

Using the .NET Wrapper in IDL Compiler Command-line Mode

The table below shows the command-line options for the .NET Wrapper if the IDL Compiler is used. Options can be valid for client and server side.

Option	Req/ Opt	Description
-D BROKER= <i>nnn</i>	R	The EntireX Broker.
-D SERVICE= <i>nnn</i>	R	The EntireX Service.
-t <i>nnn</i>	R	Template for client (csharp_client.tpl) or server generation (csharp_server.tpl).
-o <i>nnn</i>	R	Project relative output directory or absolute Path
-D ATOSTRING=String StringBuilder	O	String handling (Default if omitted).
-D CLASSNAMEPREFIX= <i>nnn</i>	O	Class name prefix for inner classes.
-D A1TOCHAR= <i>n</i>	O	Generate "char" for A1 instead of String (1 if required).
-D B1TOBYTE= <i>n</i>	O	Generate "byte" for B1 instead of byte (1 if required).
-D TRIM= <i>n</i>	O	Remove trailing blanks (1 if required).
-PSANITIZE	O	Sanitize.
-F <i>nnn</i>	O	File base name for output.

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

Example

To start the IDL Compiler with the parameters for the stub generation, enter, for example the following in a single command line:

```
java -classpath "%ProgramFiles%\Software
AG\EntireX\classes\saglic.jar"; "%ProgramFiles%\Software
AG\EntireX\Classes\exxidlcompiler.jar"
-Dsagcommon="%CommonProgramFiles%\Software AG"
com/softwareag/entirex/idlcompiler/TplParser -PSANITIZE -D
BROKER="localhost:1971" -D SERVICE="RPC/SRV1/CALLNAT" -t
"%ProgramFiles%\Software AG\EntireX\Template\csharp_client.tpl" -F
example -o NET example.idl
```

The client stub is generated in the subdirectory NET.

Status and processing messages are written to standard output (stdout), which is normally set to the executing shell window.