

IDL to XML Mapping with the XML Mapping Editor

The EntireX XML Mapping Editor allows you to map XML document structures to IDL libraries, programs and parameters. The mappings can be defined for the request and response to the server application, or from the server to the client. The input for the XML Mapping Editor can be a Software AG IDL file and/or an IDL-XML mapping file (perhaps produced by a previous XML Mapping Editor session or by importing a WSDL file, XML Document or XML Schema). The output is an IDL-XML mapping file, other XML structure definitions (such as sample XML files), and perhaps a created or changed IDL file.

There are two ways to create an IDL to XML mapping: automatically (and then modify the resulting structures manually), or manually.

This chapter covers the following topics:

- Mapping IDL to XML Automatically
 - Mapping IDL to XML Manually
 - Defining the XML Encoding
 - Validity Checks
 - Exporting the IDL-XML Mapping to an XML Schema
-

Mapping IDL to XML Automatically

This section covers the following topics:

- Scope
- Mappings
- SOAPAction
- Using Mapping Parameters
- Mapping Parameters
- Document Style
- Encoding Settings
- Null Value Suppression
- Namespace Definitions

Scope

The following approaches are available:

- Map all programs in the current IDL file with the element-preferred strategy.
- Map all programs in the current IDL file with the attribute-preferred strategy.
- Create a SOAP mapping for all programs in the current IDL file.
- Mapping and XML Schema generation for all programs in the current IDL file. This creates an element-preferred mapping.

The mapping strategies generate XML tree nodes for all IDL parameters with the appropriate direction:

- For IN and INOUT IDL parameters, XML Request nodes are generated.
- For INOUT and OUT IDL parameters, XML Response nodes are generated.
- Fault trees (Error trees) are generated, see **Fault Document Manager** under *XML Response Page*.

Format, length and default values of the XML nodes are set according to the IDL parameter:

In the table below, the following metasympols and informal terms are used for the IDL.

- The metasympols "[" and "]" surround optional lexical entities.
- The informal term *number* (or in some cases *number1.number2*) is a sequence of numeric characters, for example 123.

Software AG IDL	Description	XML Data Type
<i>Anumber</i>	Alphanumeric	Variable-length <code>string</code> node with maximum length.
AV	Alphanumeric variable length	Variable-length <code>string</code> node.
AV[<i>number</i>]	Alphanumeric variable length with maximum length	Variable-length <code>string</code> node with maximum length.
<i>Bnumber</i>	Binary	<code>binary</code> node with length of binary representation (base 64)
BV	Binary variable length	Variable-length <code>binary</code> node.
BV[<i>number</i>]		Variable-length <code>binary</code> node with maximum length.
D	Date	XML <code>date</code> node. The length depends on the specified time pattern length. This is trimmed, and what is included in single quotes will be counted (without the single quotes). For example: 'T'HH:mm:ss has length=9.
F4	Floating point (small)	<code>float</code> node of length 32.

Software AG IDL	Description	XML Data Type
F8	Floating point (large)	float node of length 64.
I1	Integer (small)	integer node of length 4.
I2	Integer (small)	integer node of length 6.
I4	Integer (small)	integer node of length 11
<i>Knumber</i>	Kanji	Variable-length string node with maximum length.
KV	Kanji variable length	Variable-length string node.
KV[<i>number</i>]	Kanji variable length with maximum length	Variable-length string node with maximum length.
L	Logical	Boolean node.
N <i>number1</i> [. <i>number2</i>]	Unpacked decimal	number node. If <i>number</i> > 0, length increased by 2, otherwise by 1.
NU <i>number1</i> [. <i>number2</i>]	Unpacked decimal unsigned	number node. If <i>number</i> > 0, length increased by 2, otherwise by 1.
P <i>number1</i> [. <i>number2</i>]	Packed decimal	number node. If <i>number</i> > 0, length increased by 2, otherwise by 1.
PU <i>number1</i> [. <i>number2</i>]	Packed decimal unsigned	number node. If <i>number</i> > 0, length increased by 2, otherwise by 1.
T	Time	dateTime node. The length depends on the specified time pattern length. This is trimmed, and what is included in single quotes will be counted (without the single quotes). For example: <code>YYY-MM-dd'T'HH:mm:ss</code> has length=19.
U <i>number</i>	Unicode	Variable-length Unicode node with maximum length.
UV	Unicode variable length	Variable-length Unicode node.
UV <i>number</i>	Unicode variable length with maximum length	Variable-length Unicode node with maximum length.

The resulting XML structures can be modified, see *Mapping IDL to XML Manually*.

The automatic mapping process can be fine-tuned by mapping parameters to be set before mapping.

For information on the attribute-preferred and element-preferred strategies see *XML Structures and IDL-XML Mapping* in the XML/SOAP Wrapper documentation.

Mappings

For SOAP additional type attributes are generated. They describe the XML Schema compliant data types of the elements.

SOAPAction

The SOAPAction tag in the SOAP header will be moved to the HTTP header. The SOAPAction element is generated automatically with SOAP mapping. Verify the default setting. If SOAPAction is missing, insert it: select the SOAPHeader node and add a new child node with the name SOAPAction. Then select the new element node with the name SOAPAction, open the XML Details panel and enter the value for that HTTP header into the Default attribute field. The NullValueSuppression property should have the value "Suppress Element".

Using Mapping Parameters

> To use the mapping parameters

1. Switch to the Mapping Parameters page or define the XML Mapping Editor preferences.
2. Check the check boxes, dialog boxes and complete the text fields you need.

Mapping Parameters

The mapping parameters are on the following sections.

Document Style

Mapping Parameters Restore Defaults

The mapping parameters can only be used for generating a new IDL-XML mapping. Just changing mapping parameters has no effect on an existing mapping; generate the mapping again for the changes to take effect.

▼ **Document Style**

Generate Array Envelope Element

WSDL Style: document/literal

▶ **Encoding Settings**

▶ **Null Value Suppression**

▶ **Namespace Definitions**

Overview | XML Request | XML Response | **Mapping Parameters** | XML Samples

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.

Encoding Settings

Mapping Parameters
Restore Defaults

The mapping parameters can only be used for generating a new IDL-XML mapping. Just changing mapping parameters has no effect on an existing mapping; generate the mapping again for the changes to take effect.

▶ Document Style

▼ Encoding Settings

XML Default Encoding:

Use incoming Encoding

▶ Null Value Suppression

▶ Namespace Definitions

Overview
XML Request
XML Response
Mapping Parameters
XML Samples

Parameter	Description
XML Default Encoding	This encoding is used for the sent XML/SOAP document 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.

Null Value Suppression

Mapping Parameters

Restore Defaults

The mapping parameters can only be used for generating a new IDL-XML mapping. Just changing mapping parameters has no effect on an existing mapping; generate the mapping again for the changes to take effect.

Document Style

Encoding Settings

Null Value Suppression

Control empty elements or attributes may be omitted in the XML documents.

 Enable Null Value Suppression

Elements

Simple Element

No Suppression

Complex Types

Suppress Group Elements

Array Items

Cells at End (Trim)

Attributes

No Suppression

Preview

```
<e1 />
<group1>
  <eg1 />
  <eg2>aaa</eg2>
</group1>
<group2 />
<array>
  <item1 />
  <item2>two</item2>
  <item3 />
  <item4>four</item4>
</array>
<e1 att1="" att2="red" />
```

Namespace Definitions

Overview | XML Request | XML Response | Mapping Parameters | XML Samples

Parameter	Description
Enable null value suppression	Switch on/off the null value suppression.
Simple Element	Suppress Elements. Possible Values: No Suppression or Suppress Element.
Simple Attribute	Suppress Attributes. Possible Values: No Suppression or Suppress Attribute.
Array Types	Suppress Array Types. Possible Values: No Suppression, All empty cells or Cells at end (Trim).
Complex Types	Suppress Complex Types. Possible Values: No Suppression (no special handling of complex types - null value suppression defined for 'Simple Element' is used), or Suppress Group Elements.

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

Namespace Definitions

Mapping Parameters

Restore Defaults

The mapping parameters can only be used for generating a new IDL-XML mapping. Just changing mapping parameters has no effect on an existing mapping; generate the mapping again for the changes to take effect.

▶ Document Style

▶ Encoding Settings

▶ Null Value Suppression

▼ Namespace Definitions

Table of defined Namespace definitions:

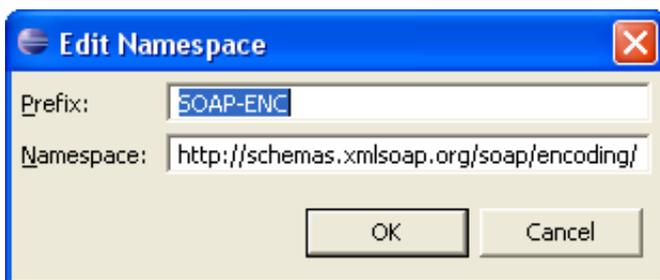
Prefix	Namespace	Default	
m	urn:com-softwareag-entirex-rpc:%l-%p	(default)	Insert...
SOAP-ENC	http://schemas.xmlsoap.org/soap/encodin...		Edit...
SOAP-ENV	http://schemas.xmlsoap.org/soap/envelop...		Remove
xsd	http://www.w3.org/2001/XMLSchema		Default
xsi	http://www.w3.org/2001/XMLSchema-inst...		

Overview | XML Request | XML Response | **Mapping Parameters** | XML Samples

Parameter	Description
Namespace Definitions Table	Manage all Namespaces with prefix and URI.

New namespace definitions can be entered with **Add...** . To change an entry choose **Edit...** . All selected entries can be deleted with the **Remove** button. With the **Default** button you mark the Namespace used for the Payload Root node, e.g. in XML Mapping the Root Element and in SOAP Mapping the first element in the SOAP-Body.

The Edit dialog (and similar Add dialog) looks like:



➤ **To map automatically with your preferred result**

- On the Mapping section of the Overview Page choose the preferred Mapping style and press the **Generate All** button.

The various generated XML Structures can be modified, see *Mapping IDL to XML Manually*.

If errors/warnings have occurred, the Validity Check window will be displayed. The errors/warnings for all programs will be found in the Problems View. In the first line of this View you will find a short summary of the problems. If you click on an error/warning, the XML structure for it is displayed with the invalid node highlighted.

**Warning:**

There are rare cases when the SOAP default mapping generates warnings (for example for IDL files with cascaded arrays). A dialog is displayed stating that illegal mappings have been generated. The warnings mostly deal with duplicate element names. You may ignore them as long as you do not use the mapping for WSDL or XML Schema. For details, see the Problem View.

Mapping IDL to XML Manually

This section covers the following topics:

- Summary of Steps
- Choosing XML Request or XML Response XML Structure
- Adding an XML Node
- Grouping XML Elements or Attributes
- Mapping an IDL Node to an XML Node
- Unmapping XML Nodes
- Deleting Nodes
- Deleting XML Structures
- Modifying the Element or Attribute Name
- Changing Elements to Attributes and Vice Versa
- Assigning Default Values

Summary of Steps

➤ **To map an IDL file manually to an XML structure or modify an IDL-XML mapping file**

1. Choose direction (XML Request or XML Response).
2. Add/modify XML nodes.
3. Map IDL nodes to XML nodes.

Choosing XML Request or XML Response XML Structure

➤ **To choose the XML Structure**

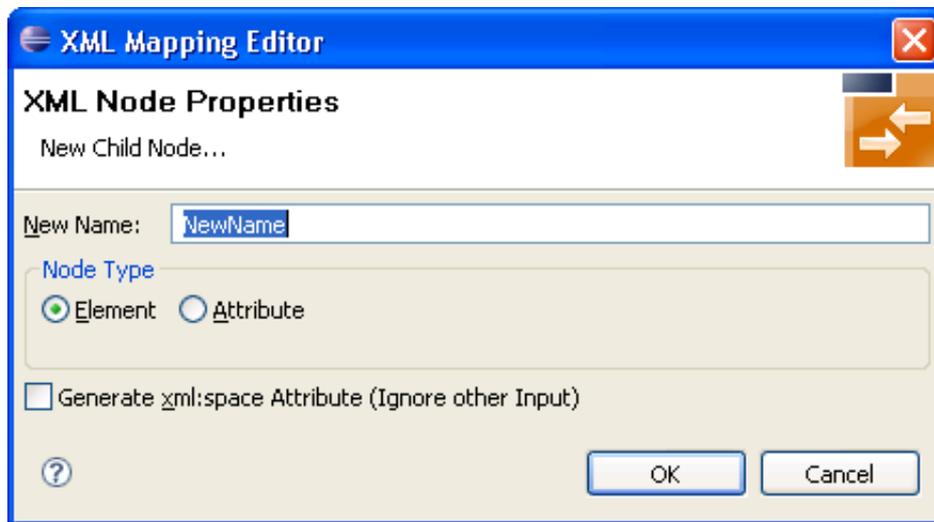
1. Select the XML Request to get the IDL IN and INOUT Parameter Mapping.
2. Select the XML Response to get the IDL OUT and INOUT Parameter Mapping.

Adding an XML Node

➤ **To create a child node to an existing (selected) XML node**

1. Click on the existing node.
2. From the context menu, choose **New child node**.

A dialog window is displayed.



3. Enter the new node's name and type (element or attribute) and use **OK**.

> To insert a new node before or after an existing (selected) XML node

1. From the context menu, choose **Insert before** or **Insert after**, respectively.

A dialog window is displayed.

2. Enter the new node's name and type (element or attribute) and use **OK**.

Note that new attribute nodes do not have a format, length and default property assigned. You must assign at least a format to the node before saving the mapping file. See *Quick Test Details Dialog*. Use the *Validity Checks* to make sure that all nodes have the necessary properties.

Grouping XML Elements or Attributes

You can introduce new elements by grouping one or more existing elements or attributes. See also *XML Structures and IDL-XML Mapping* in the XML/SOAP Wrapper documentation.

> To group XML elements or attributes

- Add a new XML part and move the existing nodes into the new node.

You can use cut-and-paste or a drag-and-drop operation to reorder the XML parts. Four move functions are available in the XML context menu: **Bring to top**, **Bring to bottom**, **Move up**, **Move down**, which are equivalent to drag-and-drop operation. See *Using the Context Menu* and *Using Drag-and-drop*.

Mapping an IDL Node to an XML Node

> To add (or modify) an IDL-XML node mapping link

1. Click into the **Mapped to** field in the *XML Node Properties Dialog*.

A dialog displays the possible IDL mapping links.

2. Choose one of the menu items to map the IDL node to the XML node.

If the XML node had a previous IDL mapping link, it is released.

Or:

Use a drag-and-drop operation to move the IDL node to be mapped directly to the XML node. If the XML node had a previous IDL mapping link, it is released. See *Using Drag-and-drop*.

Or:

Use the keyboard to tab into the Mapped to text field, and press Enter. This will open the dialog Mapped to. Select a mapping link from the list and use the **OK** button to confirm. Use the **Cancel** button to cancel the action.

IDL-XML mapping uses the full path notation. This notation is a slash-separated list of IDL node names, starting with the program name, ending with the addressed IDL node, and containing all IDL nodes between program and the IDL node (the path from the program name to the node).

> To modify the format of the XML node

- Select an entry from the dialog box in the *XML Node Properties Dialog*.

The dialog box contains the possible format values. You cannot add new format codes.

Unmapping XML Nodes

> To unmap XML nodes

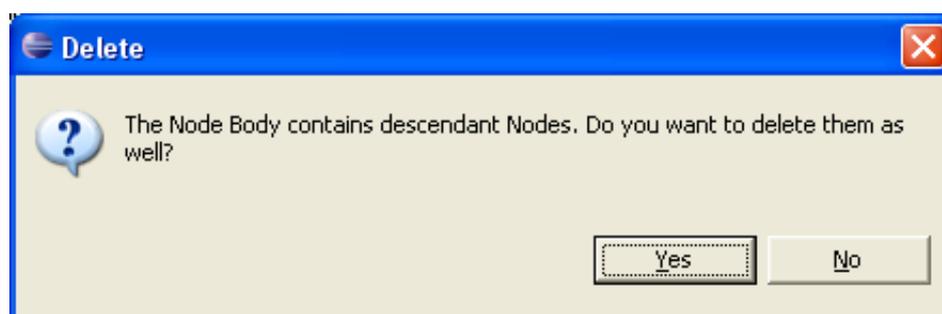
1. Select one or more XML nodes mapped to IDL node(s).
2. From the context menu, choose **Unmap**.

This will unlink the IDL and XML nodes.

Deleting Nodes

> To delete arbitrary elements or attributes

1. Select the node(s) to delete.
2. From the context menu, choose **Delete**. If nodes have descendants, the following window is displayed:



3. To delete the subtree attached to the node, choose **Yes**.
4. To keep the elements of the subtree and move them one level higher, choose **NO**.

**Warning:**

If IDL nodes are not mapped in the incoming XML request, the result for the runtime component is that the corresponding IDL parameters are filled with zero strings.

Deleting XML Structures

> To delete the complete IDL to XML mapping (all XML Structures)

- In the mapping menu, choose the item **New mapping for all programs**.

> To delete the currently selected XML Structures

- In the mapping menu, choose the item **New mapping**.

Modifying the Element or Attribute Name

In the automatically mapped structures the XML node names are built from the IDL parameter names.

> To change the node name

- Choose **Rename** from the context menu.

Or:

Double-click on the name field in the *XML Node Properties Dialog*.

To avoid accidental changes, there is no other way to modify the name.

When the name of an XML part is modified, various checks of the XML structure are performed, e.g. attribute name duplication, IDL mapping legality. The IDL-XML-mapping link is not influenced by the name change.

Changing Elements to Attributes and Vice Versa

You have the possibility to switch the node type between element and attribute. Note that attributes may not have descendant nodes and namespace definitions.

> To switch between elements and attributes

- From the context menu, choose **Set to Element** and **Set to Attribute**.

This function works for multiple nodes, too.

Important:

Arrays must always be modelled with elements.

Assigning Default Values

Every XML part can contain default values in case elements are missing in the incoming XML document.

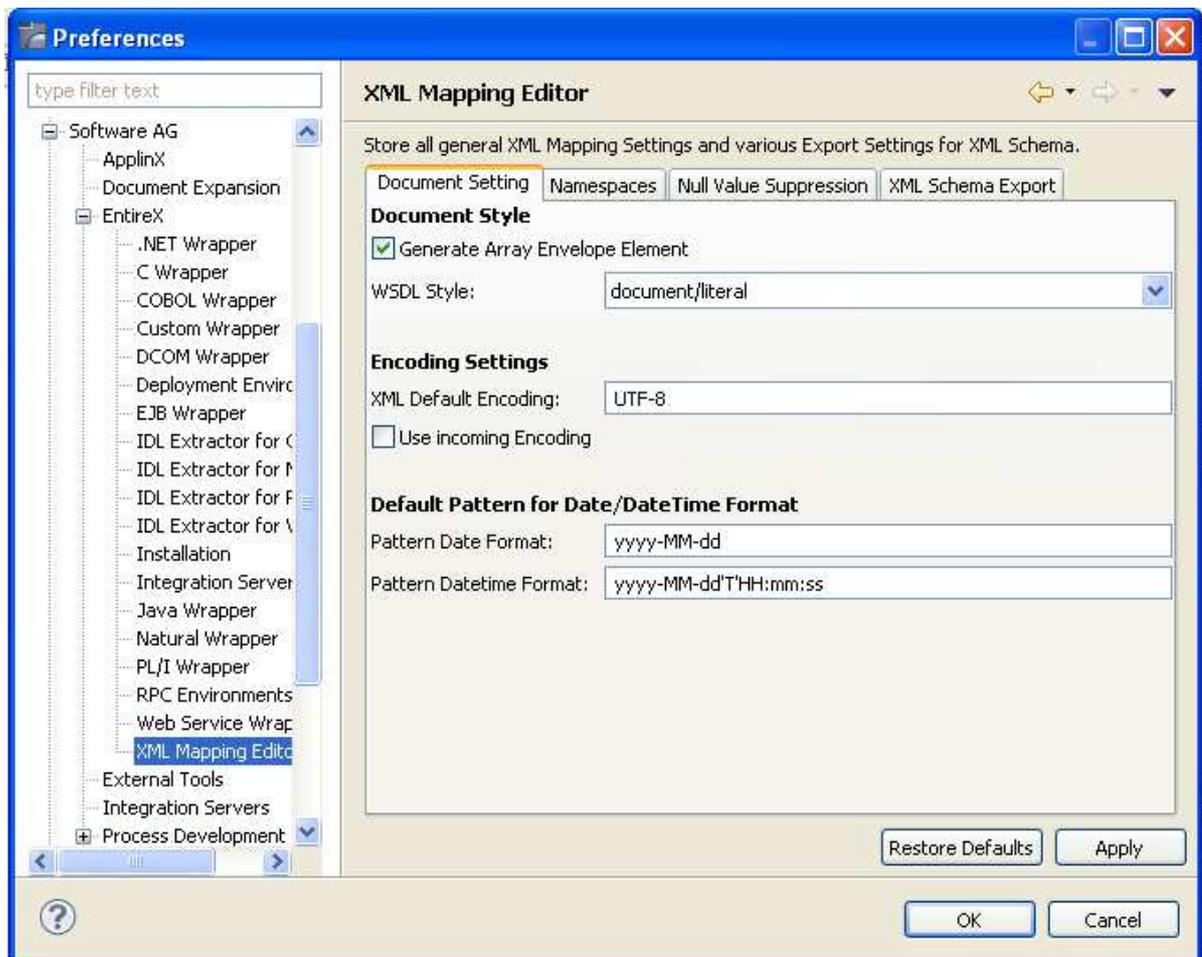
> To set the default values

- In the *XML Node Properties Dialog*, enter a **Default Value**.

Defining the XML Encoding

➤ To define default encoding for IDL to XML mapping

1. Open the IDL properties and choose the XML tab



2. Modify the Encoding Settings section:

- **XML default encoding (default: UTF-8)**

This is the encoding that will be used to write the mapping file itself and if there are no XML declarations in the incoming document. Together with the next property (Use incoming XML encoding) this encoding will be used for the response documents.

- **Use incoming XML encoding (default: checked)**

If this check box is checked, the last two properties are disabled and the same encoding of the incoming document will be used for the response documents.

Validity Checks

The Validity check is automatically performed for automatic mapping and when an existing IDL-XML mapping file is opened, but can also be performed manually.

The following validity checks are available:

1. Checks that all elements have unique tags - produces warnings
2. Checks that all attributes of an element have unique names (mandatory)
3. Checks that the generated XML parts have correct property values (mandatory), e.g. minimum occurrences less than or equal to maximum occurrences
4. Checks that the IDL mapping is valid (mandatory) and the IDL parameters are not mapped to multiple XML parts (warning)
5. Checks that all IDL parameter nodes are mapped to one or more XML parts, (so that the runtime component finds XML values for all IDL parameters or vice versa) (warning)

The first four of the checks apply to XML tree nodes, i.e. search for invalid settings or mapping rules of XML elements or attributes. The fifth check applies to the IDL tree.

The check messages may be divided into warnings (message text starts with (W)) and errors. Warnings can even be caused by automatic mapping. Generally, they can be ignored. Errors should not be ignored.

Exporting the IDL-XML Mapping to an XML Schema

➤ To save the XML structure tree as an XML Schema (XSD) document

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

See *Mapping IDL Data Types to an XML Schema (XSD)* for mapping of IDL data types to XSD.

See also *XML Schema Parser Standards Conformance* and *XML Schema Writer Standards Conformance* under *XML Schema Standards Conformance (XML/SOAP Wrapper)* in the XML/SOAP Wrapper documentation.