Using the DCOM Wrapper in Command-line Mode

- Command-line Options
- Examples for Generating DCOM
- Further Examples

Command-line Options

See *Using the EntireX Workbench in Command-line Mode* for the general command-line syntax (note that option -data is not required). The table below shows the command-line options for the DCOM Wrapper.

Task	Command	Option	Description
Generate DCOM Wrapper objects for the specified IDL file(s).	`-dcom:generate`	`-help`	Display this usage message.
		`-broker`	The EntireX Broker.
		`-server`	The EntireX Service.
		`-productversion`	Specify the product version. Format: `<number>.<number>.<number>.<number>`
		`-fileversion`	Specify the file version. Format: `<number>.<number>.<number>.<number>`
		`-convnpdouble`	Natural data types Numeric and Packed Decimal to data type double (otherwise BSTR).
		`-logonnaturallibrary`	Logon to Natural library.
		`-generateproxy`	Generate a DCOM Proxy to enable the use of DCOM.
		`-proxyserverlocation`	Specify the IP address or location name of the generated DCOM Wrapper object.
		`-discardexistingguids`	Discard all existing GUIDs.
		`-registerobject`	Register the generated DOM Wrapper object after generating.
		`-enableASPScripting`	Create extended interface for ASP scripting support.
		`-stringtrimming`	Trim trailing space characters from string for the received string.
		`-exactvalue`	Check if data types N,P,NU or PU after converting contain the original value.
		`-aonames`	Specify the name of the registered objects. Format: `library=aoname[,library=aoname]*`
		`-dllnames`	Specify the name of DLL files. Format: `library=dllname[,library=dllname]*`
		`-compiler`	Location of Microsoft Visual Studio 2010.
		`-output`	Directory for generated batch and other files (relative to IDL file). The location of the generated files depends on the name of the output directory. **NT** Default. Batch files are generated in the same directory as IDL file; other files are generated to subdirectory *NT*. This was the behavior in the initial version of EntireX 9.0 or earlier. ***other*** Batch and other files are generated in a user-defined subdirectory.

Examples for Generating DCOM

```
<workbench> -dcom:generate /Demo/Example.idl
```

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 within 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.

The generated DCOM source files will be stored in parallel to the Software AG IDL file, in the generated subfolders *win32\<Library Name>*, e.g. *Demo\win32\EXAMPLE*.

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

Further Examples

Example 1

```
<workbench> -dcom:generate C:\Temp\example.idl
```

Uses the IDL file *C:\Temp\example.idl* and generates the DCOM source files (several .bat files, the subfolders *win32\EXAMPLE* within the different files) in parallel to the IDL file. Slashes and backslashes are permitted in the file name. Output to standard output:

```
Using workspace file:/C:/myWorkspace/.
LIBRARY = EXAMPLE
        Program   = CALC
        Program   = POWER
        Program   = HELLO
(C) Copyright Software AG 2000-2008. All rights reserved.
Set environment for MS Visual Studio C++
...
Exit value: 0
```

Example 2

```
<workbench> -dcom:generate -help
```

or

```
<workbench> -help -dcom:generate
```

Both show a short help for the DCOM Wrapper.