

Using the C Wrapper in IDL Compiler Command-line Mode

The table below shows the command-line options for the C Wrapper if the IDL Compiler is used. Default values are underlined. Options can be valid for client and server side, see column Client and Server.

Option	Client	Server	Value	Description
-DCONTEXT	Yes	No	<u>0</u> 1	<p>Multithreaded 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>
-DDATA_CONV_A	Yes	Yes	<u>0</u> 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>
-DDATA_CONV_NP	Yes	Yes	0 <u>1</u>	<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>
-DDATA_CONV_U	Yes	Yes	<u>0</u> 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>

See also *Starting the IDL Compiler* and *IDL Compiler Usage Examples*.

Example Generating an RPC Client

```
erxidl -t client.tpl -DDATA_CONV_A=1 -DDATA_CONV_U=1 example.idl
```

The generated C source files (client interface object and its associated header file) will be stored in parallel to the IDL file.

Example Generating an RPC Server

```
erxidl -t server.tpl -DDATA_CONV_A=1 -DDATA_CONV_U=1 example.idl
```

The generated C source files (server interface object and its associated header file) will be stored in parallel to the IDL file.