Using the Wrapper for EJB in Command-line Mode

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 Wrapper for EJB.

The behavior of the generation in command-line mode depends on the IDL file:

- If the IDL file is part of the workspace, the generator generates the sources, controlled by the properties of the IDL file. The properties can be overwritten by the command-line options.
- If the IDL file is not part of the workspace but part of the file system, the generator creates the subdirectory EJB in the parent directory of the IDL file and generates the source files into this subdirectory. The generation can be controlled by the command-line options.

Task

Generate the EJB sources for the specified IDL file(s).

Command	Option	Default Value	Description
`-ejb:generate`	`-broker`	localhost:1971	Broker identification
	`-brokerpassword`	null	Broker user password.
	`-brokeruser`	EjbUserName	Broker user ID.
	`-compresslevel`	no	Compression level (No,Yes,0-9).
	`-encryption`	0	Encryption level (0,1,2).
	`-log`	null	Java ACI trace file.
	`-logonnaturalsecurity`	false	Logon to Natural.
	`-package`	IDL file name without extension	Package name for EJB classes.
	`-packageprefix`	null	Package name prefix.
	`-rpcpassword`	null	RPC user password.
	`-rpcuser`	null	RPC user ID.
	`-security`	false	Use EntireX Security.
	`-server`	RPC/SRV1/CALLNAT	Service name (class/server/service).
	`-sourcefolder`	<IDL file parent directory>/EJB	Folder of the generated classes.
	`-trace`	0	Java ACI trace level (0,1,2).
	`-usecodepage`	false	Send default codepage to the broker.
	`-verbose`	false	Enable EJB log information.

Example

```
<workbench> -ejb:generate /Demo/Example.idl -sourcefolder /Demo/src1
```

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 and the source folder include the project name. In the example, the project *Demo* is used.

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 file name in the file system. Thus, the IDL files do not need to be part of an Eclipse project.

If the source folder does not exist in the workspace but the first part describing the project exists, the source folder is created.

If the IDL file is located outside the Eclipse workspace, the source folder is also a folder in the file system.

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