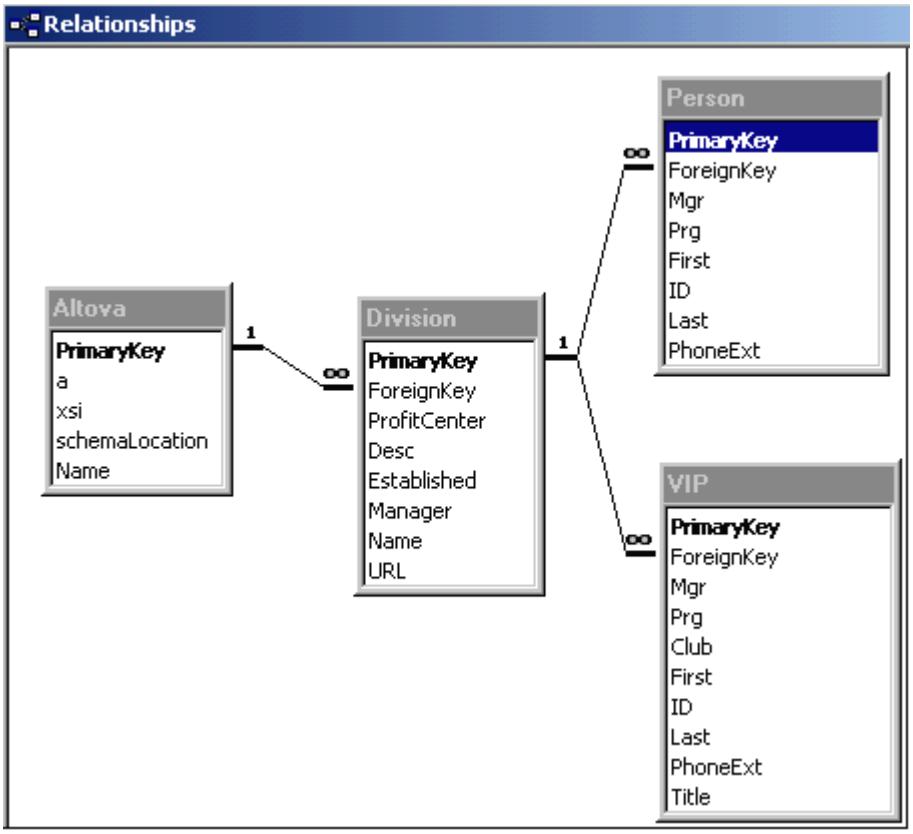
XML Spy enables you to create a schema based upon an external database file. Microsoft Access databases, as well as ADO and ODBC compatible databases, are supported.

Goal of this section:
To convert an existing MS Access database into a schema file, having the same table structure.

This will be achieved by:

- Using the menu option **Convert | Create Database Schema**, to create the schema in XML Spy.

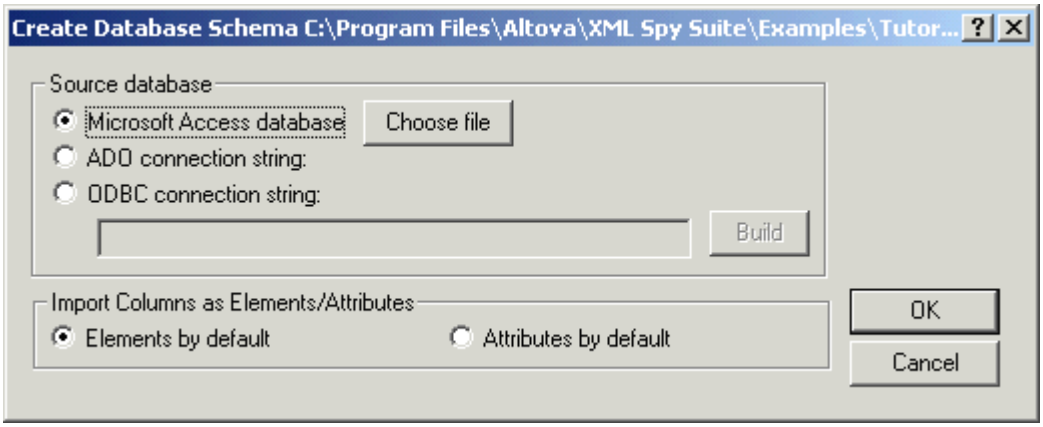
- This example uses the **DB2Schema.mdb** file supplied with this tutorial. The Relationships view of the DB2schema.mdb file is visible in the diagram below. Use the menu option **Tools | Relationships** in MS Access, to view the relationships.



Converting a database to a schema

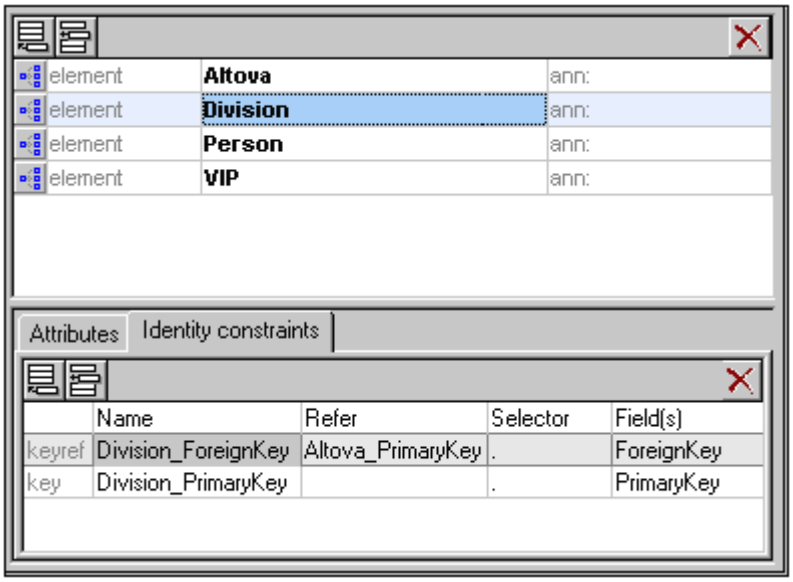
To create a schema from a database file:

1. Select the menu option **Convert | Create Database Schema.**

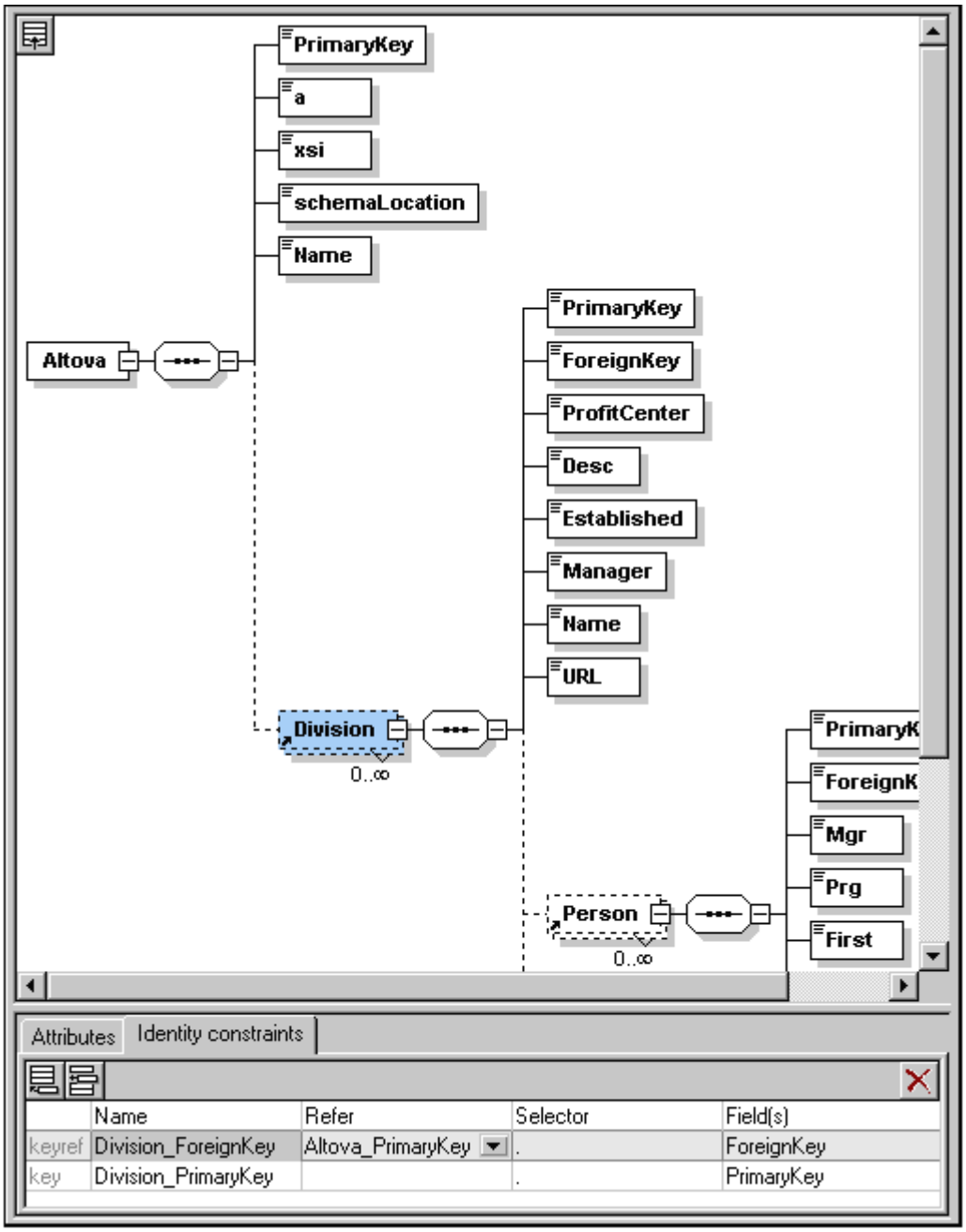


2. Select **Microsoft Access database**, and click the Choose file button.
3. Select the **DB2schema.mdb** file supplied with XML Spy, and click the **Open** button.
4. Click the **OK** button of the Create Database Schema dialog, to start the conversion process.

   The generated schema appears in the Schema Design View. Click the "Identity constraints" tab, to see the keyref and key fields of the respective elements.



5. Click the component icon next to the **Altova** global element, to see the content model.



6. Select the menu option **File | Save as**, and save the new schema e.g. **DB2schema.xsd**.
7. Click the Display all globals icon, to return to the schema overview.

Please note:

When generating the schema, all namespace prefix colons are automatically converted into underscore characters.

Databases currently supporting the key and keyref fields:

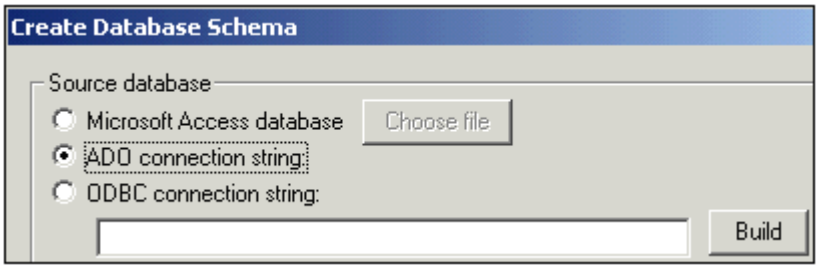
MS Access and several other databases are able to automatically provide the key and keyref information for the ADO driver, used to create the database hierarchy.

Please note:

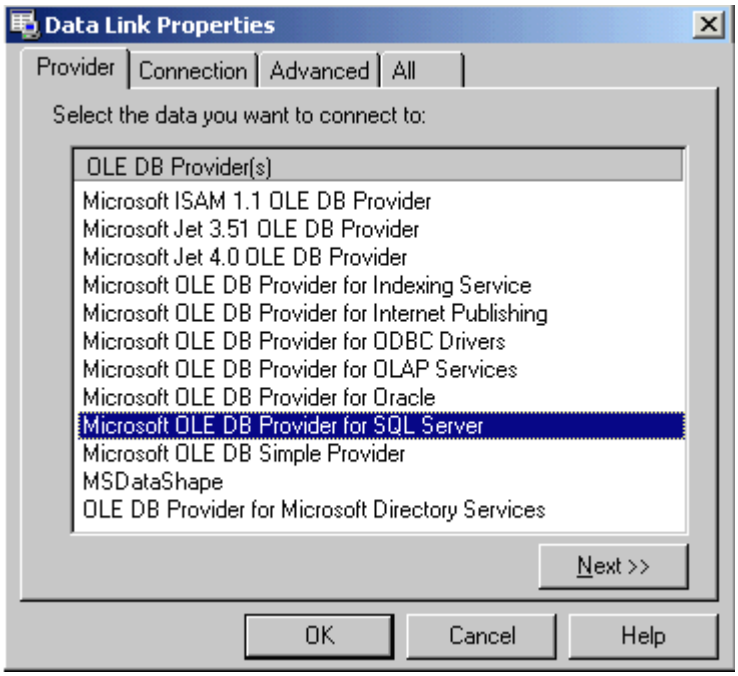
The following text describes in condensed form, how to create a database schema using other types of databases. This text is added for the sake of completeness, and is not a tutorial task. Please contact your database administrator for further information regarding the setup and use of these databases!

To create relationships for NON MS Access databases:

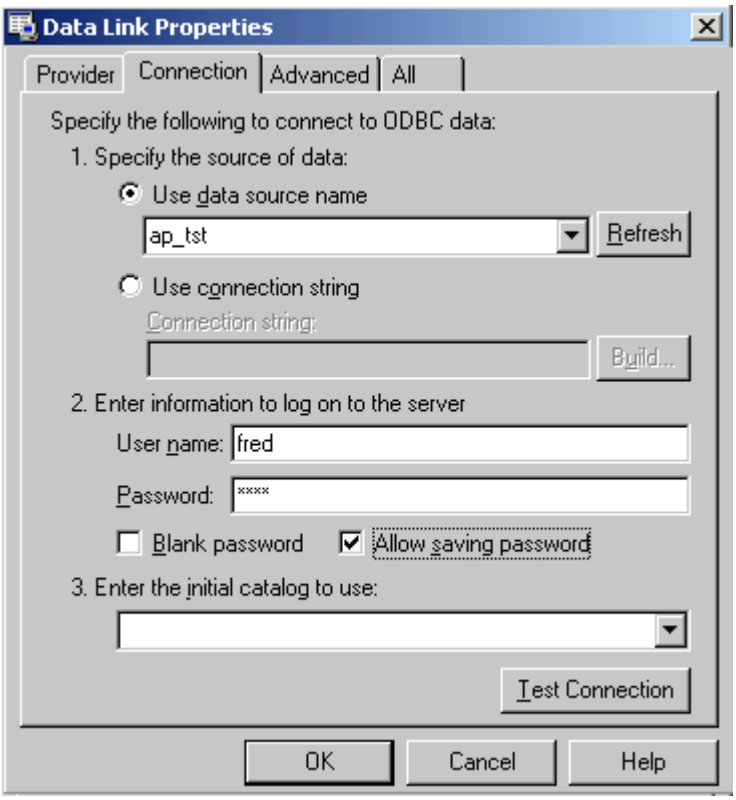
1. Click the "**ADO connection string**" radio button in the Create Database Schema dialog box.



2. Click the **Build** button that has now become active.
   This opens the Data Link Properties dialog box.
3. Select the corresponding **Microsoft OLE DB Provider** (or vendor specific provider) for the database you use, do not select one of the generic drivers. Please see the "To convert from..." list at the end of this section.



4. Click the **Next** button to switch to the Connection tab and fill in the required information: the data source, the user name and password, and activate the **Allow saving password** check box.
5. Click the Test Connection button to test the connection, and Click OK to confirm the settings.



6. Click OK in the Create Database Schema dialog box, to create the schema.

To convert from... SQL server databases:

- Select the **Microsoft OLE DB provider for SQL server** provider.

To convert from Oracle... databases:

- Select the **Microsoft OLE DB provider for Oracle** provider.

To convert from... MS Access:

- Click the **Microsoft Access Database** radio button in the Create Database Schema dialog box. This selects the correct provider, there is no need to use the ADO connection string and Data Link Properties dialog box.
- If however, you want to build the connection string yourself, please use the **MicrosoftJet 4.0 OLE DB** provider.

To convert from... Other databases:

- Select the corresponding Microsoft OLE DB, or vendor specific provider, from the Data Link Properties dialog box.

To convert from... databases without a specific provider:

Other databases will create a flat structured schema, including all tables and their corresponding datatypes.

- Use drag and drop in the schema overview, to create the necessary relations between the imported elements.
- To create an element hierarchy you have to directly edit the key and keyref fields, visible in the Identity constraints tab. Please see "Creating Identity Constraints" in the Reference manual for more information.

Creating a project

Goal of this section:
To create an XML Spy project that includes all the files currently open in the main window.

This will be achieved by:

- using the menu option **Project** to create a project folder
- using the specific Project commands to add files to the project

Functions (and their icons) in this section:

Project | New Project
Project | Add active and related files to project
Project | Add active file to project

Advantages of Projects:

- Files and URLs can be grouped into folders by common extension or any other criteria
- Batch processing can be applied to specific folders, or the project as a whole
- A DTD / Schema can be assigned to specific folder, allowing immediate validation
- XSL transformations can be assigned to specific folders, allowing immediate transformations
- The destination folders of XSL transformation files can be specified

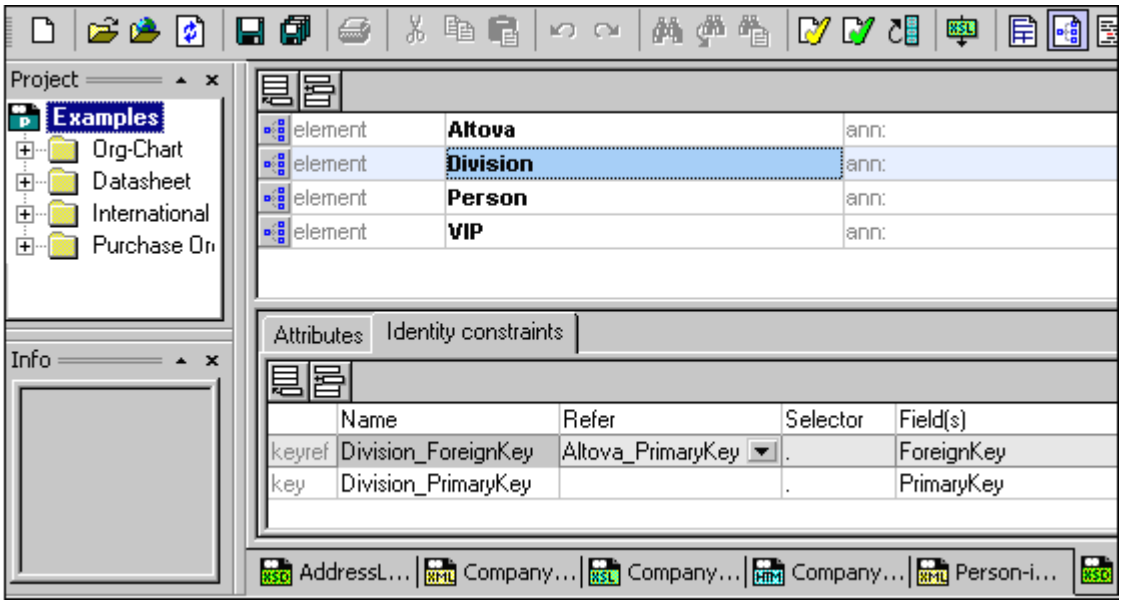
  These settings can all be defined by using the menu option **Project | Project Properties**.

- XML files can be placed unter Souce control using the menu option **Project | Source control | Add to source control** (Please see the Reference manual or online help for more information).
- Personal, network and web folders can be added to projects, allowing batch validation.

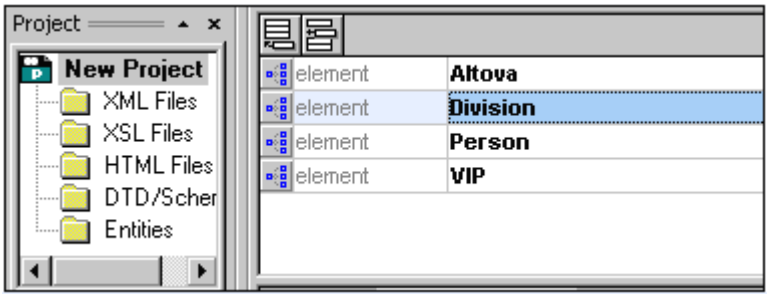
Adding files to a project

Creating and adding files to a Project:

There are now many different files open in the main window. You can access these files as an unit by grouping them into an XML Spy project.

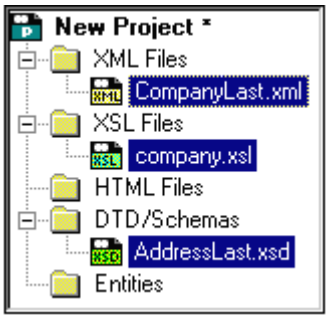


1. Select the menu option **Project | New Project**.



The project window now contains several folders under the New Project folder.

2. Click the **CompanyLast.xml** tab, to make it the active file in the main window.
3. Select the menu option **Project | Add active and related files to project.**

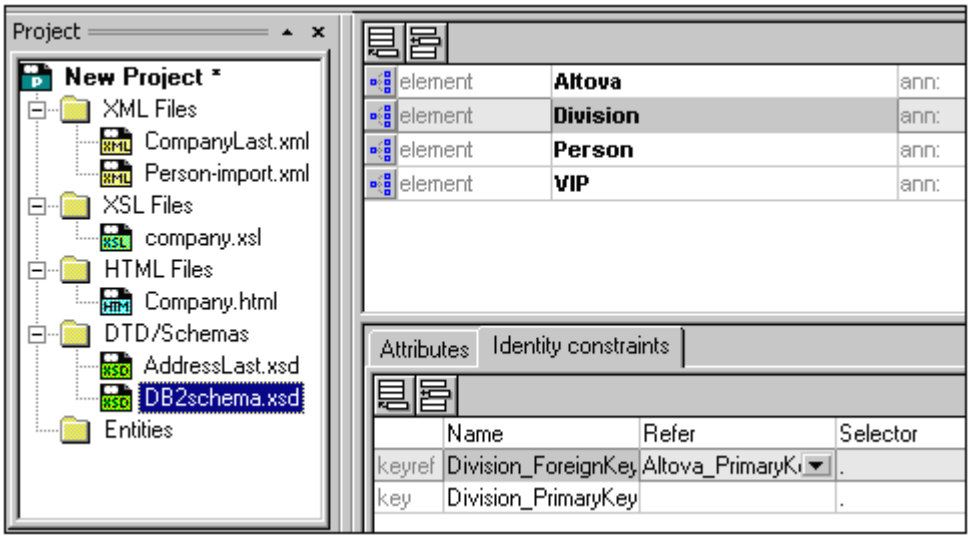


Three files have been added to the New Project folder: CompanyLast.xml, Company.xsl and AddressLast.xsd.

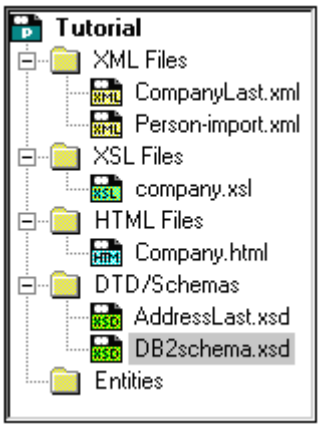
4. Click the **Person-import.xml** tab, and then select the menu option **Project | Add active**

file to project.

5. Use the same method to add the **Company.html** and **DB2schema.xsd** files to the project. The project should now look like the diagram below.



6. Select the menu option **Project | Save Project** and enter "**Tutorial**" as the project name.



To delete files from a project:

1. Click on the file you want to delete in the project window, and hit the **Delete** key.

To add new folders to a project:

1. Select the menu option **Project | Add folder to Project**, and fill in the Properties dialog.

To add a file to a specific folder:

1. Click the file to make it active in the main window.
2. **Right click** the **folder** you want to place the active file into (in the Project window), and select the menu option "**Add active file**".
   This method allows you to add a file to any folder in the current project.

That's it !

If you have come this far congratulations, and thank you!

We hope that this tutorial has been helpful in introducing you to the basics of XML Spy. If you need more information please use the context-sensitive online help system, or print out the Tutorial.PDF file that accompanied your version of XML Spy.

If you have any comments regarding the tutorial, please feel free to contact us at support@xmlspy.com.

Index

- A -

Accessing 33
elementTypes 33
Add 7
child element 7
elements with drag and drop 10
files to project 69
sequence compositor 7
Add to source control 68
ADO connection string 64
Append 22
attribute 22
attribute / element 44
Assign 54
XSL 54
Attribute 22
adding 22
append 22
element 22
entry helper 36, 44
enumerations 22
import 60
limiting contents 22
window 22

- B -

Base 16
element 16
Browser view 54

- C -

Check 40
validity 40
well-formedness 40
Child element 7
adding 7
Com 16
Com(plex) tab 16
type 16
type extending 16
Component 16
global 16
icon 7
navigator 4
Compositor 7
sequence 7
Configure 11
schema design view 11
Connection string 64
ADO - create DB Schema 64
Constraints 64
identity 64
Convert 64
create database from a schema 64
export to text / database 58
import database data 60
Copy 47
data to/from 3rd party prod. 47
Create 69
addtnl. project folder 69
new schema 4
new XML document 33
project 69

- D -

Database 64
create schema from 64
export restrictions 58
export XML data to 58
hierarchy - keyref 64
import attribute 60
import element 60
OLE DB provider 64
primary / secondary key 60
skip field 60
SQL ,Oracle etc. 64
Database / Table view 47
sort data 47
Declarations 22
global 22
Delete 33
element 33
file from project 69
Details 7
entry helper 7
tab 12

© 2002 Altova

Display all Globals 7
icon 7
Display as Table 47
Document 40
save invalid doc. 40
schema 28
Drag and drop 10
add elements with 10
DTD / Schema 33

- E -

Edit 36
intelligent editing 36
Element 7
adding 7
attributes 22
base 16
deleting 33
display (shortcut) 25
facet 12
import 60
mandatory 40
optional 12
reference 22
restricting definition 16
accessing 33
Elm 4
Element (global) 4
Enhanced Grid 47
Database / Table view 47
view 36
Entry helper 2
attribute 36, 44
component navigator 4
Details 7
facet 12
insert attribute / element 44
update 44
Enumeration 22
attribute 22
Example files 1
tutorial 1
Excel 47
copy data to/from 47
Export 58
database data 58
parameters 58
restrictions 58
Extend 16
complex type definition 16

- F -

Facet 12
element 12
entry helper 12
updating schema 47
Folder 69
create addtnl. project folder 69

- G -

Global 16
component 16
declarations 22
Go to 25
Element definitions (schema) 25
XSL 54
Grayed-out 36
attribute / element 36
Grouping files 68
project 68

- H -

HTML 54
from XML doc. 54

- I -

Identity constraints 64
Import 60
attribute 60
auto root element 60
database data 60
element 60
primary & secondary keys 60
select statement 60
skip field 60
Info 2
window 2
Insert 44
attribute / element 44
Intelligent editing 36

© 2002 Altova

Invalid 40
document - save 40
isRef 22
reference 22

- K -

Keyref 64
and key - database hierarchy 64

- L -

Limit 22
attribute contents 22

- M -

Main window 2
Mandatory 40
element 40
Marquee 10
drag and drop 10
maxIncl 12
facet 12
maxOcc 7, 12
details 7, 12
minOcc 7, 12
details 7, 12

- N -

Namespace 4
schema 4
Navigation 25
shortcuts in schemas 25

- O -

OLE DB Provider 64
Optional 12
element 12
Oracle 64
create DB Schema 64
Overview 2

- P -

Primary 60
& secondary key - import 60
Project 69
adding files to 69
advantages 68
create 69
create addtnl. folder 69
delete file from 69
properties 68
save 69
window 2
Provider 64
OLE DB and vendor specific 64

- R -

Reference 22
element 22
Refresh 44
entry helper 44
Restriction 16
element 16
export 58
Revalidate 40
Root element 60
on importing data 60
schema 4
selecting 33

- S -

Save 40
anyway 40
invalid document 40
project 69
Schema 7
adding elements 7
creating from a database 64
creating from a database - Overview 63
creating new 4
definition 3
documentation 28
namespace 4
root element 4

© 2002 Altova

Schema 7
Schema / DTD 33
settings 4
shortcuts 25
view 4
Schema overview 4
configuring 11
display all globals 7
Secondary 60
& primary key - import 60
Select statement 60
import 60
Sequence 7
compositor 7
unbounded 7
Settings 4
schema 4
Shortcuts 25
in schema documents 25
Show 47
Database / Table view 47
Simple type 12
Skip 60
database field 60
Sort 47
table columns 47
SQL 64
create DB schema 64
select statements import 60
String 64
connection string 64
Syntax check 40
well-formedness 40

- T -

Tab 16
Com(plex) element 16
details 12
Elm (global element) 4
Template folder 1
Text view 36
Transformation 54
XSL - HTML 54
Tutorial 1
example files 1
goals 1
Type 16
AddressType 16
complex 16
simple 12

- U -

Unbounded 7
element 7
Update 44
entry helper 44
schema facet 47

- V -

Valid 40
invalid document - save 40
check validity 40
marks incorrect data 47
revalidate 40
XML document 40
View 54
Browser view 54
content model (shortcut) 25
Database / Table view 47
Enhanced Grid 36
Schema view 4
Text view 36

- W -

Well-formedness 40
Windows 2
overview 2

- X -

XML 47
copying to 3rd party prod. 47
document and icons 32
new document 33
sorting data 47
validate document 40
xs 22
xs:boolean 22
xs:date 12
xs:integer 12

© 2002 Altova

xs 22
xs:positiveInteger 16
xs:string 7, 12
xsi:type 33
XSL 54
assigning to an XML doc. 54
go to XSL 54
transformation 54

© 2002 Altova

Back Cover