Using the XML Mapping Editor

This chapter covers the following topics:

- Starting the XML Mapping Editor
- The XML Mapping Editor Pages
- XML Schema Export
- XML Mapping Editor Settings
- Removing Unused Namespaces

See also Using the XML Tester.

Starting the XML Mapping Editor

To start the XML Mapping Editor

In the EntireX Workbench

• Double click on an XMM file.

Or:

Select an IDL or XMM file and choose **Open With > EntireX XML Mapping Editor**.

If the XMM file for the selected IDL file exists, the XML Mapping Editor loads this. The **Generate** button is disabled because a mapping exists.

💼 example.xmm 🕱	🖓 🗖 📴 Outline 🛛 💝 🕀 🖨	
Overview	A A CALC	
▼ General Information	Operation (A1) In	
This section describes general information about this mapping:	P Operand1 (I4) In	
Software AG IDL File: C\Demo\Demo\example.idl	Operand2 (I4) In	
Last modification: 2014-10-06 08:53:08	P SOLARS) RESULT
XMM Mapping File: C:\Demo\Demo\example.xmm	SQUARE P Operand (11) In	
Last modification: 2014-10-06 09:00:41	♦ Operand (4) III ♦ P Result (14) Out	
✓ Mapping Parameters		
Subset of available Mapping Parameters. To get the full list, follow the Link below or choose Parameter page directly.	se the Mapping	
Namespace URI: urn:com-softwareag-entirex-rpc:%I-%p		
Enable Null Value Suppression	F 5	
Mapping Parameters Get the full list of Mapping Parameters		
Set current null value suppression setting to all mappings: Update Null Value Suppression	n	
▼ Mapping		
Choose a mapping style for all programs and directions. Existing mappings will be overwritt	tten!	
Element	Generate	
▼ Testing		
Test this Mapping by launching the XML Tester:		
P Quick Test Launch the XML Tester.		
HTTP Test Requires a running XML/SOAP Listener.		
▼ <u>V</u> alidation		
Make all the available Validity Checks in all Programs: Element Names, Attribute Names, val Node Properties, the XML Node Mapping, the encoding when using unicode data types, loc Namespace Prefix definitions used in xsi:type attributes and inform about unmapped IDL Pa	alidate the XML bok for existing arts.	
2 <u>Validate</u> Start the Validation		
Overview XML Request XML Response Mapping Parameters XML Samples		

If no XMM file exists for the IDL file, the mapping style is set to "SOAP" as default value. Choose the required mapping style and press **Generate**.

Overview	A AL EXAMPLE
- Concellatorestica	
This section describes general information about this manning	P Operand1 (I4) In
C-fe and AC IDLET - CAR - AD -	Operand2 (I4) In
Software AG JDL File: C:\Demo\Demo\example.idi	Function_Result (I4) RESU
Last modification: 2014-10-06 08:55:08	A 🖓 SQUARE
XMM Mapping File: /Demo/example.xmm	Operand (I4) In
Last modification:	💞 Result (I4) Out
 Mapping Parameters 	
Subset of available Mapping Parameters. To get the full list, follow the Link below or Parameter page directly.	choose the Mapping
Namespace URI: urn:com-softwareag-entirex-rpc:%l-%p	
Enable Null Value Suppression	II.
Mapping Parameters Get the full list of Mapping Parameters	
Set current null value suppression setting to all mappings: Update Null Value Suppr	ession
▼ Mapping	
Choose a mapping style for all programs and directions. Existing mappings will be ov	erwritten!
© Element © Attribute © SOAP © Customized	Generate
▼ <u>I</u> esting	
Test this Mapping by launching the XML Tester:	
P Quick Test Launch the XML Tester.	
HTTP Test Requires a running XML/SOAP Listener.	
▼ Validation	
Make all the available Validity Checks in all Programs: Element Names, Attribute Nam Node Properties, the XML Node Mapping, the encoding when using unicode data ty Namespace Prefix definitions used in xsi:type attributes and inform about unmapped	ies, validate the XML ses, look for existing IDL Parts.
A Validate Start the Validation	
Kal Agunare proving Agungringu	-
Overview XMI Request XMI Response Mapping Parameters XMI Samples	1557

> To close the XML Mapping Editor

• From the **File** menu, choose **Close**.

Or:

Click the **Close** icon in the editor title bar.

If the current XML mapping has not yet been saved, a dialog box appears from which the close operation can be stopped or the IDL-XML mapping can be saved. An XML mapping that is not clean is indicated with a leading asterisk in the file name, displayed in the editor title.

The XML Mapping Editor Pages

- Overview Page
- XML Request Page
- XML Response Page
- Mapping Parameters Page
- XML Sample Documents Page

Overview Page

The **Overview Page** is the central page and provides sections for file information, the most important mapping parameters, mapping generation, testing and validation.

💼 *example.xmm 🔀				- 0
Overview				
▼ General Informa	tion			
This section describ	es general informati	on about this mappin	g:	
Software AG IDL File	e: C:\Demo\Demo\	example.idl		
Last modification:	2014-10-06 08:53:	08		
XMM Mapping File	: /Demo/example.	xmm		
Last modification:				
• Mapping Parame	ters			
Subset of available I Parameter page dire	Mapping Parameters ectly.	a. To get the full list, fo	ollow the Link below or choose	the Mapping
Namespace <u>U</u> RI:	urn:com-softwareag	-entirex-rpc:%I-%p		
🔽 Enable Null Valu	e Suppression			E
Mapping Paran	neters Get the full lis	st of Mapping Parame	ters	
Cat aurorat auriliurale			ladata Noll Value Companyian	8
Set current nuil valt	te suppression settin	ig to all mappings: Lo	pdate will value suppression	0
▼ Mapping				
Choose a mapping	style for al <mark>l</mark> program	s and directions. Existi	ng mappings <mark>will be overwritt</mark> e	en!
💮 <u>E</u> lement	<u> Attribute </u>	SOAP	© <u>C</u> ustomized	Generate
▼ <u>T</u> esting				
Test this Mapping b	y launching the XM	L Tester:		
🥠 Quick Test Lau	inch the XML Tester.			
AP HTTP Test Red	quires a running XML	./SOAP Listener.		
▼ Validation				
Make all the availab Node Properties, the Namespace Prefix d	le Validity Checks in e XML Node Mappin lefinitions used in xsi	all Programs: Elemen ig, the encoding wher type attributes and in	t Names, Attribute Names, vali n using unicode data types, loo form about unmapped IDL Par	date the XML k for existing rts.
💈 <u>Validate</u> Start t	he Validation			

These sections are:

• General Information

This section describes the names and locations of the selected XML mapping (XMM) file and the related IDL file.

• Mapping Parameters

Provides the **Namespace** and **Null Value Suppression** controls and contains a link to the **Mapping Parameters Page**.

Note:

Press **Update Null Value Suppression** to apply the mapping parameters (see *Mapping Parameters Page*) to a generated mapping.

- **Mapping** Starts the mapping generation for all programs.
 - Note:

Existing mappings will be overwritten!

- **Testing** Allows you to launch the XML Tester in Quick mode or HTTP mode.
- Validation

Performs validity checks in all programs and reports the result in a new dialog. Detected problems will be displaced in the **Problems** view.

XML Request Page

This page contains the **Mapping** tree of the XML Request, which is linked together by mapping paths with the IDL tree in the **Outline** view.



The XML Request shows the XML tree loaded or created so far for the selected IDL program. The XML tree can be modified in various ways. See *IDL to XML Mapping with the XML Mapping Editor*.

The **Outline** view shows the IDL tree of the currently selected IDL file. The IDL tree display has the same functionality as the IDL Editor's tree display; however, due to the fact that every IDL program must be separately mapped, only one of the IDL program tree nodes can be expanded.

Use the right mouse button to display the context menu. Double-click on a tree node to display detailed information on that node. The following keyboard options are also available in the XML request mapping area:

Key	Description
A or SHIFT+INS	Create new parameter node after currently selected one.
D or DELETE	Delete the selected IDL tree node.
F	Find IDL node with given name pattern.
I or ALT+INS	Create a new group parameter node.
L	Create new IDL library node.
М	Find mapped XML node for selected IDL tree node.
N or INS	Create new parameter node.
O or CTRL+INS	Create new RPC program node.
P or ALT+ENTER	Open property dialog on selected IDL tree node.
R	Rename the selected IDL tree node.
F3	Follow-up of M, find next mapped XML node for selected IDL tree node.

XML Node Properties Dialog

The XML Node Properties dialog contains the XML node details, the bottom panel contains namespace settings.

🖨 XML Mapp	oing Edi	tor		
XML Node Properties Properties of Operation [idl:CALC / Operation]				
Element <u>N</u> ame:		Operation		
Mapped to:		CALC / O	peration	
Program Mappin	g;	No		×
Eormat:		string		*
Value Lengt <u>h</u> :		1		
Defau <u>l</u> t Value:				
Namespace Pref	ix:			
Min. <u>O</u> ccurrence	:	1		
Ma <u>x</u> , Occurrenci	e:	1		
Null Value <u>S</u> uppr	ession:	No Suppression		~
Nilla <u>b</u> le:				
Default <u>U</u> RI:				
Table of defined Namespace definitions:				
Prefix	Namesp	ace	Default	Insert
				<u>E</u> dit
				Remove
				Default
0			ОК	Cancel

> To open the XML Node Properties Dialog

• Double-click on an XML node in the tree display.

Or:

Choose the Properties menu item in the context menus.

Or:

Select an XML tree node and press Enter.

The XML Node Properties can be modified. The node details panel consists of two subpanels.

The upper panel contains the following:

Item	Description		
Element or attribute name	Name of the element or attribute.		
Mapped to	The IDL mapping link i information.	n the full path name format. For error (fault) trees, context	
Program	Combo Box		
Mapping	No No Program Mapping		
	Element value	The value of this node identifies the program	
	Element name	The name of this node identifies the program	
Format	Format	IDL	
	string	A, AV, K, KV	
	-	G	
	I1 (deprecated)	I1	
	I2 (deprecated)	I2	
	I4 (deprecated)	I4	
	I8 (deprecated)	I4	
	integer	11, 12, 14	
	date	D	
	time (deprecated)	Т	
	dateTime	Т	
	float	F4, F8	
	Boolean	L	
	binary	B, BV	
	number	N, P, NU, PU	
	For Tamino/special nodes: Describes the data type of the Tamino node (for example, integer, string).		

Item	Description
Value Length	The length parameter contains the length or precision of the data type, i.e.
	• length of string for alphanumeric
	• length of field in bytes for binary
	• length of string representation for floating point
	• length of string representation for integers
	• length of string for Kanji
	• length of bit vector for logical (Boolean)
	Length V is legal for string types. It denotes variable-length strings, e.g. alphanumeric or Kanji.
	For Tamino/special nodes: Length of the storage reserved for the data must match format.
Default Value	Must match the data type of the node, e.g. do not use alphanumeric defaults for integer nodes. Defaults that cannot be interpreted by the runtime component are ignored. See also <i>Assigning Default Values</i> .
Namespace Prefix	Contains the namespace prefix to the tag name.
Min. Occurrence	The minimum and maximum occurrence properties are numeric values greater than or equal to zero. They describe the number of allowed occurrences of the node in the incoming XML, and the number of occurrences to be generated in the outgoing XML. The minimum occurrences value must be less than or equal to the maximum occurrences value. Incorrect settings will be found in the validity check Validate XML node properties. If Min. Occurrence is greater than 0 for incoming documents, the corresponding attribute or element is required, i.e. must be set.
Max. Occurrence	See Min. Occurrence.

Using the XML Mapping Editor

Item	Description				
Null Value Suppression (NVS)	Specify suppression of null values (NVS), that is, whether empty elements or attributes may be omitted in the outgoing (generated) XML documents. Mainly determined by Min.Occurrences and Max. Occurrences as follows:				
	• For $\min = 0$, $\max = 0$, there is no NVS.				
	• For $\min = 0$, $\max = 1$, the element / attribute may be omitted if empty.				
	• For min = 0, max > 1, you can omit all empty elements or just omit empty elements at the end of the sequence (trimming).				
	• For min =	1, $\max = 1$, there is no NVS.			
	• For min = elements a	1, max > 1, you can omit all empty elements or just omit empty t the end of the sequence (trimming).			
	• For 0 < mi at the end	In < max, you can omit all empty elements or just omit empty elements of the sequence (trimming).			
	It is also possible to suppress an attribute (independent of its value) if the associated element has a null value and null value suppression is enabled; to do this choose Depends on element in property Null Value Suppression .				
	For the data typ	bes, the following null values are defined:			
	Format	NVS Default			
	string	" " (empty string)			
	-	" " (empty string)			
	I1 (deprecated)	0			
	I2 (deprecated)	0			
	I4 (deprecated)	0			
	integer	0			
	date				
	time (deprecated)				
	dateTime				
	float	0.0			
	Boolean	false			
	binary	Only BV with length =0; Only binaries corresponding to IDL parameter with type BV have an NVS default [element/attribute with data length=0].			
	number	0.0			
Null Value	If NVS is switc and it is decided table for NullV	hed on, this value is compared with the values of the XML document d whether the values need to be transmitted. The defaults are in the alue Suppression (NVS).			

XML Request Page

Item	Description				
Nillable	If NVS is set to "No Suppression", the Nillable option is available in the dialog. If the option is selected, an empty element is represented as an empty-element tag with attribute xsi:nil="true". This means that if an nillable group element only has elements without value and without attributes, only the enclosing group tag with attribute xsi:nil="true" is displayed.				
Time Pattern	Optiona	al for date, time (deprecated	d) and dateTime.		
	The def	fault patterns conform to th	e XSD Schema 2001 specification:		
	date: yyyy-MM-dd, for example 2003-04-15 dateTime: yyyy-MM-dd'T'HH:mm:ss, for example 2003-04-15T18:48:23				
	Value f	or user-defined time patter	n:		
	Letter	Date or Time Component	Examples		
	G	Era Designator	AD		
	у	Year	1996;96		
	М	Month in year	July; Jul; 07		
	w	Week in year	27		
	W	Week in month	2		
	D	Day in year	189		
	d	Day in month	10		
	F Day of week in mon		2		
	E	Day in week	Tuesday; Tue		
	a	AM/PM marker	PM		
	Н	Hour in day (0-23)	0		
	k	Hour in day (1-24)	24		
	K	Hour in AM/PM (0-11)	0		
	h	Hour in AM/PM (1-12)	12		
	m	Minute in hour	30		
	s	Second in minute	55		
	S	Millisecond	978		
	Z	General Time Zone	Pacific Standard Time: PST; GMT-08:00		
	Z	RFC 822 Time Zone	-0800		

Correlation of Occurrence Setting and Null Value Suppression

The value for null value suppression has a higher priority than the occurrence setting of affected elements or attributes. The following rules apply:

1. NVS="Suppress Element"/"Suppress Attribute"

Setting NVS to Suppress Element" or "Suppress Attribute" results in a value of minimum occurrence of zero, which makes it optional.

2. NVS="No Suppression"

Setting NVS to "No Suppression" means that the element/attribute is always displayed so the value of #minOccur must be 1 - otherwise the element/attribute would be optional and could be suppressed.

3. NVS for arrays

For arrays, the dependency of minimum occurrence and Null Value Suppression is more complex:

1. NVS set to "No Suppression"

```
For an array with fixed size the setting is #minOccur = #maxOccur = #array size.
```

An array with variable size will contain all non-suppressable elements, but at least #minOccur elements. Empty elements will be generated to guarantee the number of #minOccur.

2. NVS set to value other than "No Suppression"

The number of array elements is at least #minOccur (empty elements will be generate and appended if number of elements is lower than #minOccur).

The lower panel is only available for element nodes and contains namespace-related properties:

Item	Description
Namespace URI	A default Namespace (xmlns=) for this element node.
Namespace definitions (table)	A table of namespace prefix / URI assignments for this element node. The listed namespaces are defined for this element and all subelements, as described in the XML namespaces specification document. Two buttons, Add and Delete, allow you to add or delete namespace definitions, respectively.

Using the Context Menu

The context menu of the XML tree enables you to modify the XML structure. It contains menu items to insert new nodes, remove or rename nodes, remove an IDL mapping, change the XML part type to element or attribute, move the selected node to the top or bottom of the current subtree, move the selected node up or down within the subtree, or show the XML Node Properties dialog.

To open the context menu

• Select the XML tree node and click the right mouse button.

Menu Item	Shortkey	Description
New child node	Ctrl-N	Insert a child node under the selected node.
Insert before	Ctrl-B	Insert a new node before the selected node.
Insert after	Insert	Insert a new node after the selected node.
Set to Attribute	Ctrl-S	Turn an attribute into an element. Also works for multiple selected nodes.
Set to Element	Ctrl-E	Turn an element into an attribute. Also works for multiple selected nodes.
Bring to top	Ctrl-T	Move the selected node to the top of the current subtree.
Bring to bottom	Ctrl-O	Move the selected node to the bottom of the current subtree.
Move up	Ctrl-U	Move the selected node up within the subtree.
Move down	Ctrl-D	Move the selected node down within the subtree.
Cut	Ctrl-X	Cut the selected node.
Сору	Ctrl-C	Copy the selected node.
Paste	Ctrl-V,Ctrl-P	Paste a node from the clipboard (after copy or cut).
Unmap	Ctrl-M	Unmap the selected IDL and XML nodes, i.e, remove link. Also works for multiple selected nodes.
Delete	Delete/Backspace	Remove the selected node. Also works for multiple selected nodes.
Rename	Ctrl-R	Rename the selected node.
Properties	Alt-Enter	Open the XML Details Panel.

Using Drag-and-drop

You can use drag-and-drop operations within the XML tree instead of move and cut-and-paste. Select a node (even with subnodes) and drag it to another place. The dragged subtree is then inserted after, before or under the corresponding drop node (to insert it under the drop node, use the Ctrl key as a toggle).

Some drag-and-drop operations are illegal, e.g. dragging a node into one of its descendant nodes (children). This would result in a cyclic reference and is thus forbidden.

Dropping a dragged node into an attribute node will convert the attribute node to an element node. This is because attributes may not have descendant nodes.



Warning:

If you drag a subtree into or out of an array, the IDL mapping links of all nodes of that subtree will be deleted. This is because the cardinality of the node occurrence has changed, and the resulting IDL mapping is very likely to be incorrect.

Drag-and-drop from an IDL Tree Node to the XML Tree

If you drag an IDL tree node onto an XML tree node, the IDL-XML mapping link for that XML node is changed. The new XML-IDL mapping of this XML tree node is to the dragged IDL node. The IDL tree is not changed.

This is a quick way to change IDL mappings for XML parts. Note that there is no immediate check for duplicate assignment of an IDL node; however, the validity checks will detect it.

XML Response Page

This page contains the Mapping Tree of the XML Response, which is linked together by mapping paths with the IDL tree in the Outline View. The page extends the XML Request page with an additional section, Fault Document Manager, at the bottom.

📾 example.xmm 🛛	🗖 🗖 📴 Outline 🛛 🕢 🖪	
Mapping	CALC CALC CALC Coperand_1 (I4) In Coperand_2 (I4) In Coperand (I4) In POWER POWER POWER Coperand (I4) In Coperand (I4) In Coper	4) RESULT 4) RESULT

The Fault Document Manager contains the entry for the XML Fault Response Document. By selecting the entry, the **Edit...** button will be enabled. Press **Edit...** to display the following dialog:

🖨 EntireX XML Mapping Editor	×
Fault Response	
Mapping © © SOAP-ENV:Header © © SOAP-ENV:Body © © SOAP-ENV:Fault [idlprog: CALC] © @ faultcode [info:faultcode] © @ faultstring [info:faultstring] © @ faultactor [info:faultactor] © @ detail [info:detail]	
OK Cancel	כ

The same operations as for the XML Request and XML Response trees are possible. Confirm the changes with **OK**, or click **Cancel** to exit without changes.

Mapping Parameters Page

This page covers the Mapping Parameters to affect the Mapping Structure. The default values for that page will be managed in the preferences and can be loaded by using the **Restore Defaults** button in the upper right corner.

apping P	arameters		Restore Default		
e mapping p st changing	parameters can only be used for generating a new mapping parameters has no effect on an existing	v IDL-XML mapping	mapping. generate the mapping again for the c	hanges to take effec	
Document	Style				
<u>Generate</u> A	Array Envelope Element				
<u>/</u> SDL Style: [document/literal				
En <u>c</u> oding S	ettings				
ML Default	incoding: UTF-8				
Use incom	ing Encoding				
Null Value	Suppression				
ontrol empty	y elements or attributes may be omitted in the XM	ML docum	ents.		
Enable N	Jull Value Suppression		Preview		
Elements			<e1></e1>		
Simple Ele	ement		<group1> <eg1></eg1> <eg2>aaa</eg2></group1>		
No Supp	ression				
Complex	Types				
Suppress	Group Elements	-	<array></array>		
Juppicss	stoup ciements		<item1></item1> <item2>two</item2>		
<u>Array</u> Iten	ពនៈ្		<item3></item3>		
Cells at E	nd (Trim)		<item4>four</item4>		
Attributes	5 M		<e1 att1="" att2="red"></e1>		
No Supp	rection	•			
Into Subh	10551017				
Namesnace	Definitions				
III CIC	IN.				
	ied Namespace definitions:		Ť		
Prefix	Namespace	Default		Insert	
	urn:com-softwareag-entirex-rpc:%l-%p	(default)		<u>E</u> dit	
SOAP-ENC	http://schemas.xmlsoap.org/soap/encodin			Remove	
xsd	http://www.w3.org/2001/XMLSchema			10000	
xsi	http://www.w3.org/2001/XMLSchema-inst			Default	
			I		

Parameter	Description
Generate Array Envelope Element	Determines whether for each array a surrounding additional element (envelope) is generated or not.
WSDL Style	Prepare the SOAP Mapping for selected WSDL Style. Possible values: document/literal or RPC/encoded.
XML Default Encoding	This encoding is used for the XML/SOAP document sent if the box Use incoming XML encoding is not checked (for XML-based clients), or if the XML/SOAP RPC Server is used.
Use Incoming Encoding	Check this box to enable the XML/SOAP Wrapper to use same encoding for the incoming document as for the outgoing document.
Enable Null Value Suppression ⁽¹⁾	Switch on/off the null value suppression. ⁽¹⁾
Simple Element ⁽¹⁾	Suppress Elements. Possible Values: No Suppression or Suppress Element. ⁽¹⁾
Simple Attribute ⁽¹⁾	Suppress Attributes. Possible Values: No Suppression or Suppress Attribute. ⁽¹⁾
Array Types ⁽¹⁾	Suppress Array Types. Possible Values: No Suppression, All empty cells or Cells at end (Trim). ⁽¹⁾
Complex Types ⁽¹⁾	Suppress Complex Types. Possible Values: No Suppression Suppression (no special handling of complex types - null value suppression defined for 'Simple Element' is used), or Suppress Group Elements. ⁽¹⁾
Namespace Definitions Table	Manage all Namespaces with prefix and URI.

⁽¹⁾ For more details on null value suppression, see below or *Null Value Suppression* in *Writing Advanced Applications with the XML/SOAP Wrapper*.

XML Sample Documents Page

This page allows the generation and modification of XML Sample Documents. They can be used to test the Mapping by sending the generated Request Document to the XML Tester. Another useful point is to compare the XML Sample Document structure with the real XML Document returned by your application to detect differences, for example Namespace Definitions, typos or anything else.

📾 example.xmm 🛛	
XML Sample Documents	
IDL	XML Documents
EXAMPLE CALC Request Document Response Document POWER HELLO	<pre><?xml version="1.0" encoding="UTF-8" ?> <!-- Generated by Software AG, EntireX XML Mapping Editor <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xr <SOAP-ENV:Header--> <!-- <SOAPAction-->CALC> <soap-env:body> <m:CALC xmlns:m="urn:com-softwareag-entirex-rpc:Ex <Operand_1>-1686024913 <operand_2>-1252560089</operand_2> </soap-env:body> </pre>
Overview XML Request XML Response Ma	apping Parameters XML Samples

All generated XML Sample Documents can be stored in parallel to the selected XMM file by using the Save All toolbar button in the upper right corner of this page or the Save command in the context menu. The generated file name is built as follows:

[xmm name] .[library name].[program name].[direction (request | response)].xml

For example: example.EXAMPLE.CALC.request.xml

If a file already exists, a dialog will prompt you for confirmation to overwrite it.

Using the Context Menu of XML Samples

The XML Samples context menu allows the generation of XML Sample Documents.

There are two different menus:

• Library or Program is selected



Generate All means both directions (Request and Response). When a Library is selected, the commands here will be inherited by all Programs.

• Request or Response Document is selected

<u>G</u>enerate Delete XML Tester... Save

The **XML Tester...** command sends the selected XML Sample Document to the XML Tester as Quick Test and enters the name of the XMM file.

XML Schema Export

The current XML Mapping can be exported as XML Schema, using the context menu of the XMM file. The XML Schema style "Russian Doll" or "Venetian Blind" can be selected in the preferences.

💾 Packag 🛛 🔋 Hierarc 🗖 🗖	
Demo Src JRE System Library [jre6] example.idl example.xmm	
New	•
<u>O</u> pen Open Wit <u>h</u>	F3
Sho <u>w</u> In	Alt+Shift+W ♪
☐ ⊆opy ☐ Copy Qualified Name	Ctrl+C
Paste	Ctrl+V
X Delete	Delete
<u>B</u> uild Path	Altushiftut 🕨
Import	
Export	
🔗 Re <u>f</u> resh	F5
Assign Working Sets	
Test Software AG XML/SOAP mapping	
Generate <u>X</u> ML Schema (XSD)	
Generate Web Service from EntireX Mapping	
Deploy to EntireX XML/SOAP RPC Server	
<u>R</u> un As	•
Debug As	•
Team	•
Comp <u>a</u> re With	•
Rep <u>l</u> ace With	•
Properties	Alt+Enter

The generated XML Schema files will be stored in parallel to the selected XMM file, where one file represents the direction (request or response) and the defined prefixed element.

For example: a SOAP mapping creates four files from the example CALC program: two prefixes ("SOAP-ENV" and "m"), and two directions (request and response). The generated file name is built as follows.

```
[xmm name].[library name].[program name].[direction (request |
response)][(optional).additional prefix].xsd
```

XML Schema Export

Example

- example.EXAMPLE.CALC.request.xsd
- example.EXAMPLE.CALC.request.m.xsd
- example.EXAMPLE.CALC.response.xsd
- example.EXAMPLE.CALC.response.m.xsd

If a file already exists, a dialog will prompt you for confirmation to overwrite it.

XML Mapping Editor Settings

The preference page XML Mapping Editor manages the default values.

For XML Schema the style "Russian Doll" or "Venetian Blind" can be selected.

All other parameters are described under Mapping Parameters.

Preferences	
type filter text	XML Mapping Editor 🛛 🗇 🗘 🗠
 General Ant Connectivity Help Install/Update Internet Java JPA Plug-in Development Report Design Run/Debug Server Software AG EntireX COBOL Wrapper COBOL Wrapper DCOM Wrapper DCOM Wrapper DElb Wrapper DCOM Wrapper DElb Wrapper DL Extractor for COBOL IDL Extractor for PL/I Installation Java Wrapper PL/I Wrapper PL/I Wrapper PL/I Wrapper PDDI Registries Web Services Stack SQL Development Web and XML Web Services XDoclet 	Store all general XML Mapping Settings and various Export Settings for XML Schema. Document Setting Namespaces XML Schema Export XML Schema Style P gussian Dol Yenetian Blind
0	OK Cancel

Removing Unused Namespaces

- > To remove one or more unused namespaces
 - 1. Open the XML Mapping Editor for the XMM file.
 - 2. Select tab XML Request.

📾 example.xmm 🛛	- 0
XML Request	
Mapping	
<pre> SOAP-ENV:Envelope SOAP-ENV:Header SOAPAction [idlprog: CALC] SOAP-ENV:Body Gree m:CALC [idlprog: CALC] Gree m:CALC [idlprog: CALC] Gree Operation [idl:CALC / Operation] Gree Operand1 [idl:CALC / Operand1] Gree Operand2 [idl:CALC / Operand2] </pre>	
Overview XML Request XML Response Mapping Parameters XML Samples	

- 3. Select the element that defines namespace(s) to be removed.
- 4. From the context menu, choose **Properties**.

🖨 XML Mapp	ing Edi	tor			
XML Node Properties Image: Solar properties of SOAP-ENV:Envelope					
Element <u>N</u> ame:		Envelope			
Mapped to:		<none></none>			
Program Mappin(];	No		~	
<u>F</u> ormat:		-		~	
Value Lengt <u>h</u> ;		V (variable)			
Defau <u>l</u> t Value:					
Namespace Prefi	ix:	SOAP-ENV			
Min. <u>O</u> ccurrence	:	1			
Ma <u>x</u> , Occurrence	e:	1			
Null Value <u>S</u> uppression:		No Suppression			
Nilla <u>b</u> le:					
Default <u>U</u> RI:					
Table of defined	Namespa	ce definitions:			
Prefix	Namesp	ace	Default	Insert	
SOAP-ENV xmm xsd SOAP-ENC	http://sc http://n/ http://w	hemas.xmlsoap.org/soap/envelope/ amespace.softwareag.com/entirex/xml ww.w3.org/2001/XMLSchema hemas.xmlsoap.org/coap/encoding/		Edit	
xsi	http://w	ww.w3.org/2001/XMLSchema-instance		Default	
?			ок	Cancel	

- 5. Select (unused) namespace entry/entries in list and choose **Remove**.
- 6. Press OK.
- 7. Repeat these steps for the XML response (select the **XML Response** tab and repeat the steps above).
- 8. Save the XMM file.
- 9. Switch to XML Samples tab, which shows the following:

🖨 XML Mapping Editor 🛛 🔀				
XML Node Properties Image: Comparison of SOAP-ENV:Envelope				
Element <u>N</u> ame:		Envelope		
Mapped to:		<none></none>		
Program Mappin	g;	No		~
<u>F</u> ormat:		-		~
Value Lengt <u>h</u> ;		V (variable)		
Defau <u>l</u> t Value;				
Namespace Pref	ix:	SOAP-ENV		
Min. <u>O</u> ccurrence		1		
Ma <u>x</u> , Occurrenc	e:	1		
Null Value Suppression:		No Suppression		~
Nilla <u>b</u> le:				
Default <u>U</u> RI:				
Table of defined	l Namespa	ce definitions:		
Prefix	Namesp	ace	Default	Insert
SOAP-ENV xmm xsd	http://sc http://na http://w	hemas.xmlsoap.org/soap/envelope/ mespace.softwareag.com/entirex/xml ww.w3.org/2001/XMLSchema		<u>E</u> dit
SOAP-ENC xsi	http://so http://w	hemas.xmlsoap.org/soap/encoding/ ww.w3.org/2001/XMLSchema-instance		Rem <u>o</u> ve
0			ок	Cancel

10. The XMM file defines the following:

```
...
<FromXml>
<Method relatedIdlLibrary="EXAMPLE" relatedIdlProgram="CALC" encoding="UTF-8"
useIncomingEncoding="true">
<XmlNode name="Envelope" length="0" min="1" max="1" namespacePrefix="SOAP-ENV"
nullSuppression="NVS_NONE" nullValue="" >
<XmlNamespaceDef prefix="SOAP-ENV" uri="http://schemas.xmlsoap.org/soap/envelope/"/>
<XmlNamespaceDef prefix="SOAP-ENC" uri="http://schemas.xmlsoap.org/soap/encoding/"/>
<XmlNode name="Header" length="0" min="1" max="1" namespacePrefix="SOAP-ENV"
nullSuppression="NVS_NONE" nullValue="" >
<XmlNode name="Header" length="0" min="1" max="1" namespacePrefix="SOAP-ENV"
nullSuppression="NVS_NONE" nullValue="" >
</XmlNode name="Header" length="0" min="1" max="1" namespacePrefix="SOAP-ENV"
nullSuppression="NVS_NONE" nullValue="" >
</XmlNode name="SOAPAction" format="string" type="xsd:string" length="0" min="1" max="1"
default="CALC" nullSuppression="NVS_NONE" nullValue=" programNode="ev" >
</XmlNode>
...
```