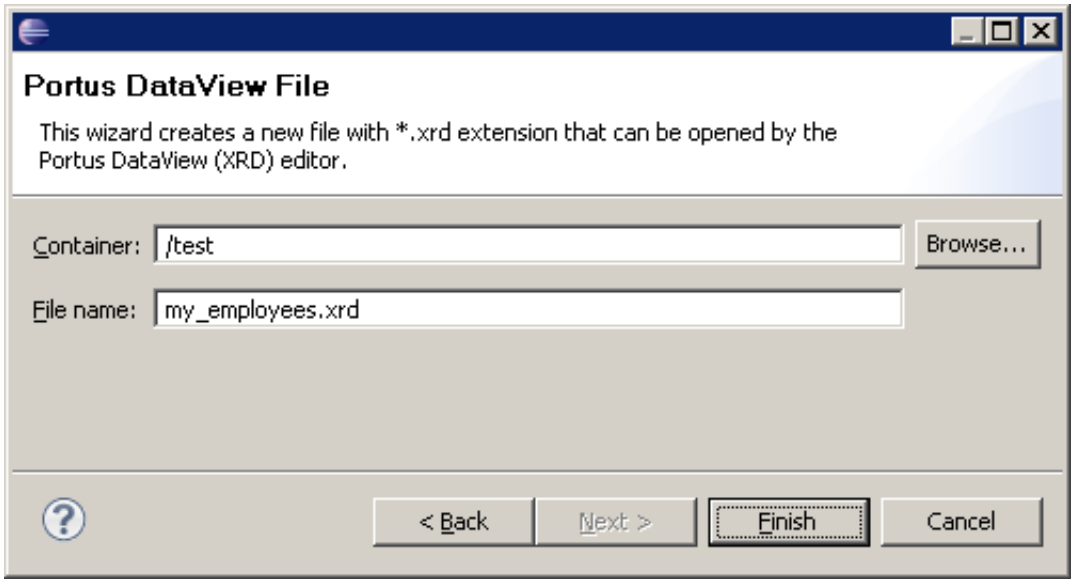
SOA Gateway DataViews

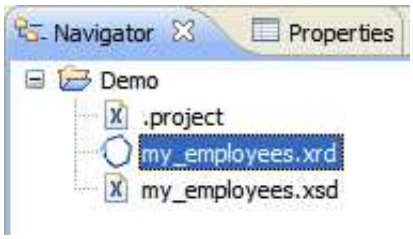
- Create a DataView from scratch
  - adding MU (muliple value) fields
  - adding PE (periodic group) fields
  - adding "special" fields (Adabas Super-Descriptors etc.)
- Opening a SOA Gateway DataView for editing
- Editing a SOA Gateway DataView
- Export a DataView to a SOA Gateway server
- Importing an existing SOA Gateway DataView
- Creating a XML Schema (XSD) for a DataView

Create a DataView from scratch

1. A vanilla SOA Gateway DataView is created with an Eclipse "**New Wizard**", start it with **File -> New -> Other** (or use the shortcut Ctrl+N) to bring up the list of available wizards. Select **Other-> SOA Gateway DataView** from this list. Click **Next**.
2. Enter or select a Destination folder, specify a name for the DataView file, click **Finish** to create it.



3. You have now created an "empty" (i.e. no fields defined yet) SOA Gateway DataView file in the selected folder.

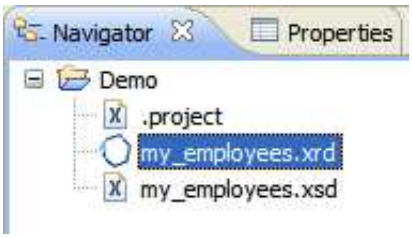


4. For information on actually editing the DataView, adding fields etc., please continue reading at Edit a SOA Gateway DataView

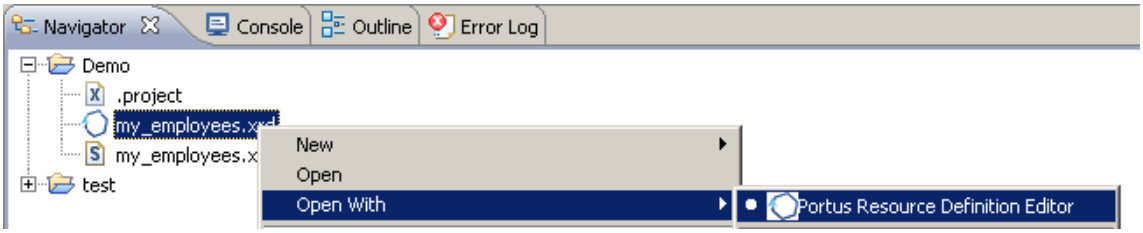
Opening a SOA Gateway DataView for editing

1. Open a "local" file, contained in an Eclipse project within the active workspace...

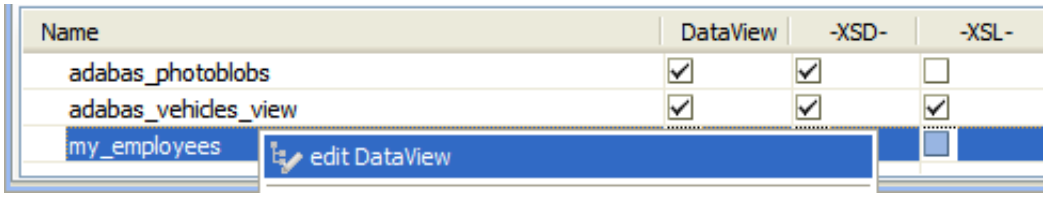
   - by double-clicking on it’s name in the Package Explorer or Navigator



   - by right-clicking the DataView (.xrd) file, Open With -> SOA Gateway Resource Definition Editor



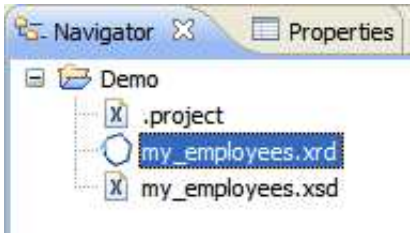
2. Open a "remote" DataView file, directly on the server, without importing it into a workspace / project first, by right-clicking the DataView name in the Configuration View to bring up the context menu, then select "edit DataView"



Editing a SOA Gateway DataView

1. For the purpose we will open an empty DataView file and populate it with all information required to be able to start issuing requests against an Adabas Resource (= Adabas file on an Adabas database). This tutorial will be based on the Adabas "Employees" demo file.

DataViews are not tied to a specific "resource type", the same mapping can be used to access an Adabas file or a SQL table. There are, however, elements of a DataView which are only meaningful in the context of a specific resource type, for example "special descriptors" (super-, sub-, ...) for Adabas.

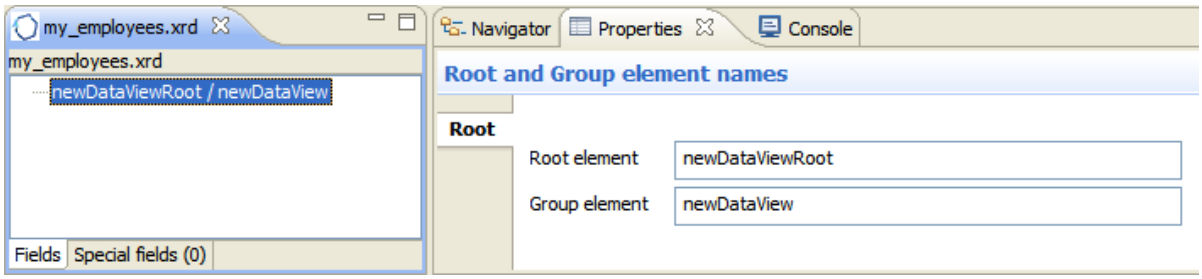


2. The display areas relevant for editing the DataView file are

   - the actual editing area tab, showing the DataView file name in it’s tab header
   - the Properties area

   In case the "Properties" view somehow got hidden, right-click into the edit window and select "Show Properties View" from the context menu.

   The result should look like this:

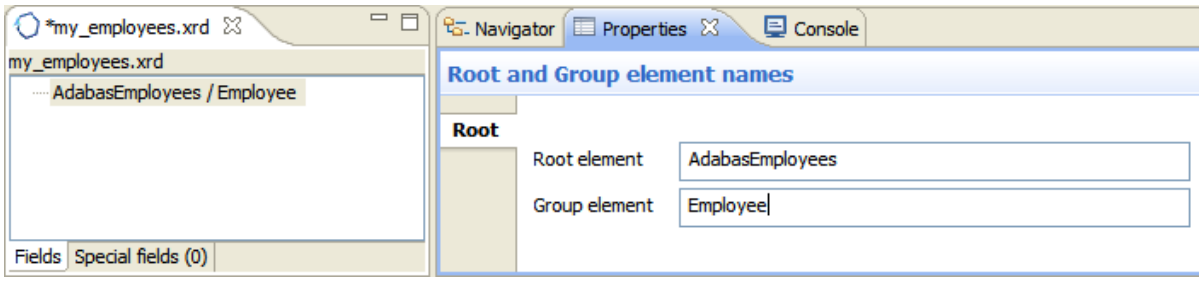


3. First of all, enter the Root element name: The (XML) "structure" or "set" name under which items (records) for a Resource linked to this DataView will be referred to. E.g.: AdabasEmployees

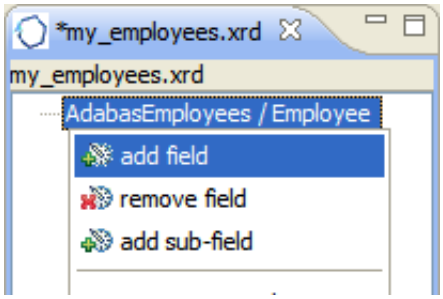
   Enter the Group element name, this is the SOA Gateway "record name". (E.g.: Employee).

   Changes applied to name fields in the Properties view are reflected in the editor’s tree view immediately.

   Important:
   The ’Root’ and ’Group’ element names may NOT be identical, as this would lead to duplicate XML element names and thus ’loops’ in the schema.

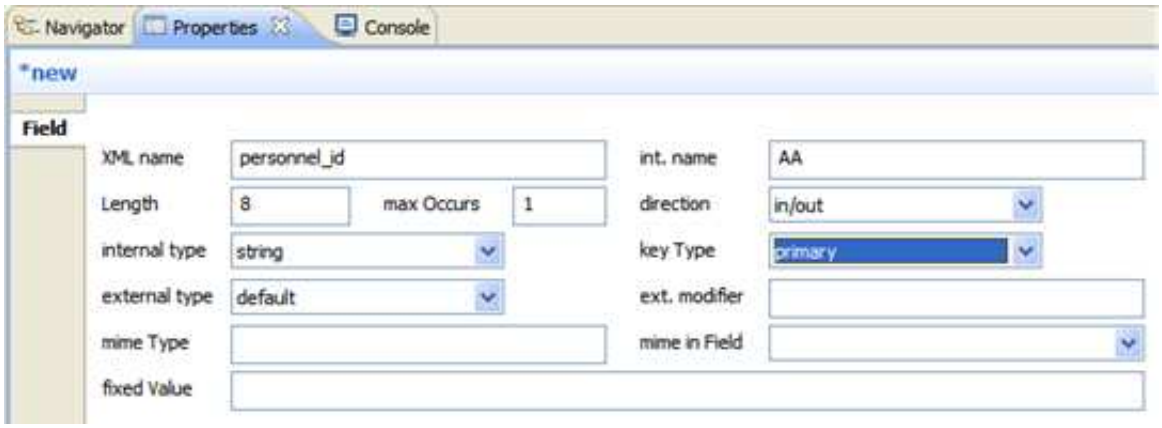


4. We can start adding the actual DataView Definition elements now, right-click on the root element in the editor window, from the context menu appearing now select **add field**.

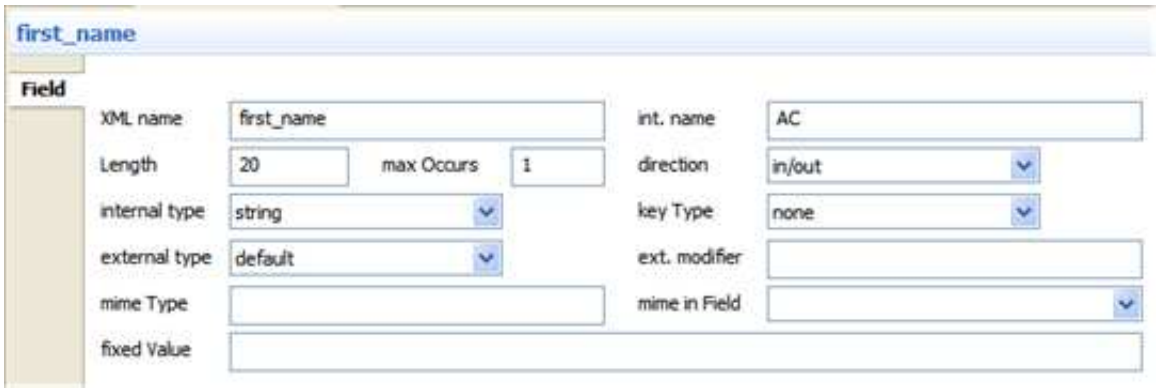


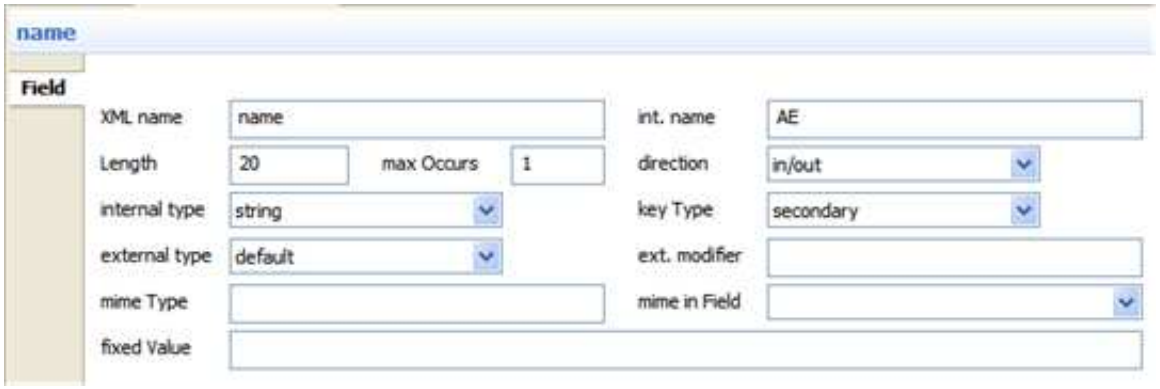
5. A new element ("Field") has been added with all properties set to initial values.

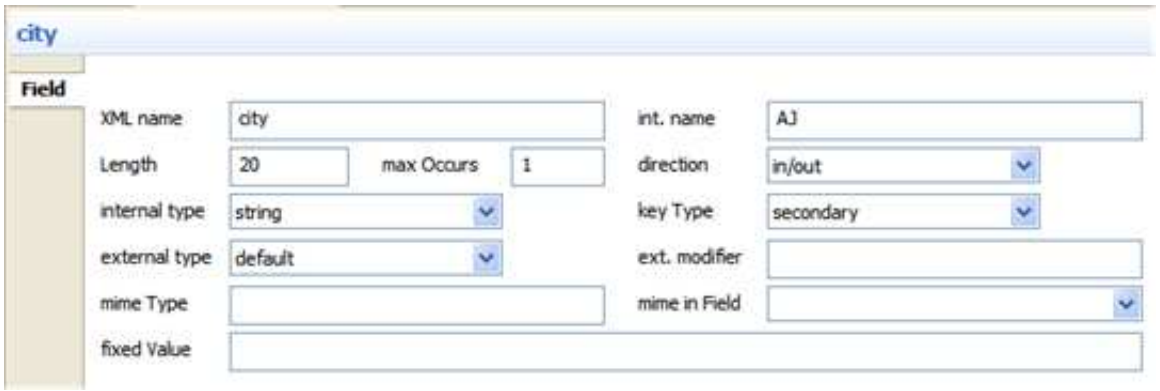
   The first field we are going to add is the "personnel Id", set the properties as follows:



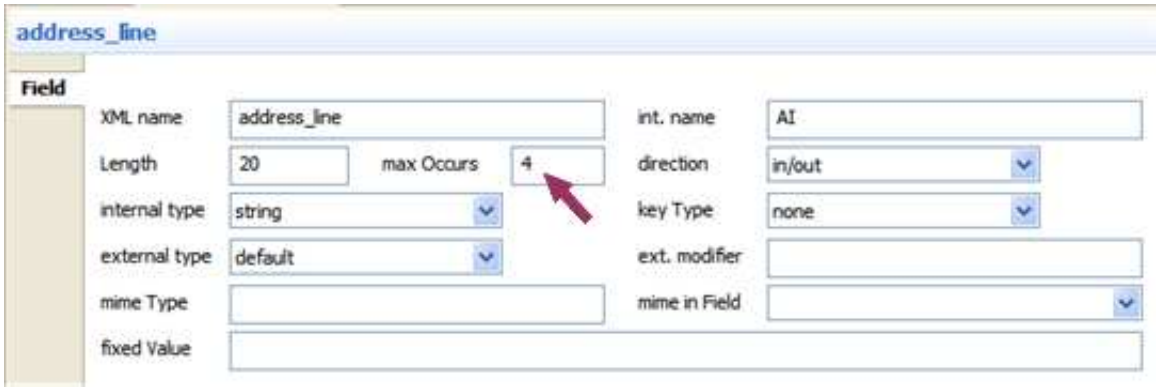
6. Add a few more fields with the following attributes:







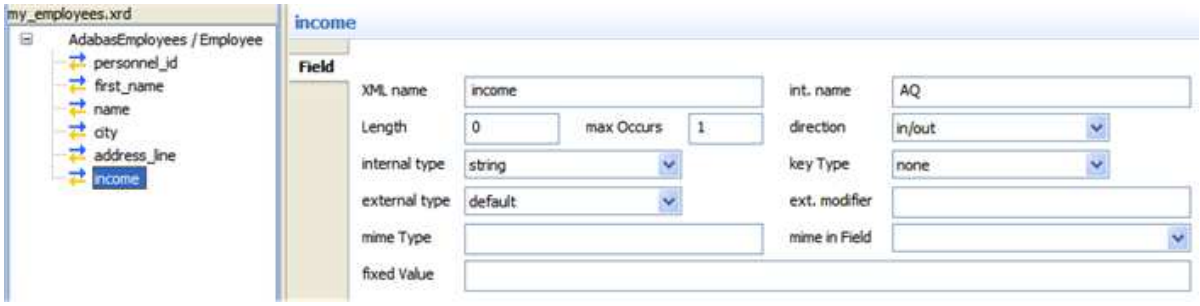
Next we are going to define the "address line" field, which is a MU (multiple value) field. MU fields are defined like a "flat" field, with the exception of the max Occurs Property being set to a value >0



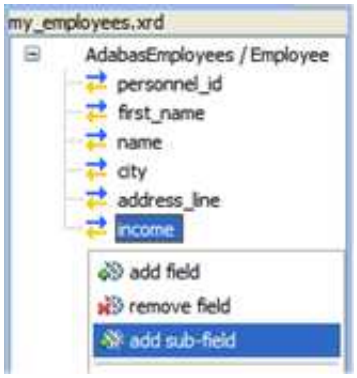
7. Lastly, we will define a PE (periodic group), the structure of the "income" group of the "Employees" file:

   - currency Code (simple field)
   - annual Salary (simple field)
   - annual Bonus (MU field)

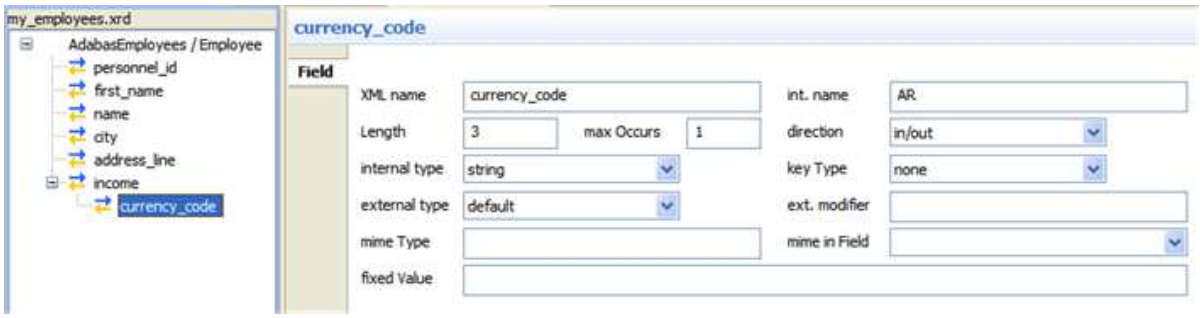
   So we first add the "group field" income. Only the xml Name, int. name (internal name = Adabas field name) and max Occurs properties are relevant here.

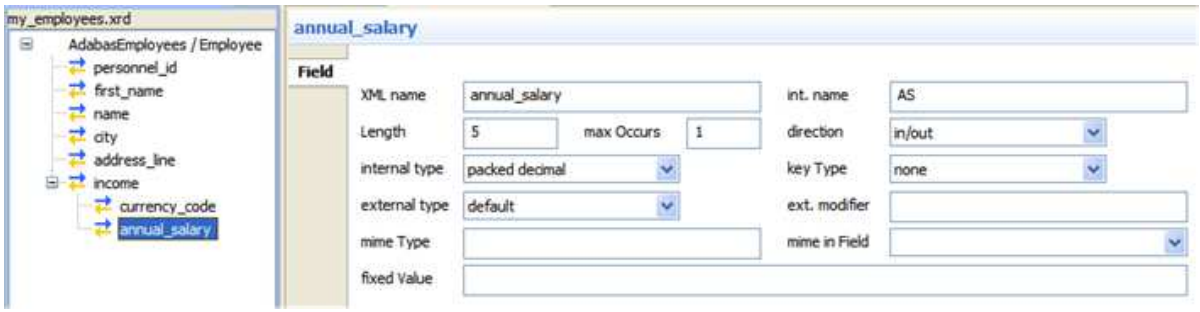


To actually turn the income field to a PE group field, right-click it, select **add subfield**

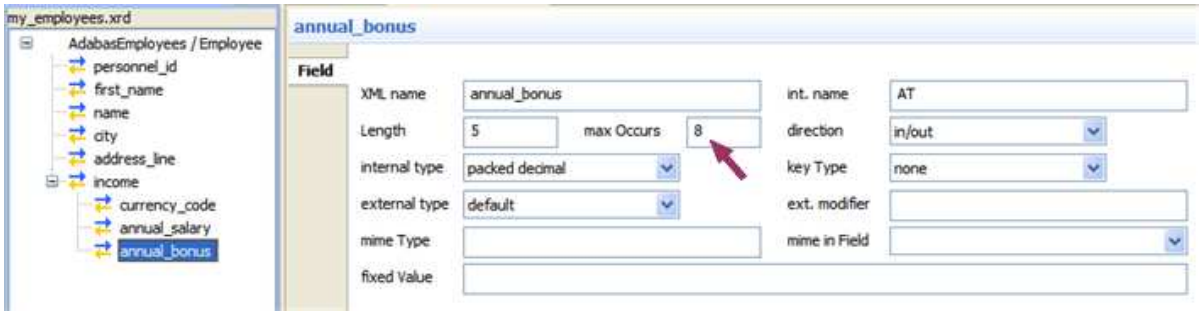


PE Sub Field properties are equivalent to those of "regular" fields. Define the three PE-fields as follows:

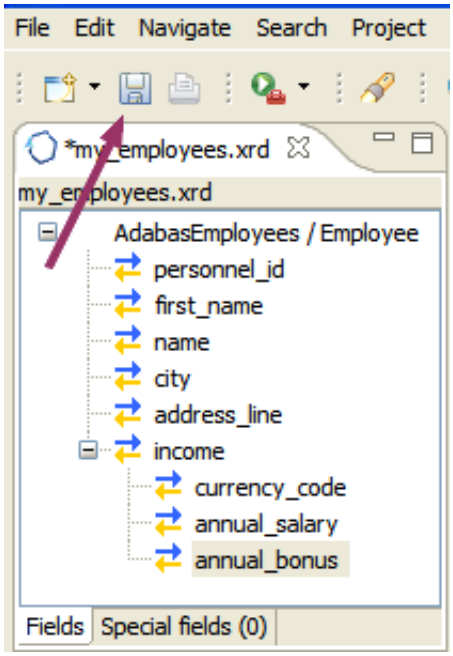




max Occurs >0 on the PE-field level denotes a MU within a PE.



8. The "view" to the "Employees" file is now complete. Save the DataView by selecting the Save button or Ctrl+S.



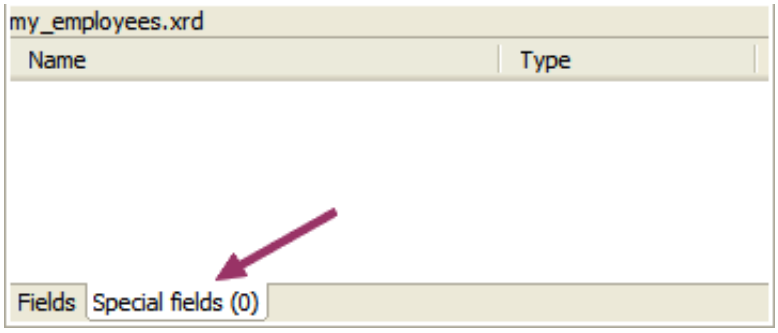
9. Export the DataView to the target SOA Gateway server.

Editing "special" fields

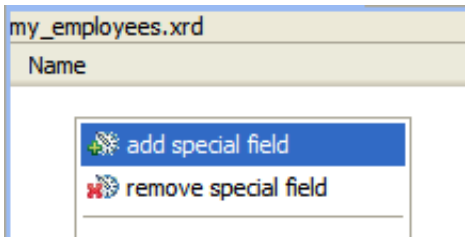
The following "special" fields can be defined for a DataView

- Adabas SuperDescriptor
- Adabas SubDescriptor
- Adabas HyperDescriptor
- Adabas Phonetic Descriptor

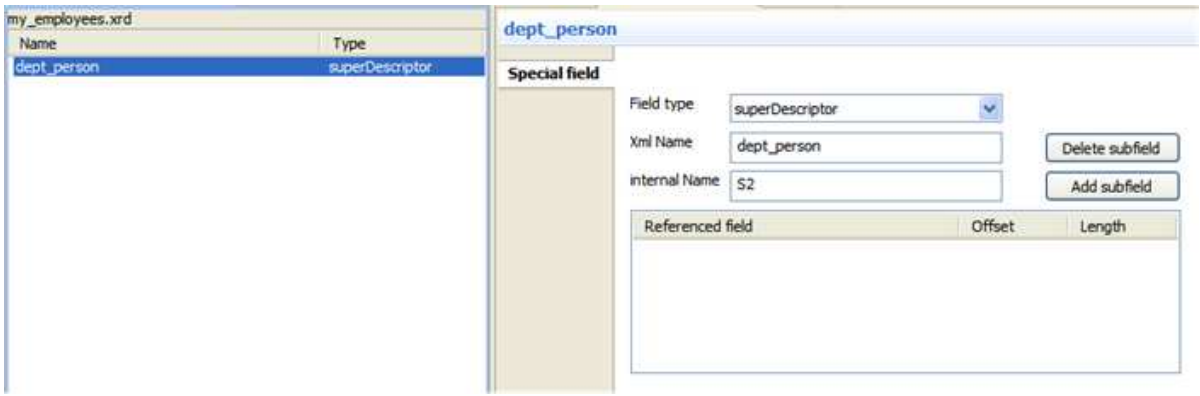
1. In the DataView editor window, click on the "Special fields" tab to add special fields



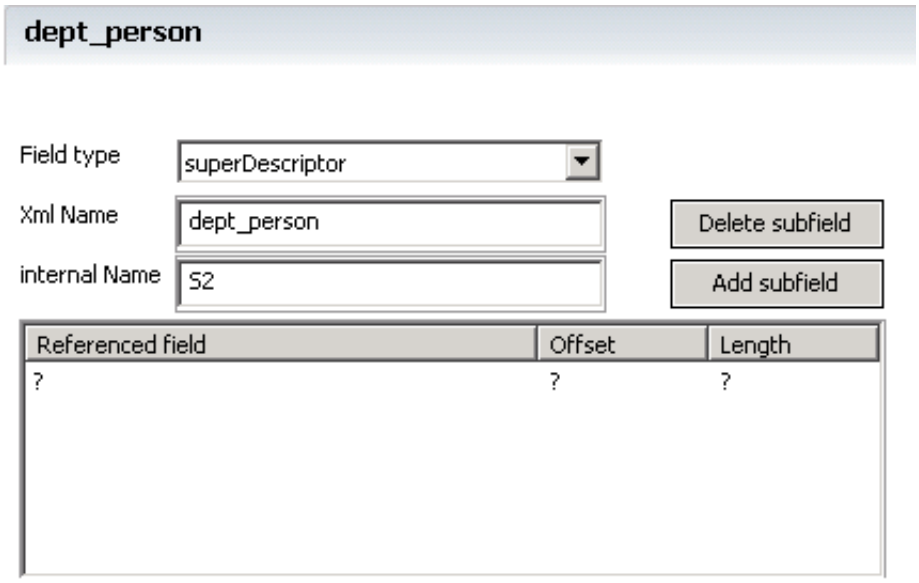
2. Right-click into the empty editor area, select "add special field"



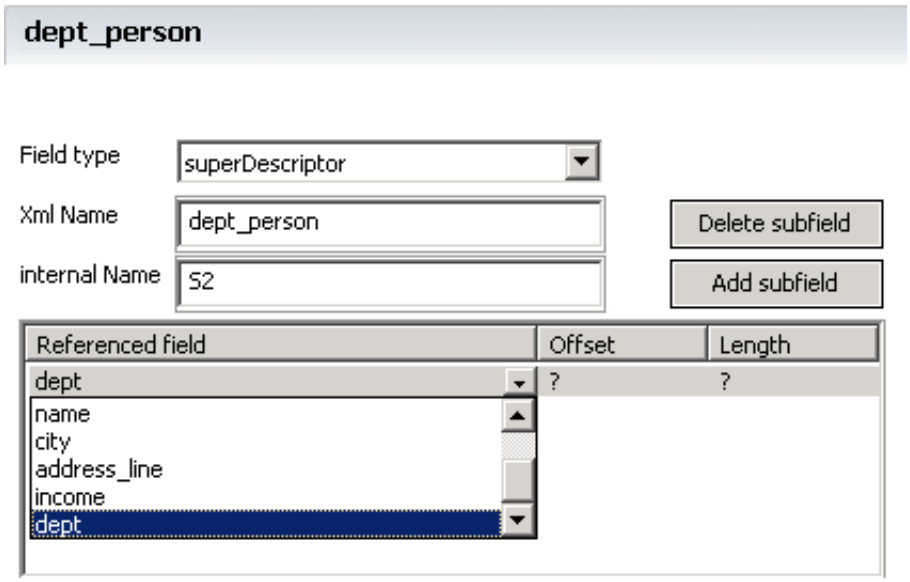
3. For example we define a special field called "dept_person" with an "internal name" (the Adabas short name) of "S2", being of type "superDescriptor"



4. Click on the "Add subfield" button, a new field element will appear in the subfield table

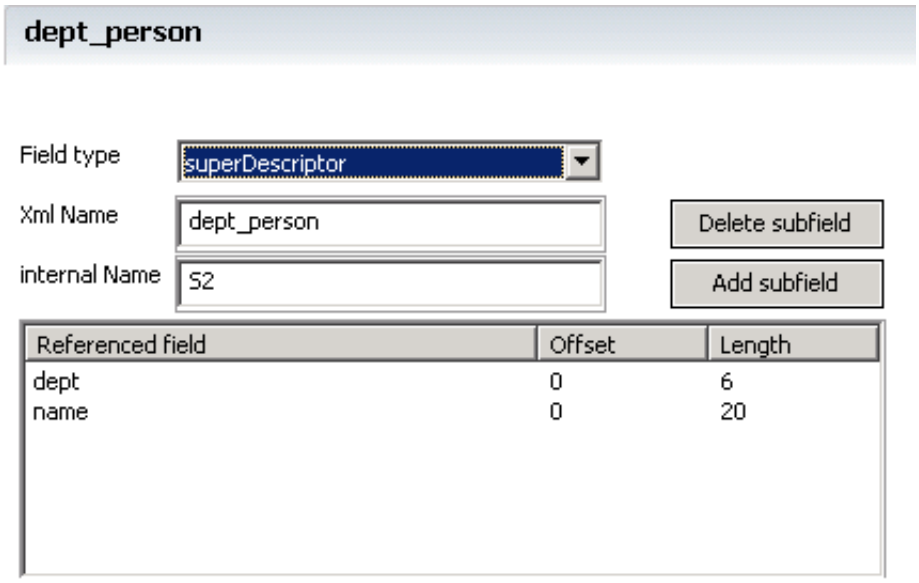


5. Click on the field value under the "Referenced field" heading, select the field to be added from the list of fields in the dropdown-box. Here we select the "dept" field



Initial "Offset" and "Length" values will be derieved from the selected field’s definition

6. Select the "name" field as the second subfield just like the "dept" field, the result should look like this



Enhanced Type Conversion

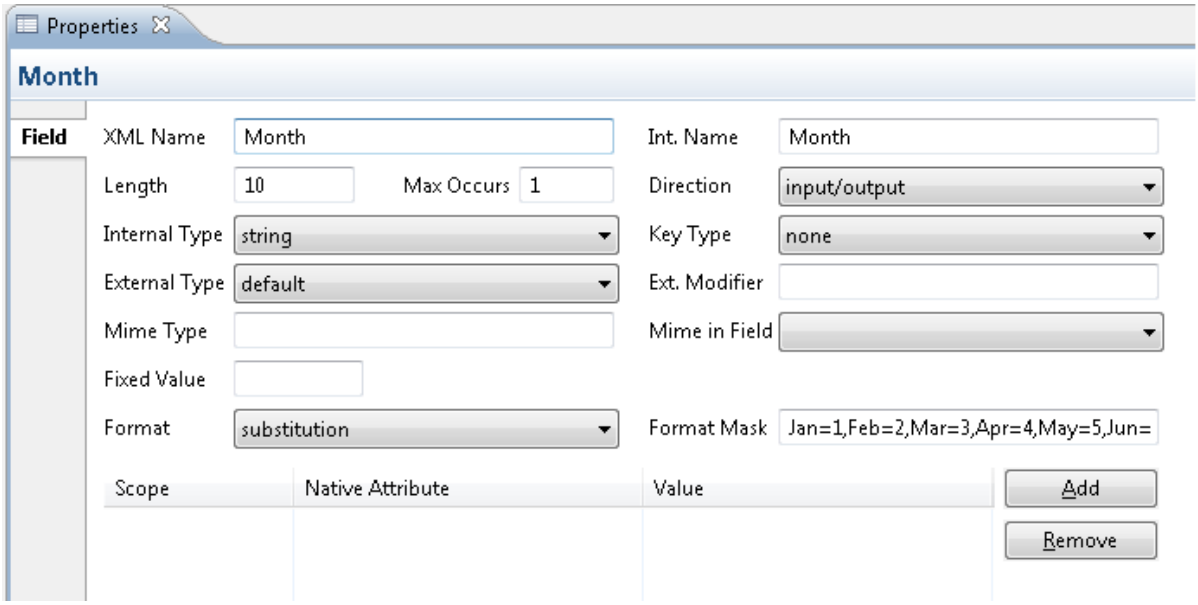
This facility allows the conversion of a string value to an integer equivalent and visa versa. This is analogous to an enumeration i.e. for Jan substitute 1, Feb substitute 2 etc.

In the DataView editor window, select the field.

In the Properties View 2 items need to be changed

1. Open the Format dropdown list and select *substitution*.
2. Set the Format Mask field to the value of the enumeration string. This should be in the format strvalue1=num1, strvalue=num2, strvalue3=num3 e.g.

   Jan=1,Feb=2,Mar=3,Apr=4,May=5,Jun=6,Jul=7,Aug=8,Sep=9,Oct=10,Nov=11,Dec=12



To save select Ctrl+S or close the DataView editor.

Right-click on the Service and select ’Refresh Service’.

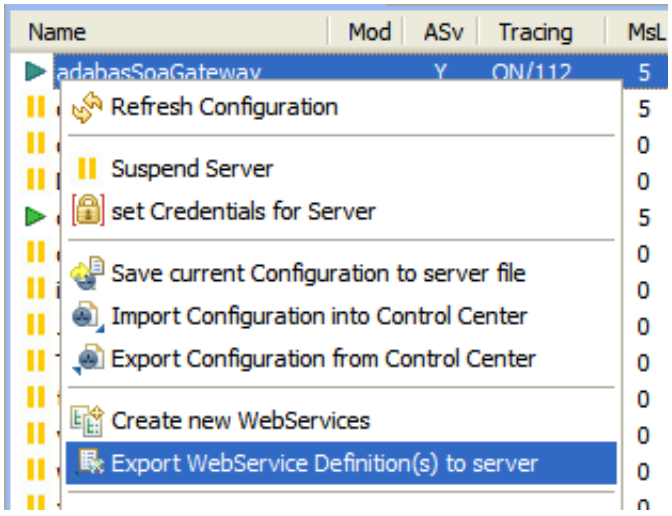
Exporting a DataView to a SOA Gateway server

Whether you created a new one or imported and edited an existing DataView, you will now need to export it to the SOA Gateway server. DataViews are stored in the "xrd" subdirectory of the server configuration directory.

To export a SOA Gateway DataView to a SOA Gateway server execute one of the the following procedures:

Using the server based export function

1. Select **Export Resource Definitions** from the context menu of the server you wish to export to.

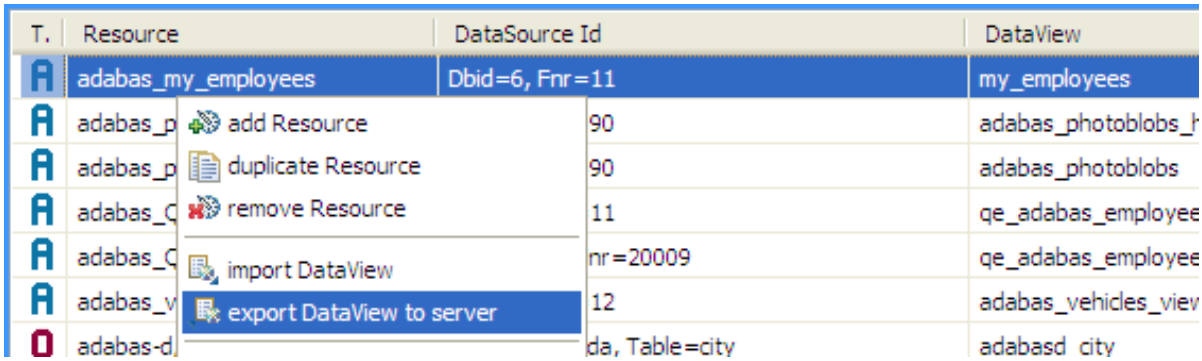


2. Select the DataView to be exported from the file selection dialog, click **OK**

3. The newly added DataView appears in the list

Using the configuration based export function

1. On the configuration view select **export DataView to server** from the context menu of a Resource definition pointing to the DataView to be exported.



2. Select the DataView to be exported from the file selection dialog, click **OK**

3. You are now asked if the DataView is to be activated immediately.

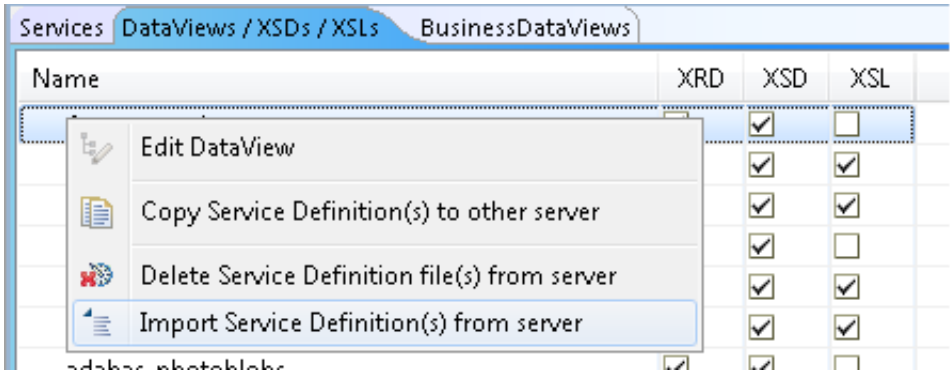
   Reply **Yes** if you want to use the newly exported DataView immediately, that is as soon as all requests currently using that DataView have completed, the new copy will be used for all subsequent requests.

   A successful export will be indicated by a message in the status line.

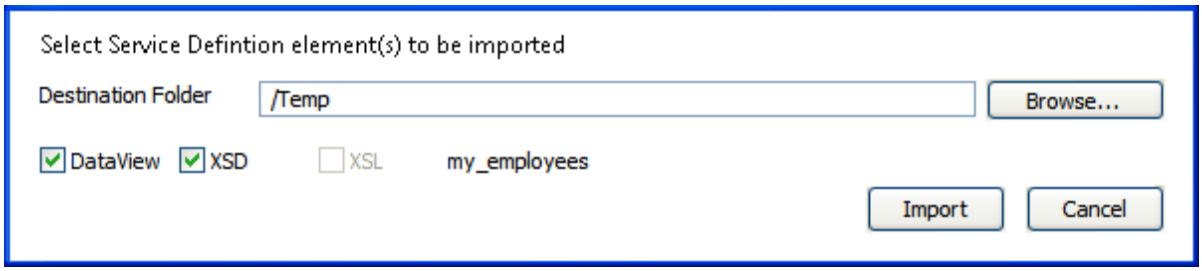
Importing an existing SOA Gateway DataView

To import an existing DataView:

1. From the SOA Gateway Configuration View's "DataViews / ..." tab, select the elements and right click. Select "Import Service Definition(s) from server"



2. Select the destination folder and click "Import"



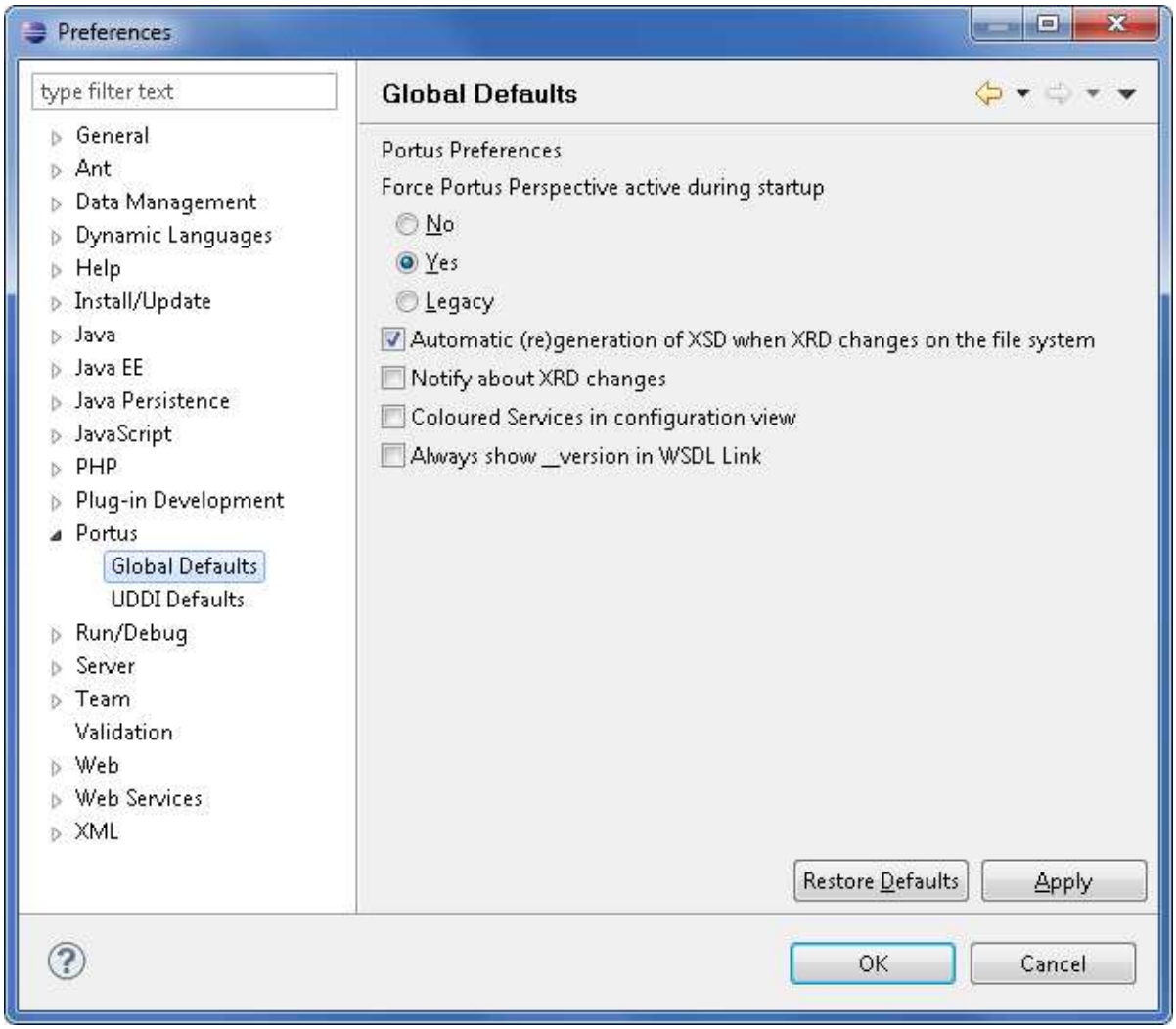
Creating a XML Schema for a DataView

An XML Schema (XSD) can be used to express a schema: a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema. In SOA Gateway an XML Schema can be used to validate the input coming from the user. This validation occurs at a very early stage of the processing, so this can be a useful method of enforcing data rules in SOA Gateway in a fast and efficient manner. For more information about the structure, rules and possibilities refer to the W3C XML Schema Specification

There is a one-to-one relationship between the DataView and the XML Schema. For example, if a personnel_id field is present in the DataView, the XML Schema can be used to enforce rules that this field must be an integer of at least, and not more than, 8 digits.

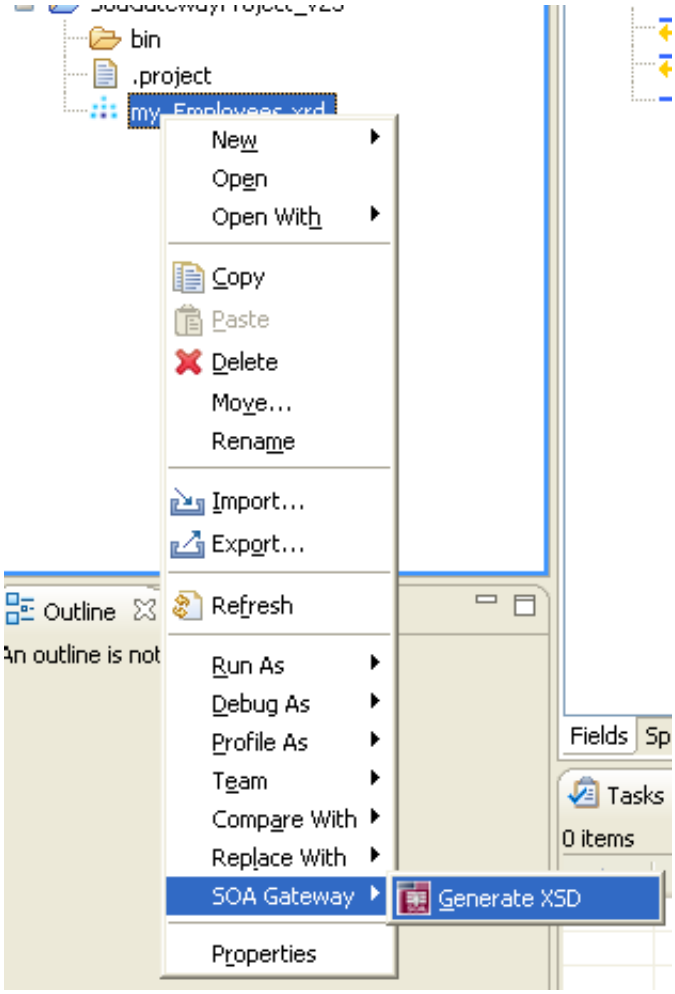
A XML Schema for a DataView can be created

- Automagically when the "Automatic (re)generation of XSD when XRD changes on the file system" preference is enabled. This option takes effect for BOTH local and remote edition of DataViews. See Window -> Preferences -> SOA Gateway -> Global Defaults.



- By right-clicking on the DataView in the Package Explorer.

  Select **SOA Gateway** and then **Generate XSD**



An XML Schema will be created at the same level as the DataView.

Important:
The filenames of the DataView and XML Schema must be identical, the only difference being the file extenstion (xrd versus xsd)

You may now export this XML Schema and/or the DataView to the server.