

# Broker Command-line Utilities

The Web Services Wrapper generates a WSDL file, a mapping file (extension .xmm) and a service archive (extension .aar) to deploy into the common Web Services Stack. This section covers the following topics:

- Command-line Options
- Example for Generating Web Services
- 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 option for the Web Services Wrapper.

Task	Command	Option	Description
Generate WSDL, mapping and archive files from specified IDL file.	-wsdl	-out	Output directory, absolute path (fully qualified, must exist). Ignored if the input is part of a project in the Eclipse workspace. Same as -o.
		-url	Service URL. Same as -u.
		-service=<service>	Service name.
		-properties	Use the file-specific properties. This option makes the others superfluous, but is only available if the input is part of an Eclipse project.

## Example for Generating Web Services

```
<workbench> -wsdl /Demo/example.idl -properties
```

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 mapping file gets the name of the IDL file. The WSDL file and the service archive get the name of the service, if specified, otherwise they get the name of the IDL file.

```
<workbench> -wsdl /Demo/example.xmm -properties
```

This command generates the WSDL file and the service archive from the mapping file. If a service name is specified, the WSDL file and the service file get the name of the service, otherwise they get the name of the mapping file.

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

## Further Examples

### Windows

#### Example 1

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

Uses the IDL file *C:\Temp\example.idl* and generates the files (*EXAMPLE.wsdl* and *example.xmm*) 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   = SQUARE
WSDL file "C:\Temp\EXAMPLE.wsdl" created.
Exit value: 0
```

#### Example 2

```
<workbench> -wsdl -help
```

or

```
<workbench> -help -wsdl
```

Both show a short help for the Web Services Wrapper.

### Linux

#### Example 1

```
<workbench> -wsdl /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 (*EXAMPLE.wsdl* and *example.xmm*) will be stored in */Demo*, parallel to the IDL file.

#### Example 2

```
<workbench> -wsdl -help
```

or

```
<workbench> -help -wsdl
```

Both show a short help for the Web Services Wrapper.