

Using the C Wrapper in Command-line Mode

Commands are available to generate an RPC client or 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
-

Command-line Options

See *Using the EntireX Workbench in Command-line Mode* for the general command-line syntax. The table below shows the command-line options for the C Wrapper if the Workbench is used. Default values are underlined.

Task	Command	Option	Value	Description
Generate RPC client from the specified IDL file.	-c:client	-DATA_CONV_NP	0 1	<p>Mapping of IDL type P, PU, N and NU fields.</p> <p>0 The IDL data types are mapped to C data type unsigned char[..] with packed or unpacked (mainframe Natural, COBOL, PL/I style) contents.</p> <p>1 The IDL data types are mapped to C data type double.</p> <p>See <i>Mapping Options</i> for more information.</p>
		-DATA_CONV_A	0 1	<p>Mapping of IDL type A fields.</p> <p>0 Map IDL data type A to the C data type unsigned char[..] without null termination (mainframe Natural, COBOL, PL/I style).</p> <p>1 Map IDL data type A to C style strings.</p> <p>See <i>Mapping Options</i> for more information.</p>
		-DATA_CONV_U	0 1	<p>Mapping of IDL type U fields.</p> <p>0 Map IDL data type U to the C data type unsigned wchar_t[..] without null termination (mainframe Natural, COBOL, PL/I style).</p> <p>1 Map IDL data type U to wide char strings.</p> <p>See <i>Mapping Options</i> for more information.</p>
		-DATA_CONTEXT	0 1	<p>Multithread client.</p> <p>0 Off. The generated client interface objects can be used in single-threaded client environments.</p> <p>1 On. The generated client interface objects are thread-safe and can be used in multithreaded client environments.</p> <p>See <i>Generate RPC Client</i> for more information.</p>
		-help		Display this usage message.

Task	Command	Option	Value	Description
Generate RPC server from the specified IDL file.	-c:server	-DATA_CONV_NP	0 1	Mapping of IDL type N, P, NU and PU fields, for more information see above.
		-DATA_CONV_A	0 1	Mapping of IDL type A fields, for more information see above.
		-DATA_CONV_U	0 1	Mapping of IDL type U fields, for more information see above.
		-help		Display this usage message.

Note:

The commands "-C:Generate client" and "-C:Generate server" are deprecated. Use -c:client and -c:server instead.

Example Generating an RPC Client

```
<workbench> -c:client /Demo/example.idl -DATA_CONV_A 1
```

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

The generated C source files (client interface object and its associated header file)

- will be stored in parallel to the IDL file, for example in project *Demo*.
- will overwrite existing files from a previous command-line mode generation.

Example Generating an RPC Server

```
<workbench> -c:server /Demo/example.idl -DATA_CONV_A 1
```

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

The generated C source files (server interface object and its associated header file)

- will be stored in parallel to the IDL file, for example in project *Demo*
- 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 move your server implementation to a different folder, or rename the server implementation.

Further Examples

Windows

Example 1

```
<workbench> -c:client C:\Temp\example.idl
```

Uses the IDL file *C:\Temp\example.idl* and generates the C source files (*CEXAMPLE.c* and *CEXAMPLE.h*) in parallel to the IDL file. Slashes and backslashes are permitted in the file name. Output to standard output:

```
Using workspace file:/C:/myWorkspace/.  
Exit value: 0
```

Example 2

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

or

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

Both show a short help for the C Wrapper.

Linux

Example 1

```
<workbench> -c:client /Demo/example.idl
```

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 (*CEXAMPLE.c* and *CEXAMPLE.h*) will be stored in */Demo*, parallel to the IDL file.

Example 2

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

or

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

Both show a short help for the C Wrapper.