Using the COBOL Wrapper in Command-line Mode

Commands are available to generate a COBOL RPC client or COBOL RPC server from a specified IDL file.

This chapter covers the following topics:

- Command-line Options
- Example Generating an RPC Client
- Example Generating an RPC Server
- Further Examples

See also *Command-line Mode*.

Command-line Options

This section provides the command-line options for the following tasks:

- Generate a COBOL RPC Client from IDL File
- Generate a COBOL RPC Server from IDL File

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

Generate a COBOL RPC Client from IDL File

To generate a COBOL RPC client from the specified IDL file, use the following command with options in table below:

```
-cobol:client
```

Option	Description
`-comm`	The RPC communication area. Valid values: EXTERNAL, LINKAGE, COPYBOOK. See *RPC Communication Area* for more information. EXTERNAL *`External Clause`* LINKAGE *`Linkage Section`* COPYBOOK *`Copybook`* For possible combinations with `-target` and `–interface` option, see below.
`-folder`	Folder where the COBOL files will be stored.
`-help`	Display this usage message.
`-interface`	Interface type, either DFHCOMMAREA or LINKAGE. For possible combinations with `-target` and `–comm` option, see below.
`-literal`	Enclose string literals in quotes or apostrophes. Valid values: QUOTE, APOST. See *Characters Used for String Literals* for more information.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td><code>-target</code></td><td>Target operating system and environment, one of BATCH_ZOS, BATCH_VSE, BATCH_BS2000, BATCH_I5OS, CICS_ZOS, CICS_VSE, IMS_MPP, IMS_BMP, IDMS_ZOS, MICROFOCUS_WINDOWS or MICROFOCUS_UNIX. See <i>Client Interface Types</i> for more information. For possible combinations with the <code>-interface</code> and <code>-comm</code> option.
<table>
<tr><th>-target</th><th>-interface</th><th>-comm</th><th>Usage for</th></tr>
<tr><td>CICS_ZOS</td><td>DFHCOMMAREA</td><td>LINKAGE</td><td>CICS with DFHCOMMAREA calling convention for z/OS.</td></tr>
<tr><td></td><td>LINKAGE</td><td>LINKAGE
EXTERNAL
COPYBOOK</td><td>CICS with standard linkage calling convention for z/OS.</td></tr>
<tr><td>CICS_VSE</td><td>DFHCOMMAREA</td><td>LINKAGE</td><td>CICS with DFHCOMMAREA calling convention for z/VSE.</td></tr>
<tr><td></td><td>LINKAGE</td><td>LINKAGE
EXTERNAL</td><td>CICS with standard linkage calling convention for z/VSE.</td></tr>
<tr><td>BATCH_VSE</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL</td><td>Batch with standard linkage calling convention for z/VSE.</td></tr>
<tr><td>BATCH_BS2000</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL</td><td>Batch with standard linkage calling convention for BS2000/OSD.</td></tr>
<tr><td>BATCH_I5OS</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL</td><td>Batch with standard linkage calling convention for IBM i.</td></tr>
<tr><td>BATCH_ZOS</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL</td><td>Batch with standard linkage calling convention for z/OS.</td></tr>
<tr><td>IMS_BMP</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL
COPYBOOK</td><td>IMS BMP with standard linkage calling convention for z/OS.</td></tr>
<tr><td>IMS_MPP</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL
COPYBOOK</td><td>IMS MPP with standard linkage calling convention for z/OS.</td></tr>
<tr><td>IDMS_ZOS</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL
COPYBOOK</td><td>IDMS_ZOS with standard linkage calling convention for z/OS.</td></tr>
<tr><td>MICROFOCUS_WINDOWS</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL
COPYBOOK</td><td>Micro Focus with standard calling convention for Windows.</td></tr>
<tr><td>MICROFOCUS_UNIX</td><td>LINKAGE</td><td>LINKAGE
EXTERNAL
COPYBOOK</td><td>Micro Focus with standard calling convention for various UNIX operating systems.</td></tr>
</table>
</td></tr>
<tr><td><code>-copybooklevel</code></td><td>Define the beginning level for COBOL data items in generated copybooks, see <i>Starting COBOL Level for Data Items in Generated Copybooks</i>. Valid values: 1-49.</td></tr>
</table>

Option	Description
`-rpcservice`	Option to generate the generic RPC service module COBSRVI. See *Generate Generic RPC Service for Module COBSRVI*. Valid values: TRUE - Generate generic RPC service module. FALSE - Do not generate the generic RPC service module.

Generate a COBOL RPC Server from IDL File

To generate a COBOL RPC server from the specified IDL file, use the following command with options in table below:

```
-cobol:server
```

Option	Description
`-channel`	A CICS channel name can be provided for the interface type ’CICS with Channel Container calling convention’. See *Using the COBOL Wrapper for CICS with Channel Container Calling Convention (z/OS)*. See also *Channel Name*.
`-folder`	Folder where the COBOL files will be stored.
`-help`	Display this usage message.
`-interface`	Interface type, one of DFHCOMMAREA, DFHLBUFFER, DFHCHANNEL or LINKAGE. See table below for possible combinations.
`-literal`	Enclose string literals in quotes or apostrophes. See *Characters Used for String Literals*.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td><code>-target</code></td><td>Target operating system and environment. For possible combinations with option <code>-interface</code>, see below and also <i>Server Interface Types</i>.
<table>
<tr><th>-target</th><th>-interface</th><th>Usage for</th></tr>
<tr><td>CICS_ZOS</td><td>DFHCOMMAREA</td><td>CICS with DFHCOMMAREA calling convention for z/OS.</td></tr>
<tr><td></td><td>DFHLBUFFER</td><td>CICS with DFHCOMMAREA large buffer interface for z/OS.</td></tr>
<tr><td></td><td>DFHCHANNEL</td><td>CICS with Channel Container calling convention for z/OS.</td></tr>
<tr><td>CICS_VSE</td><td>DFHCOMMAREA</td><td>CICS with DFHCOMMAREA calling convention for z/VSE.</td></tr>
<tr><td></td><td>DFHLBUFFER</td><td>CICS with DFHCOMMAREA large buffer interface for z/VSE.</td></tr>
<tr><td>BATCH_VSE</td><td>LINKAGE</td><td>Batch with standard linkage calling convention for z/VSE.</td></tr>
<tr><td>BATCH_BS2000</td><td>LINKAGE</td><td>Batch with standard linkage calling convention for BS2000/OSD.</td></tr>
<tr><td>BATCH_I5OS</td><td>LINKAGE</td><td>Batch with standard linkage calling convention for IBM i.</td></tr>
<tr><td>BATCH_ZOS</td><td>LINKAGE</td><td>Batch with standard linkage calling convention for z/OS.</td></tr>
<tr><td>IMS_BMP</td><td>LINKAGE</td><td>IMS BMP with standard linkage calling convention for z/OS. This target may require a PSBLIST. See below.</td></tr>
<tr><td>MICROFOCUS_WINDOWS</td><td>LINKAGE</td><td>Micro Focus with standard linkage calling convention for Windows.</td></tr>
<tr><td>MICROFOCUS_UNIX</td><td>LINKAGE</td><td>Micro Focus with standard linkage calling convention for various UNIX operating systems.</td></tr>
</table>
</td></tr>
<tr><td><code>-psblist</code></td><td>An IMS PSB list containing IMS PCB pointers can be provided for the server interface type <i>IMS BMP with standard linkage calling convention</i>. See <i>Using the COBOL Wrapper for IMS BMP (z/OS)</i> for scenarios on PCB pointer usage. See also <i>IMS PSB List</i>.</td></tr>
</table>

Example Generating an RPC Client

```
<workbench> -cobol:client /Demo/example.idl -target CICS_ZOS
```

where `<workbench>` is a placeholder for the actual Workbench starter as described under *Using the EntireX Workbench in Command-line Mode*.

The name of the IDL file includes the project name. In the example, the project *Demo* is used. If the IDL file name describes a file inside the Eclipse workspace, the name is case-sensitive.

If the first part of the IDL file name is not a project name in the current workspace, the IDL file name is used as a relative (based on the IDL file) or absolute file name in the file system. Thus, the IDL files do not need to be part of an Eclipse project.

If you do not specify a folder (option `-folder`), the generated COBOL source files (client interface objects and the client declarations) will be stored in parallel to the IDL file, in the generated subfolders *client* and *include*, e.g. *Demo/client* and *Demo/include*.

Example Generating an RPC Server

```
<workbench> -cobol:server /Demo/example.idl -target CICS_ZOS
```

where `<workbench>` is a placeholder for the actual Workbench starter as described under *Using the EntireX Workbench in Command-line Mode*.

The generated COBOL source files (server (skeletons))

- will be stored in parallel to the IDL file, in the generated subfolder *server*, e.g. *Demo/server*.
- will overwrite existing files from a previous command-line mode generation.

Warning:
Take care not to overwrite an existing server implementation with a server skeleton. We recommend you to move your server implementation to a different folder.

Further Examples

Windows

Example 1

```
<workbench> -cobol:client C:\Temp\example.idl -folder src -target CICS_ZOS
```

Uses the IDL file *C:\Temp\example.idl* and generates the COBOL source files to the subfolder *src* of the IDL file. Slashes and backslashes are permitted in the file name. Output to standard output:

```
Using workspace file:\C:\myWorkspace\.
Run COBOL client wrapper with C:/Temp/example.idl and target CICS_ZOS.
Processing IDL file C:/Temp/example.idl
Store COBOL Source (1/2): C:\Temp\src/include/CALC
Store COBOL Source (2/2): C:\Temp\src/client/CALC
Exit value: 0
```

Example 2

```
<workbench> -cobol:client C:\Temp\*idl -folder C:\Temp\src -target CICS_ZOS
```

Generates COBOL source files for all IDL files in *C:\Temp*.

Example 3

```
<workbench> -cobol:client /Demo/example.idl -target CICS_ZOS
```

Uses the IDL file */Demo/example.idl* and generates the COBOL source files in parallel to the IDL file, here to the project */Demo*.

Example 4

```
<workbench> -cobol:client -help
```

or

```
<workbench> -help -cobol:client
```

Both calls result in displaying a short help for the COBOL client wrapper.

Linux

Example 1

```
<workbench> -cobol:client /Demo/example.idl -folder src -target CICS_ZOS
```

If the project *Demo* exists in the workspace and *example.idl* exists in this project, this file is used. Otherwise, */Demo/example.idl* is used from file system. The generated output will be stored in */Demo/src*, the subfolder of */Demo*.

Example 2

```
<workbench> -cobol:client /Demo/*.idl -folder src -target CICS_ZOS
```

Generates COBOL client interface objects for all IDL files in project *Demo* (or in folder */Demo* if the project does not exist). The generated files are in */Demo/src*.

Example 3

```
<workbench> -cobol:client -help
```

or

```
<workbench> -help -cobol:client
```

Both calls result in displaying a short help for the COBOL client wrapper.