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

- 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.

€	
Portus DataView File This wizard creates a new file with *.xrd extension that can be opened by the Portus DataView (XRD) editor.	
Container: //test Eile name: ∫my_employees.xrd	Browse
	Cancel

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"

Name		DataView	-XSD-	-XSL-
adabas_photoblob	s	\checkmark	\checkmark	
adabas_vehides_v	/iew	\checkmark	\checkmark	\checkmark
my_employees	🧤 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:

🕥 my_employees.xrd 🛛 🗖 🗖	विन-Naviga	tor 🔲 Properties	s 🕱 📮 Console
my_employees.xrd	Root an	nd Group elem	ent names
newbataviewkoot / newbataview	Root		
		Root element	newDataViewRoot
		Group element	newDataView
Fields Special fields (0)			

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.

🔿 *my_employees.xrd 🛛 🗖 🗖	Pa- Navig	gator 🔲 Propertie	s 🛛 📮 Console
my_employees.xrd	Root a	and Group elem	ent names
AdabasEmployees / Employee			
	Root		
		Root element	AdabasEmployees
		Group element	Employee
Fields Special fields (0)			

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:

*new							
Field		_			-		
	XML name	personnel_ic	1		int. name	AA	
	Length	8	max Occurs	1	direction	in/out	~
	internal type	string	~		key Type	primary	×
	external type	default	×		ext. modifier		
	mime Type	1			mime in Field		

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

Field							
T ILLIG	XML name	first_name			int. name	AC	
	Length	20	max Occurs	1	direction	in/out	>
	internal type	string	×	(key Type	none	~
	external type	default	¥.		ext. modifier		
	mime Type	1			mime in Field		
	fixed Value						

	-							
Field		-			-12	-		
	XML name	name			int. name	AE		
	Length	20	max Occurs	1	direction	in/out	~	
	internal type	string	~		key Type	secondary	*	
	external type	default	~		ext. modifier			
	mime Type				mime in Field			
	fixed Value							_
-itur		ų. 						_
city								
city Field	XML name	dty			int. name	[A]		
city Field	XML name Length	dty 20	max Occurs	1	int. name direction	AJ	×	
city Field	XML name Length internal type	city 20 string	max Occurs	1	int. name direction key Type	AJ in/out secondary	~	
city Field	XML name Length internal type external type	city 20 string default	max Occurs	1	int. name direction key Type ext. modifier	AJ in/out secondary	× ×	
city Field	XML name Length internal type external type mime Type	dty 20 string default	max Occurs	1	int. name direction key Type ext. modifier mime in Field	AJ in/out secondary	*	

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

Field							
	XML name	address_line	e		int. name	AI	
	Length	20	max Occurs	4	direction	in/out	~
	internal type	string	*	1	key Type	none	~
	external type	default	~		ext. modifier		
	mime Type				mime in Field		
	fixed Value	1					

- 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.

my_em	ployees.xrd	incom	e							
8	AdabasEmployees / Employee AdabasEmployees / Employee personnel_id first_name adabase and adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adabase adaba	Field	XML name Length	income 0	max Occurs	1	int. name direction	AQ in/out	~	
	address_line		internal type	string	×		key Type	none	*	
	Manager and Andrews		external type	default	*		ext. modifier			
			mime Type				mime in Field			٧
			fixed Value							

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:

my_employees.xrd	curre	ncy_code						
AdabasEmployees / Employee	Field	1						
frst_name		XML name	currency_cod	le		int. name	AR	
t dty		Length	3	max Occurs	1	direction	in/out	~
address_line		internal type	string	~		key Type	none	~
arrency_code		external type	default	~		ext. modifier		
		mime Type				mime in Field		
		fixed Value					-11	
AdabasEmployees / Employee	annua	l_salary						
AdabasEmployees / Employee	Field	1						
frst_name		XML name	annual_salary	Y		int. name	AS	
ridine dity		Length	5	max Occurs	1	direction	in/out	~
- 7 addrage lina								
		internal type	packed decima	al 🗡		key Type	none	*
B → income income		internal type external type	packed decimi	a 💌		key Type ext. modifier	none	×
arrency_code → arrency_code → arrency_code		internal type external type mime Type	packed decim default	al 💉		key Type ext. modifier mime in Field	none	

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

my_employees.xrd	annual bonus					
AdabasEmployees / Employees personnel_id first_name name ct oty address_line currency_code	Field XML name Length internal type external type	annual_bonus 5 max Occur packed decimal e default	s 8	int. name direction key Type ext. modifier	AT in/out none	v v
annual_salary	mime Type fixed Value			mime in Field		×

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

File Edit Navigate Search Project
i 📬 • 🔛 🗁 i 💁 • i 🛷 i
🔿 *my_employees.xrd 🛿 🦳 🗖
my_enployees.xrd
AdabasEmployees / Employee
Fields Special fields (0)

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

my_employees.xrd		
Name	Туре	
Fields Special fields (0)		
ricids opecial licids (0)		

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

my_en	nployees.xrd	
Nam	e	
	💸 add special field	
	🔊 remove 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"

my_employees.xrd		dent nerso				
Name	Туре	uchc bergo				
dept_person	superDescriptor	Special field				
			Field type	superDescriptor	~	
			Xml Name	dept_person		Delete subfield
			internal Name	S2		Add subfield
			Referenced	field	Offset	Length
			-			

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

dept_perso	วท		
Field type	superDescriptor	•	
Xml Name	dept_person		Delete subfield
internal Name	52		Add subfield
Referenced fi	eld	Offset	Length
?		?	?
1			

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

dept_perso	าก		
Field type	superDescriptor	•	
Xml Name	dept_person		Delete subfield
internal Name	52		Add subfield
Referenced fi	eld	Offset	Length
dept	•	?	?
name city address_line income dent	▲ ▼		
dept	•	l	

Initial "Offset" and "Length" values will be derived 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

dept_perso	on		
Field type	superDescriptor		
Xml Name	dept_person	Delete subfield	
internal Name	52		Add subfield
Referenced fi	eld	Offset	Length
dept		0	6
name		0	20

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

🔲 Prop	perties 🛛			
Mont	h			
Field	XML Name	Month	Int. Name	Month
	Length	10 Max Occurs 1	Direction	input/output 🔹
	Internal Type	string 🔹	Кеу Туре	none
	External Type	default	Ext. Modifier	
	Mime Type		Mime in Field	±
	Fixed Value			
	Format	substitution 🔹	Format Mask	Jan=1,Feb=2,Mar=3,Apr=4,May=5,Jun=
	Scope	Native Attribute	Value	Add
				<u>R</u> emove

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.

т.	Resource	DataSource Id	DataView
A	adabas_my_employees	Dbid=6, Fnr=11	my_employees
A	adabas_p 🔊 add Resource	90	adabas_photoblobs_h
A	adabas_p 🛅 duplicate Resource	90	adabas_photoblobs
A	adabas_q 🐞 remove Resource	11	qe_adabas_employee
A	adabas_Q	nr=20009	qe_adabas_employee
A	adabas_v export DataView to	server 12	adabas_vehicles_viev
0	adabas-d	da, Table=city	adabasd city

- 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"

Servic	es 🚺	DataViews / XSDs / XSLs BusinessDataViews				
Nam	ne		XF	۶D	XSD	XSL
	Ŀ.	Edit DataView			 ✓ ✓ 	
		Copy Service Definition(s) to other server			✓	
	×>	Delete Service Definition file(s) from server			✓ ✓	
	1	Import Service Definition(s) from server			✓	\checkmark
	d-h-	se nhatablahe			1	

2. Select the destination folder and click "Import"

Select Service Defintion element(s) to be imported					
Destination Folder	/Temp		Browse		
🗸 DataView 🗸 XSD	XSL	my_employees			
			Import Cancel		

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.

Preferences		
type filter text	Global Defaults	↔ + ⊕ + +
 General Ant Data Management Dynamic Languages Help Install/Update Java Java EE Java Persistence JavaScript PHP Plug-in Development Portus Global Defaults UDDI Defaults Server Team Validation Web Web Services XML 	Portus Preferences Force Portus Perspective active during startup No Yes Legacy Automatic (re)generation of XSD when XRD changes Coloured Services in configuration view Always show _version in WSDL Link Restore	nges on the file system
?	0	K Cancel

• 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 extension (xrd versus xsd)

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