Using the EntireX Wrapper for Enterprise JavaBeans

This chapter covers the following topics:

- Generation Process
- Using the Wrapper for EJB Interactively
- Generated Classes and Interfaces
- Delivered Example

Generation Process

To generate the Enterprise JavaBeans (EJB) source code, use the *EntireX Workbench*. This can be done interactively with the UI of the *EntireX Workbench*. The generation is controlled by the following properties:

Default Properties for the IDL File

Description	Data Type	Default Value
Broker ID	String	Localhost:1971
Server class	String	RPC
Service class	String	SRV1
Service name	String	CALLNAT
Package name	String	IDL file name without extension
Package name prefix	String	empty string

Generated Files

During the generation process for each library with the name *<libname>*, the following interfaces and classes source files are created in the subdirectory *EJB* of the home directory of the *<name>.idl* file.

• The interfaces are created in the subdirectory:

<Package Name Prefix><Package Name><file.separator>interfaces.

They will be a component of the package: <Package Name Prefix><Package Name>.interfaces

Naming Conventions	Description
EJB <libname>.java</libname>	The remote interface.
EJB <libname>Home.java</libname>	The home interface.

• The EJB classes will be created in the subdirectory:

<Package Name Prefix><Package Name><file.separator>ejb

They will be a component of the package: <*Package Name Prefix*><*Package Name*>.ejb

Naming Conventions	Description
EJB <libname>Bean.java</libname>	The enterprise bean class.
<libname><innerclassname>.java</innerclassname></libname>	Serializable classes for Software AG IDL groups/structs.
<libname><progname>Input.java</progname></libname>	Serializable holder class for all IN and IN/OUT parameters of the program.
<libname><progname>Output.java</progname></libname>	Serializable holder class for all OUT and IN/OUT parameters of the program.
<libname>Mapper.java</libname>	Mapper class.

To build the JAR files for the different application servers, we generate an Ant script which uses the XDoclet tool. This file will be generated in the *EJB* subdirectory:

<Package Name Prefix><Package Name>.xml

If the package prefix/name contains dots, subdirectories will be created, for example: *abc.def.library* will become *abc/def/library/...*

Using the Wrapper for EJB Interactively

- Using the Wrapper for EJB Functions
- Setting/Modifying EntireX Enterprise JavaBeans Preferences
- Setting/Modifying IDL File Properties

Using the Wrapper for EJB Functions

To use the Wrapper for EJB functions

- 1. Open the *EntireX Workbench*.
- 2. Create a new Java Project (e.g.: "EJB"), using **File > New > Project**.

Sew Project	
Select a wizard	
Create a Java project	
<u>W</u> izards: 	
Java Java Project Java Project from Existing Ant Buildfile Simple Software AG	
< <u>Back</u> <u>Next</u> > <u>Finish</u> Ca	incel

😂 New Java Project 🛛 🛛
Create a Java project Create a Java project in the workspace or in an external location.
Project name: Contents © Create new project in workspace © Create project from existing source Directory: C:\home\user\EclipseScreens\Workspace Bgowse JDK Compliance © Use default compiler compliance (Currently 1.4) Configure default © Use a project specific compliance:
Project layout Use project folder as root for sources and class files C create separate source and output folders Configure default
< <u>B</u> ack <u>N</u> ext > <u>Einish</u> Cancel

3. Add the EntireX classes to the build path of the project (*entirex.jar*).

	Source Relactor Navigate Se		un Utenlaru Unie				
(La .	B B Ø • O • Q •]		i 🗊 i 🕭 🛷 i 🎱 i 👔 -	<u>[</u>] - 1,5 (2	- ¢ -	Java	
Packag	ge Explor 🐹 🔰 Hierarchy					an outline is not available.	
	New Go Into	•					
	Open in New Window Open Type Hierarchy Show In	F4 Alt+Shift+W ♪					
	Copy Copy Qualified Name Paste	Ctrl+C Ctrl+V					
	Build Path Source Refactor	Alt+Shift+S Alt+Shift+T	器 Link Source New Source Folder				
i i i	≥ Import 2 Export		Image: Source Folder Image: Add External Archives Image: Add Libraries				
	Refresh Close Project Assign Working Sets	F5	🇞 Configure Build Path				
	Run As Debug As Profile As) } }	s 🛛 🥥 Javadoc 🔯 Declaration	[바다	~ - 6
	Validate Team Compare With Restore from Local History PDE Tools Software AG Web Services Stac	• • k •	varnings, O inhos n 🔺	Resource	Path	Locatio	n
-	Properties	Alt+Enter					

JAR Selection				? 🛛
Look jn:	🗁 classes	•	← 🗈 📸 -	
My Recent Documents Desktop My Documents My Computer	uddi4j-1.0.3 activation.jar ant.jar bsf.jar com.ibm.mq.jar domsdk.jar entirex.jar entirex.jar entxrt.jar exxargus.jar exxargus.jar exxcorba.jar exxcorba.jar exxidicompiler.jar exxjms.jar	exxjmsdev.jar exxutil.jar exxutildev.jar fscontext.jar jakarta-regexp-1.2.jar jolel.jar jolel.jar jolel.jar jolel.jar jolel.jar jolel.jar jolel.jar jolel.jar jolel.jar jolel.jar mail.jar providerutil.jar saglic.jar	smtp.jar soap.jar uddi4j.jar w3cdom1.jar w3cdom2.jar xdoclet.jar xercesImpl.jar xml-apis.jar	
My Network Places	File name: entirex. Files of type: *.jar;*.z	jar ip	•	<u>O</u> pen Cancel

4. Add the Enterprise JavaBeans classes to the build path of the project (e.g.: *j2ee.jar*).

Java	- Eclipse SDK - C:\Workspa	ce EJB	Barris Discloser Dales			
e Edic [1] - 상] -		Search Project		1 🕭 🛷 1	•	Java
Packa	ige Explor 🛛 🏌 Hierarchy) (> -> 🐼 🕞 1	□ □ <u>}</u> ▽			An outline is r	available.
	New Go Into		×			
	Open in New Window Open Type Hierarchy Show In	F4 Alt+Shift+W	•			
	Copy	Ctrl+C				
	🕋 Paste 💥 Delete	Ctrl+V Delete				
	Build Path		Link Source			
	Source Refactor	Alt+Shift+S Alt+Shift+T	New Source Folder	8		
	≥ Import ≥ Export		Add External Archives	s		
	Refresh Close Project Assign Working Sets	F5	🎇 Configure Build Path.			
	Run As Debug As		Bavadoc	2 Declaration		<mark>⊉</mark> ⊂ E
	Profile As Validate Team		, anings, o in os	Resource	Path	Location
	Compare With Restore from Local History PDE Tools	4 <u>0</u>	• •			
	Software AG Web Services S	tack	•			
	Properties	∆lt⊥Enter	-			3

JAR Selection		? 🗙
Look jn:	🔁 lib 🔹 🛨 💌	
My Recent Documents Desktop My Documents My Computer	 classes cloudscape connector dtds locale security system ejb10deployment.jar j2ee.jar j2ee-ri-svc.jar jhall.jar servlet.jar 	
My Network Places	File <u>n</u> ame: j2ee.jar	<u>O</u> pen
	riles or type:	Cancel

5. Import the IDL file into the project.

🖨 Java	- Eclipse SDK - C:\Workspace	e EJB					
File Edit	Source Refactor Navigate Se	earch Project R	un Window Help				
: E3 •	🖩 🖻 i 🎄 • 🔘 • 💁 •	B 🖶 🞯 -	1 📴 1 🕭 🛷 1 🕥 1 🖢 -	§]-*⊳ ⇔	- ф -	III Java	
Packa	age Explor 🙁 👔 Hierarchy 🖻		1		- 0	An outline is not available.	- 0)
	New Go Into						
	Open in New Window Open Type Hierarchy Show In	F4 Alt+Shift+W ▶					
	Copy Copy Qualified Name	Ctrl+C					
	Delete	Ctrl+V Delete					
	Build Path Source Refactor	► Alt+Shift+S ► Alt+Shift+T ►					
	Import Z Export						
	Refresh Close Project Assign Working Sets	F5					
	Run As Debug As Profile As) 	s 🛛 🖉 Javadoc 😥 Declaration	1		± 10 10 10 10 10 10 10 10 10 10 10 10 10	
	Validate		varnings, 0 infos		1 - 9		
	Team Compare With Restore from Local History	*	n	Resource	Path	Location	
	PDE Tools Software AG Web Services Stat	• :k •					
	Properties	Alt+Enter					
	ЕЈВ				1		

🖨 Import	
Select Import resources from the local file system into an existing project.	è
Select an import source:	
type filter text	
 General Archive File Breakpoints Existing Projects into Workspace File System Preferences CVS E JB J2EE Plug-in Development Team Web Web services 	
(?) < Back	Cancel

🖨 Import			
File system Import resources	from the local file system.		
From directory:	C:\Program Files\Software AG\Ent Wrapper	ireX\Examples\Java Wrapper	Browse
Into folder: EJE	Select All Deselect All		Browse
0	< <u>B</u> ack	K Next > Einish	Cancel

6. Activate the *example.idl* file.

🖨 Java - EJB/example.idl - Eclipse SDK	C:\Workspace EJB		
Eile Edit Source Refactor Navigate Search	<u>Project Run Window H</u> elp		
Elle Edit gource Reractor Wavigate Search	Project Kun Wundow Hep Control Control Contro	CALC CALC CALC CALC Coperation (A1) In Operand_1 (14) In Operand_2 (14) In Function_Result (14) RESULT POWER	
	Library 'EXAMPLE' IS Program 'CALC' IS Define Data Parameter 1 Operation (A1) In 1 Operand_1 (I4) In 1 Operand_2 (I4) In 1 Function_Result (I4) Out End-Define Program 'POWER' IS Define Data Parameter 1 Operand (I4) In 1 Function_Result (I4) Out End-Define	POWER Operand (I4) In Function_Result (I4) RESULT P HELLO Client (A80) In Mail (A80) InOut	
	Problems 22 @ Javadoc @ Declaration		
	0 errors, 0 warnings, 0 infos	-#T	
	Description 🔺 Resource Pat	h Location	
example.idl - EJB	1		

Setting/Modifying EntireX Enterprise JavaBeans Preferences

- **To set/modify the preferences**
 - 1. Display the Preferences window.



Preferences				
type filter text		EntireX		⇔ - ⇔ -
ia Help ⊡ Install/Update	^	General EntireX (Class_Server	IDL preferences to specify the Broker ID and the Service Service) that are used in the various EntireX Wranners	Description
		Ricker ID:	localbost: 1971	
🏝 Java		DIONEI ID.		
JPA		Server <u>⊂</u> lass:	RPC	
🗈 Plug-in Development		Server Name:	SRV1	
■ Report Design				
⊞ Run/Debug		Service:	CALLNAT	
Custom Wrapper				
DCOM Wrapper				
- Deployment Environments				
- EJB Wrapper				
IDL Extractor for COBOL				
IDL Extractor for Natural				
IDL Extractor for PL/I				
- Installation				
- Java Wrapper				
PL/I Wrapper				
RPC Environments				
····· XML Mapping Editor				
Proxy Settings				
UDDI Registries			Restore <u>D</u> efaults	Apply
E Web Services Stack				
0			ОК	Cancel

Preferences			
type filter text		EJB Wrapper	← - → -
ia-Help ia Install≬uodate	^	Package name:	
tariyopuate tariyopuate		Package name prefix:	
		r dekage name pren <u>z</u> .	
JPA			
🕀 Plug-in Development			
🗐 Report Design			
🔄 Run/Debug			
连 - Server			
🗐 Software AG			
EntireX			
C Wrapper			
COBOL Wrapper			
Custom Wrapper			
Declowweat Environments			
E 18 Wrapper			
IDL Extractor for COBOL			
IDL Extractor for Natural			
IDL Extractor for PL/I			
- Installation			
Java Wrapper			
PL/I Wrapper			
···· RPC Environments			
····· XML Mapping Editor			
Proxy Settings			
UDDI Registries			Restore <u>D</u> efaults <u>Apply</u>
E ± Web Services Stack	_		
0			OK Cancel

Setting/Modifying IDL File Properties

- **To set/modify the properties**
 - 1. Display the Properties window; use either the pop-up menu or, from the **File** menu, choose **Properties**.



2. In the **Properties** window, choose the **EntireX** tab.

Here it is possible to modify the common default properties for the IDL file.

🖨 Properties for example.idl			
type filter text	EntireX	\$ • \$ -	
Info EntireX EntireX .NET Wrapper EntireX COBOL Wrapper EntireX Custom Wrapper EntireX C Wrapper EntireX DCOM Wrapper EntireX EJB Wrapper EntireX Java Wrapper EntireX PL/I Wrapper EntireX Web Service Wrapper	General EntireX IDL properties to specify the Broker ID and the Service Description (Class, Server, Service) that are used in the various EntireX Wrappers. The default settings are provided by the EntireX preference page.		
	<u>B</u> roker ID:	localhost:1111	
	Server <u>C</u> lass:	RPC	
	Server <u>N</u> ame:	SRV1xxxx	
	<u>S</u> ervice:	CALLNAT	
		Restore Defaults Apply	
		OK Cancel	

3. Choose the EntireX Wrapper for Enterprise JavaBeans tab.

Here it is possible to modify the defaults (see table *Default Properties for the IDL File*, below).

🖨 Properties for example.idl			
type filter text	EntireX EJB Wrap	pper 🔶	• 🔿 -
 Resource EntireX EntireX CWrapper EntireX COBOL Wrapper EntireX Custom Wrapper EntireX DCOM Wrapper EntireX EJB Wrapper EntireX Java Wrapper EntireX PL/I Wrapper Run/Debug Settings 	<u>J</u> ava Source Folder: E <u>P</u> ackage name: Package name prefi <u>x</u> :	EJB/src	Browse
0		ОК Са	ancel

4. Confirm the entries with **OK**.

> To generate the EJB sources

1. From the context menu, choose **Other > Generate EJB**.

a 🗁 Demo .projec 📴 Examp		New Open Open With	k F	
		Copy Paste Delete Move Rename Import		
	2	Export Refresh		
		Validate Show in Remote Systems view Profile As Debug As Run As Replace With	F F F	
	<u>8</u> 8 8 8	COBOL Integration Server Natural Web Service) 	Monitor 🔂 EntireX Default B
	B	Other	۲	Generate C 🕨 🕨
8	3	Refactor Software AG IDL Software AG IDL Tester		Generate DCOM Generate EJB

2. All generated EJB sources are in the subdirectory EJB of the home directory of the IDL file.

🖨 Java - EJB/example.idl - Eclipse SDK -	C:\Workspace EJB			
Elle Edit Source Refactor Navigate Search Project Run Window Help				
📬 - 🔛 👜 🎄 - O - 💁 🖉	₩ @• # #	🔛 🐉 Java		
Package Explor X Hierarchy EJB Fr EDExample.ejb EXample.java ExampleCalcInput.java ExampleCalcOutput.java ExampleHelloOutput.java	<pre>** This IDL file contains the IDL definition ** This IDL file contains the IDL definition ** CALC, POWER and the procedure HELLO. ** ** (c) Software AG ** ** ** ***************************</pre>	Cutine S Cutine S CAUC Coperation (A1) In Coperand_1 (14) In Coperand_2 (14) In Coperand_2 (14) In Coperand (14) In Coperan		
i =0 averale till E15				
example.idl - EJB				

Generated Classes and Interfaces

The workbench generates the following interfaces and classes from the Software AG IDL file sources.

Home Interface

The generated home interfaces *EJB*<*libname*>*Home.java* contain the create methods. These methods can be called by an EJB client.

create()

```
EJB<libname> create()
```

Calling this method creates an EJB with the default settings.

create(String user)

```
EJB<libname> create(String user)
```

where: String user is the Broker user name

Calling this method creates an EJB with the default settings except for the user name.

create(String user, String password)

EJB<libname> create(String user,String password)

where: String user is the Broker user name and String password the Broker user password

Calling this method creates an EJB with the default settings except for the user name and the password.

Remote Interface

The generated remote interfaces *EJB*<*libname*>.*java* contain the customer methods. For each RPC program a customer method is generated with the following naming conventions.

<libname><program>Output <program> (<libname><program>Input input)

where: Output is the output object and Input is the input object

These methods can be called by an EJB client.

EJB Class

The generated EJB class *EJB*<*libname*>*Bean.java* extends <*libname*>*Mapper.java* class. Bean implements the EJB session bean and controls the EntireX RPC communication.

Mapper Class

The generated *<libname>Mapper.java* class extends the Java RPC client stub *<libname>.java*. It implements a method for each RPC Program. Methodname is *<program>*. The method has one parameter, an instance of *<libname><progname>*Input.class.The return value of the method is an instance of *<libname><progname>*Output.class.

Group Classes

The group classes are serializable representations of the inner classes of the groups of the RPC library, filename is *<libname><progname><innerclassname>.java*. The class contains all parameters of the group.

Struct Classes

The struct classes are serializable representations of the inner classes of the structs of RPC library, filename is *<libname><innerclassname>.java*. The class contains all parameters of the struct.

Input/Output Classes

These classes are serializable representations of all in, out, in/out parameters of RPC program. The filename of the input classes is *<libname><progname>Input.java*, it holds all in and in/out parameters of the program. The filename of the output classes is *<libname><progname>Output.java*, it holds all out and in/out parameters of the program.

Java RPC Client Stub

The generated Java RPC Stub *<libname>.java* extends com.softwareag.entirex.aci.RPCService class. It is the connecting link between the EJB and the EntireX server.

Example



Delivered Example

An example is delivered in directory *<drive>:\SoftwareAG\EntireX\examples\EJBWrapper* (Windows) or */opt/softwareag/EntireX\examples\EJBWrapper* (UNIX).

See the README.TXT.