Using EntireX Custom Wrappers

The EntireX Custom Wrappers are user-configurable, template-based wrappers and need:

- a Software AG IDL file
- a template (for example, client or server)
- the IDL Compiler

Naturally, existing *.plugin* files from former EntireX installations, called EntireX Workbench Plug-ins, can be migrated. The Custom Wrapper definitions are stored in the current Eclipse workspace and can be managed in the preferences. Any changes in the Custom Wrapper definitions require a restart of the Workbench to take effect. This chapter covers the following topics:

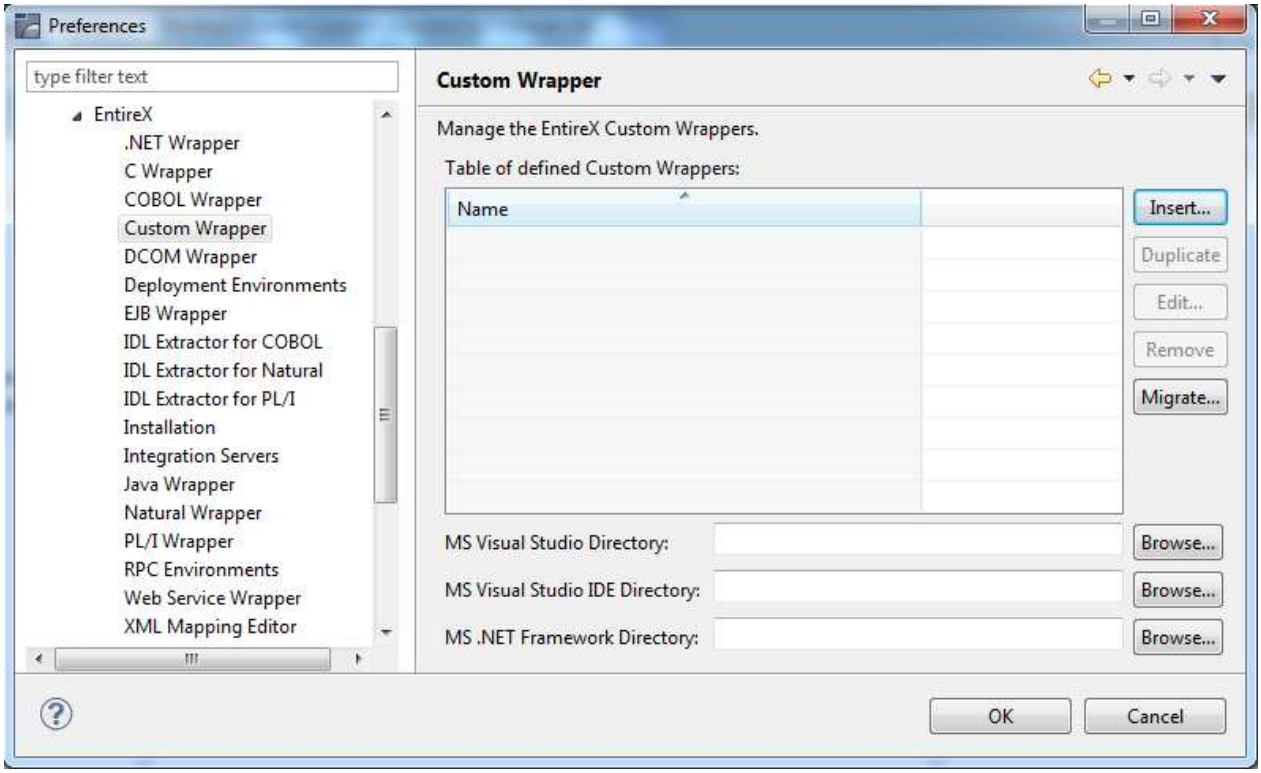
- Define EntireX Custom Wrappers
- Running a Custom Wrapper

Define EntireX Custom Wrappers

The EntireX Custom Wrappers are managed in the Custom Wrapper preferences page.

The central panel lists all known (active or inactive) Custom Wrappers, that is, all Custom Wrappers that are found in the current workspace. You can create a new empty Custom Wrapper, copy, edit or remove an existing Custom Wrapper or migrate your existing *.plugin* file from a former EntireX installation.

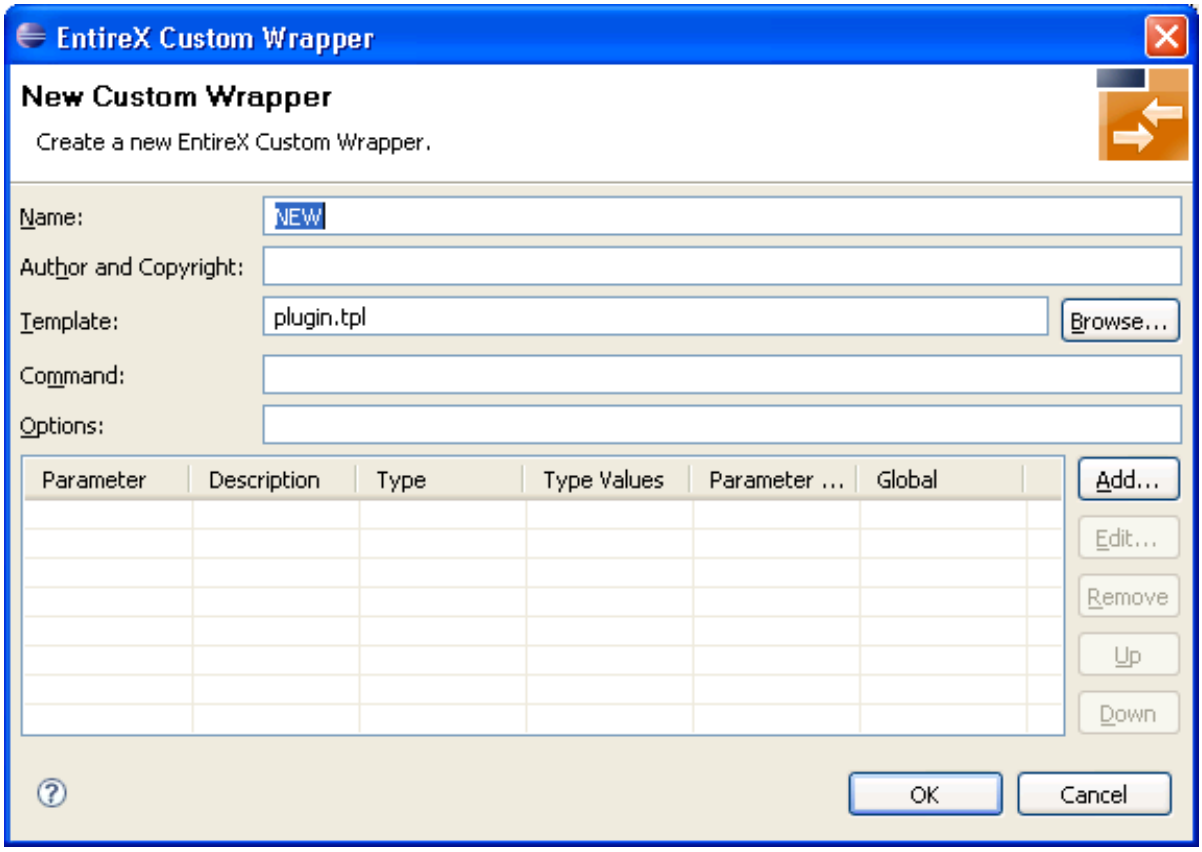
- Creating a New Custom Wrapper
- Parameters
- Parameter Values
- Wildcards



Creating a New Custom Wrapper

To create a new Custom Wrapper

1. Choose **Insert...**



2. Enter the fields. The Name is a required field and must be unique, because it is the identifier for the Custom Wrapper. This name occurs on the Custom Wrapper Properties page and will be used for the command line with the prefix minus sign.

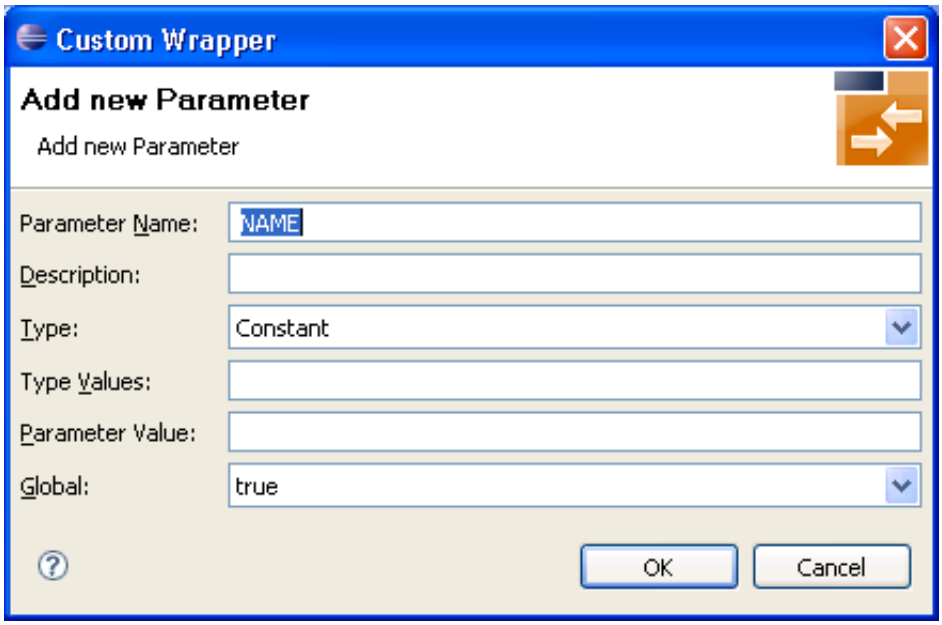
Field	Description
Name	Used as the Custom Wrapper menu item text.
Author and copyright	Important if you want to share your Custom Wrapper with other users.
Template	Name of the template file to be used by the *Software AG IDL Compiler*. Use a fully qualified file name.
Command	Executable command file name. Default is empty and means the *Software AG IDL Compiler* will be executed.
Options	Batch parameters. They are sent to the *Software AG IDL Compiler*.
Parameter	Parameters (constant or modifiable) to call the Custom Wrapper.

3. Optionally, add some parameters as described in the following sections.

To add parameters

- Choose **Add...**

A new dialog with default values is displayed. See section *Parameters* for a description of the individual fields. Modifiable parameters automatically appear with an appropriate label and GUI widget on the Custom Wrapper IDL properties tab.



To edit parameters (see *Parameters*)

1. Select the parameter to edit in the table.
2. Choose **Edit...**

   or

   Double click on the parameter in the table.

   A new dialog with the current parameter values is displayed. See section *Parameters* for a description of the individual fields. For all strings in the fields **Type Values** and **Parameter Value**, you can also use wildcards. Wildcards are like variables for actual values that may change with the platform or in other properties tabs. See the list of *Wildcards*.
3. Choose **OK** to save your entries and close the dialog or **Cancel** to close the dialog without saving.

To remove parameters

1. Select the parameters to remove in the table.
2. Choose **Remove**.

   Multiple selections are possible.

To change the parameter order

1. Select the parameter to edit in the table.

2. Use the **Up** and **Down** buttons to change the order of the parameters.

Parameters

Parameters consist of the following fields:

Field	Purpose
Parameter	The parameter identifier, as required by the *erxidl* template (must not contain whitespace characters; often in uppercase).
Description	The textual description of this parameter, as displayed in the associated Custom Wrapper IDL properties tab. Human-readable, may contain whitespace characters and short examples.
Type	Dialog box of possible GUI representations of the *Parameter Values*.
Type Values	The possible values that may be assigned to the parameter, in the form <*parametername*>=<*value*>. Values are separated by a semicolon.
Parameter Value	Default value for the GUI representation, must be one of the values listed in the **Type Values** field.
Global	Flag for the parameter value. True (default) means the parameter value is for all IDL files, false means the parameter value is IDL specific (stored in IDL properties).

Parameter Values

Parameter "types" are GUI representations of the parameter values in the IDL properties tab for this Custom Wrapper. They can be:

Entry	Purpose, Usage
Check box	Will draw a check box in the Custom Wrapper IDL **properties** tab. **Type Values** must have two values, separated by a semicolon. The first value is taken if the check box is checked, the second value is taken if the check box is cleared.
Combo box	Will draw a combo box (drop-down list) in the Custom Wrapper IDL properties tab. **Type Values** must have at least one entry, or multiple entries separated by semicolon. An entry may be: • a simple string (without the "=" character), which is displayed as an entry in the box and is sent to the Software AG IDL Compiler. • a pair "<*val*>=<*string*>", where <*string*> is displayed in the combo box, but <*val*> is sent to the Software AG IDL Compiler.
Constant	Will not generate a GUI element in the associated IDL **properties** tab. **Type Values** just has one string, which is sent to the Software AG IDL Compiler.
Text	Will show a text field in the associated Custom Wrapper IDL properties tab. The text field content is taken for the Software AG IDL Compiler command line construction. If whitespace characters are typed in the text field, the Software AG IDL Compiler command line portion is protected by double quotes. Type values may contain a string representing as default value for the text field.

Wildcards

Wildcards start with the % (percent sign). Implemented wildcards:

Wildcard Name	Purpose	Example Value
`%brokerid`	Get the Broker ID from the **General** tab.	localhost:1971
`%server`	Get the server address from the **General** tab.	RPC/SRV1/CALLNAT
`%serverclass`	Get the server class identifier from the **General** tab.	RPC
`%servername`	Get the server name from the **General** tab.	SRV1
`%service`	Get the service identifier from the **General** tab.	CALLNAT
`%idlfile`	Get the current IDL file (as absolute path name) as selected in the EntireX Workbench.	C:\Examples\example.idl
`%idlname`	Get the current IDL file (just the file name with extension, no path) as selected in the EntireX Workbench.	example.idl
`%pureidlname`	Get the current IDL file name without path and without the file extension.	example
`%idlpath`	Get the current IDL file path (without the IDL file name) as selected in the EntireX Workbench.	C:\Examples
`%xmlfile`	Get the current XML mapping file name from the XML tab.	C:\Examples\example.xmm
`%msdotnetenv`	Path of Microsoft Visual Studio environment as set in the **Tools Options** menu item.	C:\Program Files\Microsoft Visual Studio 10 \Common7\IDE\
`%netfrmdir`	Path for installation of the .NET framework.	C:\WINDOWS\Microsoft.NET\Framework\ v2.0.50727
`%version`	Get the EntireX version.	8.0
`%osname`	Get the operating system name, taken from the Java system property *os.name*.	Windows XP
`%osarch`	Get the operating system architecture, taken from the Java system property *os.arch*.	x86

Wildcard Name	Purpose	Example Value
`%osversion`	Get the operating system version, taken from the Java system property *os.version*.	5.1
`%fileseparator`	Get the platform-specific file separator character, taken from the Java system property *file.separator*.	\
`%fileencoding`	Get the platform-specific file encoding, taken from the Java system property *file.encoding*.	Cp1252
`%pathseparator`	Get the platform-specific path separator character, taken from the Java system property *path.separator*.	;
`%username`	Get the current user name, taken from the Java system property *user.name*.	administrator
`%userhome`	Get the assigned user home directory, taken from the Java system property *user.home*.	C:\Documents and Settings\administrator
`%userdir`	Get the current user directory, taken from the Java system property *user.dir*.	C:\Documents and Settings\administrator\My Documents

Duplicate an existing Custom Wrapper

1. Select the Custom Wrapper in the list.
2. Proceed as described in *Creating a New Custom Wrapper* and press the **Duplicate** button instead of the **Insert...** button.

   A new Custom Wrapper named *Copy of <name>* is displayed in the list.
3. Modify the entries as described in *Creating a New Custom Wrapper*.

Edit an existing Custom Wrapper

1. Select the Custom Wrapper in the list.
2. Proceed as described in *Creating a New Custom Wrapper* and select the **Edit...** button instead of the **Insert...** button
3. Modify the entries as described in *Creating a New Custom Wrapper*.

Remove an existing Custom Wrapper

1. Select the Custom Wrapper in the list.
2. Use the **Remove** button

Migrate an existing *.plugin* File from a previous EntireX Installation

1. Proceed as described in *Creating a New Custom Wrapper* and select the **Migrate...** button instead of the **Insert...** button.
2. Browse to the location of the *.plugin* file and select the file to migrate.
3. In the ensuing dialog, modify the entries as described in *Creating a New Custom Wrapper*.
4. Make sure the template file can be accessed correctly.

Running a Custom Wrapper

To start the Custom Wrapper in GUI

1. Select an IDL file.
2. Open the context menu, choose **Other** > **Generate Via Template** and select the desired item (for example, **New**).

When the Custom Wrapper is started, the EntireX Workbench starts the Software AG IDL Compiler and feeds it with a template, the IDL file name and (optional) parameters.

The parameters are inserted for the Software AG IDL Compiler in the form `<parametername>=<value>`. If wildcards were used for the values, they are resolved to the actual values when the Custom Wrapper was called.

To start the Custom Wrapper in Command Line

- See *Using the EntireX Workbench in Command-line Mode* for the general syntax of the command line.

The command for the Custom Wrapper is `-xxx`, where `xxx` is the case-sensitive name of the Custom Wrapper. If the name contains blanks, use `-"xxx"`.

Example:

```
-NEW /Demo/example.idl
```

The result of the Custom Wrapper is written to Standard Out.