Developing an Application Client to access EJB

Deploy and Run a Java EE Application Client

✤Application clients

This tutorial walks you through configuring, developing and deploying an enterprise application with Eclipse and Geronimo. To run this tutorial, as a minimum you will be required to have installed the following prerequisite software.

- Sun JDK 5.0+ (J2SE 1.5)
- Eclipse 3.3.1.1 (Eclipse Classic package of Europa distribution), which is platform specific
- Web Tools Platform (WTP) 2.0.1
- Data Tools Platform (DTP) 1.5.1
- Eclipse Modeling Framework (EMF) 2.3.1
- Graphical Editing Framework (GEF) 3.3.1

Details on installing eclipse are provided in the Development environment section. This tutorial is organized in the following sections:

- Setting up Eclipse IDE for EJB application development.
- Developing EJB Local interface, EJB Remote Interface and Bean Class
- Developing a Java Client
- Deploy and Run

Let us briefly understand this application. This application will take you through creating a simple Stateless Session EJB. Later we will develop a Java Application client to access this EJB. EJB development will make use of annotations which are introduced in Java EE5.

Setting up Eclipse IDE for EJB application development.

1. Launch Eclipse and select Window->Open Perspective->Other->JavaEE and select Ok

Open Perspective
CVS Repository Exploring Database Debug Database Development Debug Java (default) Java Browsing Java EE Java Type Hierarchy JPA Development Plug-in Development Resource Team Synchronizing
OK Cancel

2. Right Click under Project Explorer and create a new EJB project. Mention the fields as shown in the figure. For all the other windows give default values and select Finish.

New EJB Pi	oject			_ 🗆 🔀
EJB Project Create an EJB P	roject and add it to a new or existing En	iterprise Applica	tion.	
Project name:	SimpleEJB			
Project conten	ts:			
Use <u>d</u> efau	t			
Directory: C:	Documents and Settings\Administrator\	ag1\SimpleEJB		Browse
⊂ Target Runtime	•			
Apache Geron	imo v2.1 Runtime			✓ N <u>e</u> w
Configurations				
Default Config	uration for Apache Geronimo v2.1 Runti	ime		▼
A good starting new functional	for working with Apache Geronimo v2. : ty to the project.	1 Runtime runtin	ne. Additional facets can lat	er be installed to add
Add project	to an EAR			
EAR Project Na	me; EAR			✓ New
0		< <u>B</u> ack		h Cancel

3. Under Project Explorer, Right click on SimpleEJB and create a new package ejb. Mention the fields as shown in the figure. Select Finish.

😂 New Java	Package		_ 🗆 🔀
Java Packag Create a Java	je package.		
Creates folders	corresponding to packages.		
Source folder:	SimpleEJB/ejbModule		Browse
Name:	ejb		
0		Finis	Cancel

4. Right click on SimpleEJB project under project explorer and select properties.



5. Select Java Build Path and select Add External JARs.



6. Browse to the directory <GERONIMO_HOME>\repository\org\apache\geronimo\specs\geronimo-ejb_3.0_spec\1.0.1 and select geronimo-ejb_3. 0_spec-1.0.1.jar.

JAR Selection							? 🔀
Look in:	1.0.1		~	3	3 🖻	•	
CO Recent	geronimo-ejb_3	3.0_spec-1.0.1.jar					
Desktop							
My Documents							
My Computer							
	File name:	geronimo-ejb_3.0_	_spec-1.0.1.jar		~		<u>Rpen</u>
My Network	Files of type:	*.jar;*.zip			~	(Cancel

7. Once done select Ok. You can see the geronimo-ejb_3.0_spec-1.0.1.jar added to the build path. This jar provides the essential classes required for the EJB development.



This completes the setting of Eclispe IDE for EJB application development.

Developing EJB Local interface, EJB Remote Interface and Bean Class

1. Right Click on the ejb package and create a new Remote interface CountryCapital as shown in the figure

ava EE - Ecl	ipse SDK		
File Edit Navigat	te Search Project Run Wir	ndow Help	
i 📬 • 🖬 🖻	🎄 • 🔘 • 💁 • 🖻	🖻 👸 I	Sî 😰 🥭 🛷
Project Explore	r X		
	□ 🕏 🎽		
🖃 🎥 SimpleEJB			
ejbMoo	dule		
⊕ - <mark>⊖</mark> ME	New	•	Project
🗄 📥 JRE Sy	Open Type Hierarchy	F4	@ Annotation
🕀 🔬 geroni	Show In	Alt+Shift+W	Class
EAR Li		Ctrl+C	💕 Enum
🕀 🗁 build	Copy Qualified Name		😯 Interface
	Paste	Ctrl+V	🖶 Packinge
	💢 Delete	Delete	Example
	Build Path	•	
	Source	Alt+Shift+S	📑 Other
	Refactor	Alt+Shift+T	
	🚵 Import		
	🛃 Export		_
	🔊 Refresh	F5	_
	Run As	•	A
	Debug As	•	🚈 Tasks 🔛 Properties 🖗
	Profile As	•	Sta
	Validate		eronimo v2.1 Server at 📜
	Team	•	
	Compare With	•	
	Restore from Local History.		_
	Properties	Alt+Enter	
<	>		
1 🗘 🕀	ejb - SimpleEJB/ejbModule		

2. Enter the fields as shown in the figure and select Finish.

😂 New Java Inter	face				
Java Interface Create a new Java in	terface.				I
Source folder:	SimpleEJB/e	ejbModule			Browse
Package:	ejb]	Browse
Name:	CountryCa	pital			
Modifiers: Extended interfaces:	public	⊖ default) private	 protected 	Add
Do you want to add c	omments as o	configured in the	properties of	the current proje	Remove
	Generate	comments	<u>- 21 000 000</u> 01	and content proje	
0				Finish	Cancel

3. Add the following code as shown below

CountryCapital.java
package ejb;
<pre>import javax.ejb.Remote;</pre>
<pre>@Remote public interface CountryCapital {</pre>
<pre>public String capitaiName(String countryName); }</pre>

Here @Remote is an annotation used to declare the interface as a Remote Interface. 4. Similarly create a Local interface **CountryCapitalLocal**. Add the following code to the interface

CountryCapitalLocal.java package ejb; import javax.ejb.Local; @Local public interface CountryCapitalLocal { public String capitalName(String countryName); }

Similarly in this code we have @Local annotation which declares this interface as local.

In remote as well as local interface we have the declaration for only one Business method. So only this method will be visible to the application

client. Methods other than capitalName if implemented in the Bean class will be private to Bean class and will not be visible to application client.

5. Next step is to create a Bean Class which will implement the Business method to be executed. Right Click on the jsf package and create a new class.

File Edit Sourc	e Refactor Navigate Searc	th Project Run	Window Help
i 📬 • 🖪 🖻	🎄 • 🜔 • 💁 • (2 et i 📬 i	😂 i 😰 i 🕭 🔗
Project Explo	rer 🛛 🖵 [🗏 🚺 CountryCa	pital.java 🚺 Countr
	E 🔄 '	packag	ge ejb;
🖃 🞥 SimpleEJ	В		
🗐 🕮 ejbM	odule	Import	t javax.ejb.Loca.
	New	•	Project n1
	Open Type Hierarchy	E4	p:
	Show In		Annotation
	510W11	ARTSHITCHW	
🗄 🔬 gero	Copy	Ctrl+C	🕃 Enum
🛋 Apac	🗎 Copy Qualified Name		🗊 Interface
🛋 EAR	💼 Paste	Ctrl+V	🖶 Package
🗄 🗁 build	💢 Delete	Delete	Evample
	Build Path	+	
	Source	Alt+Shift+S 🕨	📑 Other
	Refactor	Alt+Shift+T 🕨	
	Rea Import		-
	Fxport		
			-
	🔊 Refresh	F5	
	References	•	
	Declarations	•	🕗 Tasks 🔲 Propertie
	Run As	•	
	Debug As	•	Geronimo v2.1 Server at
	Drofile As	•	

6. Enter the class name as CountryCapitalBean.

😂 New Java Clas	S	_ 🗆 🔀
Java Class		0
Create a new Java	dass.	\bigcirc
Source folder:	SimpleEJB/ejbModule	Browse
Package:	ejb	Browse
Enclosing type:		Browse
Name:	CountryCapitalBean]
Modifiers:	public Odefault Oprivate Oprotected	
	abstract final static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add
		Remove
Which method stubs	would you like to create?	,
	public static void main(String] args)	
	Constructors from superclass	
	✓ Inherited abstract methods	
Do you want to add	comments as configured in the properties of the current pro	ject?
	Generate comments	
0	Finth	Cancel

7. Populate the bean class with the code as follows

CountryCapitalBean.java

```
package ejb;
import javax.ejb.Stateless;
@Stateless
public class CountryCapitalBean implements CountryCapital,CountryCapitalLocal{
        public String capitalName(String countryName)
        {
                String capital=new String("No such country");
                if (countryName.equalsIgnoreCase("India"))
                {
                        capital="New Delhi";
                }
                if (countryName.equalsIgnoreCase("United States Of America"))
                {
                        capital="Washington DC";
                }
                if (countryName.equalsIgnoreCase("China"))
                {
                        capital="Bejing";
```

```
}
return capital;
}
}
```

This completes the development of EJB application.

Warning Due to some limitations in Geronimo Eclipse Plugin the generated deployment plan(openejb-jar.xml) does not have the correct namespace. Replace the existing namespace as shown in the figure with the following <openejb-jar xmlns="http://www.openejb.org/xml/ns/openejb-jar-2.2" xmlns:nam="http://geronimo.apache.org/xml/ns/naming-1.2" xmlns:pkgen=" http://www.openejb.org/xml/ns/pkgen-2.0" xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.2" xmlns:sys="http://geronimo.apache.org/xml/ns/security-1.2" xmlns:sys="http://geronimo.apache.org/xml/ns/sec



Developing a Java Client

1. Select File->New->Project->Java->Java Project. Select Next.

New Project	_ 🗆 🔀
Select a wizard Create a Java project	
<u>W</u> izards:	
type filter text	
Java Project Java Project from Existing Ant Buildfile Plug-in Project General CVS Eclipse Modeling Framework EJB J2EE Java Project Java Project Java Project Plug-in Development Web	
Image: Second	Cancel

2. In the New Project Window give the Project name ss ApplicationClient and select Next.

New Java Project		_ 🗆 🔀
Create a Java project Create a Java project in the workspace or in	n an external location.	
Project name: ApplicationClient Contents Create new project in workspace Create project from existing source		
JRE Use default JRE (Currently 'Java50') Use a project specific JRE: Use an execution environment JRE:	Java50	Configure default
Project layout Use project folder as root for sources O Create separate folders for sources a Working sets	and class files ind class files	Configure default
Add project to working sets Working sets:		Select
?	ck Next	Finish Cancel

3. On the Java Settings window. Select the Projects tab and click add.

New Java Project	_ 🗆 🔀
Java Settings Define the Java build settings.	
Bequired projects on the build path:	
	Add
? < Back Next >	Finish Cancel

4. Check the box for SimpleEJB and Select OK.



5. This will add the SimpleEJB project to the build path of the Application Client as shown in the figure. Select Finish.

🗢 New Java Project	_ 🗆 🔀
Java Settings Define the Java build settings.	r
😕 Source 🗁 Projects 🛋 Libraries 🗞 Order and Export	
Required projects on the build path:	
🕀 🗁 SimpleEJB	Add
	Edit
	Remove
? < Back Next >	Finish Cancel

Why to add EJB project to build path :(

This is because the build path is used to find the classes referenced by your Client source code. These classes will be required for compilation of Client source code.

6. Right click on package appclient and create a new Java Class.

New Java Project	
Java Settings Define the Java build settings.	
😕 Source 🗁 Projects 🛋 Libraries 😽 Order and Export	
Required projects on the build path:	
t i i i i i i i i i i i i i i i i i i i	Add Edit Remove
? <	Finish Cancel

7. Give the class name as ApplicationClient as shown in the figure. Select Finish.

ava EE - Eclips	e SDK							
File Edit Navigate	Search Project Run Window	/ Help						
	🏇 • 🜔 • 💁 • 🗄 🖻 🖻	1 📬 1 📚	i 😭 i 😕 🔗 i					
Project Explorer 🛛 🖓 🖓								
📮 🔁 ApplicationClie	nt							
🖻 🗁 src								
H → BE System	New	•	🏫 Project					
🗄 🗁 bin	Open Type Hierarchy	F4	Annotation					
🗄 😁 🎥 SimpleEJB	Show In	Alt+Shift+W 🕨	Class					
	Сору	Ctrl+C	🕞 Enutits					
	Copy Qualified Name		🗊 Interface					
	💼 Paste	Ctrl+V	🖶 Package					
	💢 Delete	Delete	Example					
	Build Path	•						
	Source	Alt+Shift+S	Other					
	Refactor	Alt+Shift+1						
	🚵 Import							
	🛃 Export							
	🔊 Refresh	F5						

8. Populate the ApplicationClient.java with the following code

```
ApplicationClient.java
package appclient;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Properties;
import ejb.CountryCapital;
import javax.naming.Context;
import javax.naming.InitialContext;
public class ApplicationClient {
         public static void main(String [] args)
         {
                 String capital=new String();
                 try{
                         Properties prop=new Properties();
                         prop.put(Context.INITIAL_CONTEXT_FACTORY, "org.apache.openejb.client.
RemoteInitialContextFactory");
                                prop.put("java.naming.provider.url", "ejbd://localhost:4201");
         Context context = new InitialContext(prop);
         CountryCapital myejb =
         (CountryCapital)context.lookup("CountryCapitalBeanRemote");
         System.out.println("Give the name of a country");
         BufferedReader in= new BufferedReader(new InputStreamReader(System.in));
        String str="";
         str=in.readLine();
         capital=myejb.capitalName(str);
         System.out.println(capital);
```

```
}
catch(Exception e)
{
    e.printStackTrace();
}
}
```

Let us try to understand following code snippet

CodeSnippet Properties prop=new Properties(); prop.put(Context.INITIAL_CONTEXT_FACTORY, "org.apache.openejb.client.RemoteInitialContextFactory"); prop.put("java.naming.provider.url", "ejbd://localhost:4201"); Context context = new InitialContext(prop); CountryCapital myejb =(CountryCapital)context.lookup("CountryCapitalBeanRemote");

Useful Information

 \oslash

The above code suggests that which InitialContextFactory to be used to create the InitialContext. Here we are using **org.apache**. **openejb.client.RemoteInitialContextFactory** to create the InitialContext. In this case as the EJB server is the naming service provider and has naming service runnning at Port 4201. So we need to specify the location of the EJB server. In this case it is ejbd://localhost:4201. Once this is done we have the context that provides us the ability to lookup and get objects.

Why is the lookup name CountryCapitalBeanRemote?? :(

This will be discussed in deploy and run section.

- 9. Next step is to add org.apache.openejb.client.RemoteInitialContextFactory Class to the build path. This is because the application client needs this Class to create the InitialContext as explained above.
- 10. Right click on ApplicationClient project under project explorer. Select Properties.



11. Select Java Build Path. Under Java Build Path select Libraries-> Add External JARs.



12. Browse to <GERONIMO_HOME>\repository\org\apache\openejb\openejb-client\3.0-beta-2 and Select the jar openejb-client-3.0-beta-2.jar. Select Ok.

This completes the Application Client Development.

Deploy and Run

This section will take you through the deployment of the EJB application. Later we will run the application client with some sample inputs.

1. To deploy the EJB application on to server Export the EJB application as a jar file. Right Click on SimpleEJB application and Select Export->EJB JAR File as shown in the figure.

😂 Java EE	- Application(lient/src	/appclie	ent/App	licat	tionClient	t.java - Ec	
File Edit So	ource Refactor	Navigate	Search	Project	Run	Window	Help	
i 📬 • 🖫	≙ : ☆ • (> - 💊	• 1 🖻		5	S 1	2 1 😕	
Project Ex	oplorer 🔀		- 0	🕖 App	licatio	nClient.jav	a 🖾 【	
		E	\$₽^ \	Θ		public	static	
😑 🔂 Appli	cationClient					{	ring or	
🗎 🗎 🕮 s	rc					50. tr	ring car m/	
<u> </u>	appclient					01	Proper	
	±	Client.java					prop.r	
🗉 🗎 🗄 🗎	RE System Library	y [Java50]					pro	
🙂 📸 o	ipenejb-client-3.0	-beta-2.jar	- C: (AG2			Contex	t conte;	
	New				•	Country	yCapita]	
	Show In		Alti	chift rw	<u> </u>	(Count:	ryCapita	
	Show In		AILT	Shirt+w	_	System	.out.pri	
	Сору		Ctrl	+C		Buffer	edReadeı	
	Copy Qualifie	ed Name				String	str="";	
	Paste		Ctrl	+V		str=in	.readL11	
	X Delete		Delete			Sustem	out pri	
	Build Path	Build Path			•	3		
	Refactor	Alt+	Alt+Shift+T		catch (Exc			
	Import				-	{		
	Export						file	
					-	2		
	🌮 Refresh		F5		Ľ	Export	· F	
	Close Project	t						
	Close Unrela	ted Projects	S)C	he Geronim	o v2.1 Serve	
	Validate							
	Run As							
	Debug As							
	Profile As							
2	Team				•			

2. Browse to the Destination of your choice and Select Finish. This will export the SimpleEJB application as a jar file. This jar will be later deployed on to Geronimo server

() Why to Export EJB? Why cannot I use Eclipse for deploying the EJB application on the server? Due to some limitation Eclipse is not able to deploy the EJB application on to server. This issue will be fixed very soon.

3. Start the server and Launch the Administrative Console. Under Applications, Select Deploy New and Deploy the SimpleEJB.jar on the server.

4. Once the Deployment is successful. Open <GERONIMO_HOME>/var/log/server.log. Find the following in server.log

Snippet from server.log after deployment of SimpleEJB

18:16:39,750 INFO [startup] Jndi(name=CountryCapitalBeanLocal) --> Ejb(deployment-id=SimpleEJB /CountryCapitalBean) 18:16:39,750 INFO [startup] Jndi(name=CountryCapitalBeanRemote) --> Ejb(deployment-id=SimpleEJB /CountryCapitalBean) As can be seen openEJB container after deployment suggests the Remote as well as Local lookup names.
In this case since the SimpleEJB application and Java Application Client are running on different instances of JVM. We need to have the CountryCapitalBeanRemote as the lookup name.
Under Eclipse Right Click on the Application Client project and Run as a Java application.

Sample Input1: India

<					
🚼 Problems	🧟 Tasks	Properties	की Servers	🙀 Data Source	e Explorer [
<terminated> A</terminated>	pplicationCli	ent [Java Applic	ation] C:\Pro	gram Files\IBM\Ja	ava50\bin\ja
Give the n India New Delhi	name of	a country			
<					

Sample Input2: China

}	
	>
🖹 Problems 🖉 Tasks 🔲 Properties 👭 Servers 🙀 Data Source Explorer 🔂 S	nippets 📃
<terminated> ApplicationClient [Java Application] C:\Program Files\IBM\Java50\bin\javaw</terminated>	.exe (Fel 📄
Give the name of a country	
China	
Beijing	
I	
<u><</u>	

Sample Input3: United States Of America

Problems	🖉 Tasks	Properties	将 Servers	随 Data Source	e Explorer	🕒 Snippets	Con
<terminated> A</terminated>	pplicationC	lient [Java Applic	ation] C:\Pro	gram Files\IBM\Ja	ava50\bin\j	avaw.exe (Fe	ж
Give the r United Sta Washingtor	name of ates Of n DC	a country America					

Sample Input4: France



<