IBM WEBSPHERE BUSINESS MONITOR V6.0.2 - LAB EXERCISE

Monitoring using the sample DB2[®] Emitter

What this exercise is about	2
Lab requirements	2
What you should be able to do	2
Introduction	3
Exercise instructions	4
Part 1: Modify code to handle CustomerOrder event using the CEI API SDK and export the DBEmitter application (EAR)	5
Part 2: Create a DB2 emitter database	20
Part 3: Configure data source for DB2 emitter database	25
Dort 4: Configure a schedular convice and deploy the DP2Emitter EAP	20
Fait 4. Configure a scheduler service and deploy the DB2Enniter EAR	20
Part 5: Testing DB2 emitter	
Part 4: Configure a scheduler service and deploy the DB2Emiller EAR Part 5: Testing DB2 emitter Part 6: Solution	36 40
Part 4: Conligure a scheduler service and deploy the DB2Emitter EAR Part 5: Testing DB2 emitter Part 6: Solution What you did in this exercise	36 40 42

What this exercise is about

This exercise demonstrates using IBM WebSphere[®] Integration Developer to update the DB2[®] Emitter source code to customize it to emit specific user defined events using the CEI SDK that will be monitored using WebSphere Business Monitor. You will customize the supplied DB2 emitter source code, export the EAR, and deploy the EAR to WebSphere Process Server V6.0.2. You will configure WebSphere Process Server V6.0.2 with a Data Source for the new JDBC Provider, and a scheduler. Finally you will conduct a simple test to see if the events are being emitted.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Integration Developer V6.0.2
- WebSphere Process Server V6.0.2 or WebSphere Application Server V6.1 with CEI API SDK

What you should be able to do

At the end of this lab you should be able to:

- Use WebSphere Integration Developer to import the DB2 Emitter source with CEI SDK emitter transport and perform necessary modifications to it, build the DB2 Emitter project and finally export an EAR file for deployment
- Use the DB2 Command Line Window to create a new database which will be accessed by the DB2 Emitter Application, run the DDL scripts to create the necessary tables and triggers.
- Use the WebSphere Process Server administrative console to configure the JDBC Provider, a Data source, a cloudscape database for the scheduler and finally the Scheduler itself
- Use the WebSphere Process Server administrative console to install the EAR
- Run a simple test to view successful events emitted
- View the events emitted using the Common Event Browser (CBE Browser)

Introduction

The Sample DB2 Event Emitter is a sample program written in Java[™] that demonstrates how an enterprise information system (EIS) resource (an IBM DB2 database in this sample) which stores data pertaining to the state of a business can be instrumented to contribute to the overall monitoring of the activities of a business.

The main goal of the Sample DB2 Event Emitter is to introduce the use of the libraries and APIs provided by the Common Event Infrastructure (CEI) to generate and emit business events in the form of Common Base Events (CBEs). Common Base Events are the data packaging and format used by the WebSphere Business Monitor (WBM) Server to propagate business events.



Exercise instructions

Some instructions in this lab may be Windows[®] operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands, and use appropriate files (.sh vs. .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference variable	Windows location	AIX [®] /UNIX [®] location
<wid_home></wid_home>	C:\WID602	
<wps_home></wps_home>	C:\IBM\WebSphere\ProcServer	
<wps_profile_dir></wps_profile_dir>	<wps_home>\profiles\<profile_name></profile_name></wps_home>	
<labfiles></labfiles>	C:\Labfiles602	/tmp/Labfiles602

Important: This LAB does not cover the **Customer Order** Monitor Model deployment to the WebSphere Business Monitor Server. You should have first hand experience deploying and running a Monitor Model on the Monitor Server.

Important: You can either use WebSphere Process Server V6.0.2 (at times designated as Process Server) or a WebSphere Application Server V6.1 with CEI API SDK installed. This LAB uses WebSphere Process Server V6.0.2.

Important: The source code modification is a demonstration on how the events emitted will be correlated with the Monitor Model. Each event to be consumed by WebSphere Business Monitor needs to be matched to the Monitor Model.

Part 1: Modify code to handle CustomerOrder event using the CEI API SDK and export the DBEmitter application (EAR)

- 1. Start WebSphere Integration Developer, creating a new workspace in the folder C:\ C:\LabFiles602\DB2Emitter\workspace, and turn off the auto-build feature
 - ____a. The workspace Launcher window will be displayed. Click the **Browse...** button and select your workspace directory.

Workspace Launcher
Select a workspace
IBM WebSphere Integration Developer stores your projects in a directory called a workspace. Select the workspace directory to use for this session.
Workspace: C:\LabFiles602\DB2Emitter\workspace Browse
\Box Use this as the default and do not ask again
OK Cancel

- ___ b. Click **OK**
- ____ c. Close the welcome window by clicking the arrow in the top right corner of the welcome window
- 2. Switch to the J2EE perspective and turn off automatic builds
 - ____a. By default WebSphere Integration Developer opens in **Business Integration** perspective. You

need to change it to **J2EE** perspective. To do this click Business Inte... on the top right corner of the WebSphere Integration Developer and choose **Other** if **J2EE** is not listed here.

\int	😭 😨 Business Inte	»
	• 😨 Business Integration	
	<u>O</u> ther	

____b. From the Select Perspective dialog select J2EE and click OK

🚯 Select Perspective 🛛 🗙
Business Integration (default) CVS Repository Exploring Data Debug Generic Log Adapter DEF Data Debug Generic Log Adapter DEF Data Debug D
Show all
OK Cancel

Note: Select the Show all check box if J2EE perspective does not show up.

____ c. Click **OK** on the confirm enablement warning.

🚯 Confirm Enablement 🛛 🔀
This action requires the enablement of "Base J2EE Support". Enable the required capability?
Always enable capabilities and don't ask me again.
OK Cancel <u>D</u> etails >>

_____d. Select **Project > Build Automatically**, to turn off automatic builds.

╊ J2EE - IBM WebSphere Integration Developer				
<u>File E</u> dit <u>N</u> avigate Se <u>a</u> rch	Project	<u>R</u> un	<u>W</u> indow	<u>H</u> elp
	0 9	pen Pri ose Pri	oject oject	
Project Explorer 🔀	B:	uild <u>A</u> ll uild Pro	iert	Ctrl+B
Enterprise Applications Application Client Proje	Bu	uild <u>W</u> o lea <u>n</u>	rking Set	•
Enclose Connector Projects	🖌 Bi	uild Aut	o <u>m</u> atically	
Do Projects Dynamic Web Projects	G	enerati	e Javadoc	
Hand Web Services	Đ	opertie	es	
Databases Database Servers				

3. Import the **DB2Emitter** Projects

In this step, import all DB2 Emitter projects; make necessary build configurations based on the intended emitter transport (CEI API). Also import the necessary jars to support CEI API. Following is a description of the projects that will be imported (part of the Project Interchange **DB2Emitter.zip**):

- DBEmitter: Project to package artifacts into EAR
- EmitterFW: Common emitter framework source
- **CEIEmitter** : CEI emitter source
- **DBEmitterEJB:** DB2 Emitter specific source
- **DBEmitterImpl:** DB2 Emitter specific source. Implementation classes specific to retrieving and formatting events
- ____a. Select File > Import from the main menu.
- ____b. From the Import window, select Project Interchange as the source and click Next
- ____ c. The Import Project Interchange Contents window will be opened.
- ____d. Now click the first **Browse...** button and select **<LABFILES>\DB2Emitter\DB2Emitter_PI.zip** as the source .zip file

For example: C:\LabFiles602\DB2Emitter\DB2Emitter.zip

👍 Import Project I	nterchange Contents	X
Import Projects		
Import Projects from	a zip file.	J.
From zip file:	C:\LabFiles602\DB2Emitter\DB2Emitter_PI.zip	Browse
Project location root:	C:\LabFiles602\DB2Emitter\workspace	Browse
CEIEmitter	ct All Select Referenced	
	< Back Next > Finish	Cancel

- ____e. Select all the projects and then click Finish
- ____ 4. Review the imported artifacts
 - ____a. Expand Enterprise Applications, EJB Projects and Other Projects to view the projects as shown below



____b. At this time there will be some errors noticed. Click on the **Problems** tab to view the problems as shown below

	Problems × Tasks Properties Servers Console	≍ ‡ ▼	- 8
1 e	rror, 0 warnings, 0 infos		
	Description	Resource	In Fol
8	Project CEIEmitter is missing required library: 'C:BAM Emitter Samples/SamplesEmitterFramework/lib/events-client.jar'	CEIEmitter	
•			

- _____5. Working towards avoiding the build path errors
 - ____a. It clearly states that the CEIEmitter project is missing a required library **events-client.jar**. Import it as J2EE Utility JAR by right-clicking on the **DBEmitter** project from the project explorer as shown below

🟫 Project Explore	×	
	1 <u>6</u> = 🔄 🕶	
	Applications	
E- 29 DBEmit	New	•
EJB Projec	Paste	
E-C Other Proj	X Delete	
⊡ 🔐 CEIEm ⊡ 🔁 DBEmit	Refac <u>t</u> or Alt+Shift+T	•
	Import	App Client JAR file
⊕ - 🕞 Web Servi	Export	🕨 🌒 RAR file
	<u>B</u> uild Project	💊 EJB JAR file
	🔗 Re <u>f</u> resh	🗔 WAR file
	Clo <u>s</u> e Project	🥛 J2EE Utility Jar
	<u>R</u> un Validation Remove Compatibility	🚵 Import

____b. In the Utility Jar Import dialog select the type of Import as Copy Utility Jars into an existing EAR from an external location as shown below

tility Jar Import Import dependent	Utility Jars for J.	2EE Enterprise Ap	oplications	Ē,
-Select EAR Projec	t			
EAR Project: DB	Emitter			_
-Select import type				
🔿 Create Java P	rojects from Uti	lity Jars		
Create linked :	Java Projects fr	om Utility Jars		
📀 Copy Utility Ja	ırs into an existi	ng EAR from an e	xternal location	
C Create Linked	Utility Jars in an	n existing EAR fro	m an external lo	ocation
-Project import opt	ions ———			
🔲 Create Binary	Projects			
🔲 Override Proje	ct Root (Specify	y location below)		
Project locations	;			
This is the paren	it directory for a	all referenced pro	jects. A subdire	ctory will be cr
Module Root Loo	ation: C:\Proj	ectScenario\DB20	EIworkspace	Browse

___ c. Click Next

____d. In the next dialog click **Browse** to locate the **events-client.jar** and make sure you select the jar as shown below

4) ×
Utility Jar Import Import dependent Utility Jars for J2EE Enterprise Applications
Available Utility Jars Directory External Jar Directory: C:\WID602\runtimes\bi_v6\CEI\client Linked Path Variable Make these imported files team-shareable by creating or re-using a Linked
Path Variable (LPV). Select an existing LPV or type a new name to create one to the directory above. Other teammates should define this value
Utility JARs and web libraries Select the utility JARs from the list to be imported as utility projects or jars.
C:\WID602\runtimes\bi_v6\CEI\client\commons-jxpath.jar C:\WID602\runtimes\bi_v6\CEI\client\events-client.jar C:\WID602\runtimes\bi_v6\CEI\client\events-messages.jar
< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel

Note: You can find these libraries in WebSphere Process Server or WebSphere Integration Developer as follows:

- <WID_HOME>\runtimes\bi_V6\CEI\client or
- <WPS_HOME>\CEI\client

If CEI SDK is not installed on your machine local to the WebSphere Integration Developer, copy the necessary libraries from a remote WebSphere Process Server installation.

- ____e. Click Finish
- _____f. In the Project Explorer, right-click on the CEIEmitter project under **Other Projects** and select properties
- ____g. In the **Properties for CEIEmitter** window, select Java Build Path in the left frame, select Libraries tab in the right frame

🚯 Properties for CEIEmitter	
 Properties for CEIEmitter Info BeanInfo Path Builders DADX Web Services Provider Java Build Path Java Compiler Javadoc Location Java JAR Dependencies Java Task Tags Project References Server Validation 	Build path entry is missing: C:/BAM Emittermpl Source Projects Libraries & Order and Export JARs and class folders on the build path: Image: Project Order of the second path Image: Project Order Order of the second path Image: Project Order of the second path Image: Pr
	Default output folder:
	CEIEmitter/bin Browse
	OK Cancel

- ____h. Select the events-client.jar from the JARs listed and click the Remove button to remove build path.
- ____ i. Now click the Add JARs button and select the events-client.jar (DBEmitter \rightarrow events-client.jar)



- ____j. Click OK over the JAR Selection pop-up window
- ____k. Now click OK over the Properties for CEIEmitter window
- ___I. There must not be any errors reflected in the Problems tab at this time

6. Now modify the DB2 Emitter specific code. The sample code that is provided is based on a CUSTOMER event to show you how to modify it to handle a different event type, CustomerOrder. You will update the **CUSTOMERFormatterImpl.java** which implements the EventFormatter class and the **CUSTOMEREventRetrieverImpl.java** which extends **AbstractEventRetriever.**

Note: Two text files named CUSTOMEREventRetrieverImpl.txt and CUSTOMERFormatterImpl.txt with lines of code that are to be modified are under <LABFILES>\DB2Emitter\src. It is easy to copy these lines of code and paste at the appropriate location.

- a. In the Project Explorer expand Other Projects > DBEmitterImpl > src > com.ibm.wbimonitor.samples.db2emitter.formatter and double click on CUSTOMERFormatterImpl.java
- ____ b. Set an extension name for the CommonBaseEvent that is being returned. Add the line event.setExtensionName("CustomerOrder"); after the Event factory is created.



____ c. Search for the following code segment

ExtendedDataElement id = event.addExtendedDataElement("ID");

id.setValuesAsInt(Integer.parseInt((String)retrieved));

And replace them with the following

ExtendedDataElement ordernumber = event.addExtendedDataElement("OrderNumber");

ordernumber.setValuesAsInt(Integer.parseInt((String)retrieved));

____d. Search for the following code segment

ExtendedDataElement id = event.addExtendedDataElement("ID");

id.setValuesAsInt(((ResultSet) retrieved).getInt("ID"));

ExtendedDataElement name = event.addExtendedDataElement("NAME");

name.setValuesAsString(((ResultSet) retrieved).getString("NAME")); ExtendedDataElement address = event.addExtendedDataElement("ADDRESS"); address.setValuesAsString(((ResultSet) retrieved).getString("ADDRESS")); ExtendedDataElement tel = event.addExtendedDataElement("TEL"); tel.setValuesAsString(((ResultSet) retrieved).getString("TEL")); ExtendedDataElement email = event.addExtendedDataElement("EMAIL"); email.setValuesAsString(((ResultSet) retrieved).getString("EMAIL")); And replace them with the following ExtendedDataElement ordernumber = event.addExtendedDataElement("OrderNumber"); ordernumber.setValuesAsInt(((ResultSet) retrieved).getInt("OrderNumber")); ExtendedDataElement customername = event.addExtendedDataElement("CustomerName"); customername.setValuesAsString(((ResultSet) retrieved).getString("CustomerName")); ExtendedDataElement country = event.addExtendedDataElement("Country"); country.setValuesAsString(((ResultSet) retrieved).getString("Country")); ExtendedDataElement city = event.addExtendedDataElement("City"); city.setValuesAsString(((ResultSet) retrieved).getString("City")); ExtendedDataElement productnumber = event.addExtendedDataElement("ProductNumber"); productnumber.setValuesAsString(((ResultSet) retrieved).getString("ProductNumber")); ExtendedDataElement quantity= event.addExtendedDataElement("Quantity"); quantity.setValuesAsInt(((ResultSet) retrieved).getInt("Quantity")); ExtendedDataElement orderprice= event.addExtendedDataElement("OrderPrice");

orderprice.setValuesAsFloat(((ResultSet) retrieved).getFloat("OrderPrice"));

```
if( retrieved instanceof String ) {
    ExtendedDataElement id = event.addExtendedDataElement( "OrderNumber" );
    id.setValuesAsInt( Integer.parseInt( (String)retrieved ) );
3
else {
  ExtendedDataElement ordernumber = event.addExtendedDataElement("OrderNumber");
  ordernumber.setValuesAsInt(((ResultSet) retrieved).getInt("OrderNumber"));
  ExtendedDataElement customername = event.addExtendedDataElement("CustomerName");
  customername.setValuesAsString((((ResultSet) retrieved).getString("CustomerName"));
  ExtendedDataElement country = event.addExtendedDataElement("Country");
  country.setValuesAsString(((ResultSet) retrieved).getString("Country"));
  ExtendedDataElement city = event.addExtendedDataElement("City");
  city.setValuesAsString(((ResultSet) retrieved).getString("City"));
  ExtendedDataElement productnumber = event.addExtendedDataElement("ProductNumber");
  productnumber.setValuesAsString(((ResultSet) retrieved).getString("ProductNumber")
  ExtendedDataElement quantity= event.addExtendedDataElement("Quantity");
  quantity.setValuesAsInt(((ResultSet) retrieved).getInt("Quantity"));
  ExtendedDataElement orderprice= event.addExtendedDataElement("OrderPrice");
  orderprice.setValuesAsFloat(((ResultSet) retrieved).getFloat("OrderPrice"));
}
```

- ____e. Save (Ctrl+S) and close the editor
- f. In the Project Explorer expand Other Projects > DBEmitterImpl > src > com.ibm.wbimonitor.samples.db2emitter.retriever and double click on CUSTOMEREventRetrieverImpl.java
- ____ g. Search for the following code segment

private static String CREATE_EVENT_SQL = "SELECT * FROM CUSTOMER WHERE ID = ?";

private static String UPDATE_EVENT_SQL = "SELECT * FROM CUSTOMER WHERE ID = ?";

And replace them with the following

private static String CREATE_EVENT_SQL = "SELECT * FROM CUSTOMERORDER WHERE ORDERNUMBER = ?";

private static String UPDATE_EVENT_SQL = "SELECT * FROM CUSTOMERORDER WHERE ORDERNUMBER = ?";

- ___h. Save (Ctrl+S) and close the editor
- _____i. Modify DB2EventFormatter.properties. In the Project Explorer expand Other Projects > DBEmitterImpl > src and double click to open DB2EventFormatter.properties
- _____j. The relationship between an event type and the EventFormatter class for the type is specified in this property file so that the Event Formatter registers to the Event Type CustomerOrderTEST. For example specify the event type as CustomerOrderTEST to be more specific to this scenario

😤 Project Explorer 🛛 🗖 🗖	CUSTOMERFormatterImpl.java	CUSTOMEREventRetrieverImpl.java Z*DB2EventFormatter.properties
	Line 32 Column 3 +12 # warranty of any kind # EITHER EXPRESS OR IM # WARRANTIES OF MERCHA # Some jurisdictions d	1 Insert 5 changes -+3
CE Projects DefiniterEJB Other Projects CEIEmitter DefiniterImpl DefiniterImpl DefiniterImpl DefiniterImpl	<pre># implied warranties, # apply to you. IBM s # a result of using, m # its derivatives. #</pre>	so the above limitations or exclusions may not hall not be liable for any damages you suffer as odifying or distributing the sample program or
com.ibm.wbimonitor.samples.dt CUSTOMERFormatterImpl.j GOUSTOMERFormatterImpl.i GOUSTOMERFormatterImpl.i GOUSTOMERFormatterImpl.i GOUSTOMEREventRetrie GOUSTOMEREventRetrie META-INF MebSphere v6.0 Runtime	<pre># # # This properties file # The accepted format # EventType=EventF # Where the EventFormat # #EventType=EventFormat CustomerOrderTEST=com. </pre>	defines the formatter to use for a given event type. is: ormatter tter is provided as the fully qualified Java class. ter ibm.wbimonitor.samples.db2emitter.formatter.CUSTOMERFormatter

_ k. Modify the event type from **CUSTOMER** to **CustomerOrderTEST** and leave the EventFormatter class

(com.ibm.wbimonitor.samples.db2emitter.formatter.CUSTOMERFormatterImpl)
as it is.

- ___ I. Save (Ctrl+S) and close the file
- ____m. If workspace errors are not removed, it may be necessary to Refresh the projects and clean by clicking on the **Project > Clean** and then choose "**Clean All Projects**" in the dialog

🔂 Clean?	×
Clean will discard all build problems and built states. The next time a build occurs th will be rebuilt from scratch.	ie projects
Clean all projects	
C Clean selected projects: <a>No projects selected>	Bro <u>w</u> se
Start a build immediately OK	Cancel

_ n. At this time the following error might show up in the Problems tab

8	Problems 🕅 Tasks Properties Servers Console	(🖆 🖛 🗖
1 er	ror, 0 warnings, 0 infos	
	Description	Resource
8	The project cannot be built until its prerequisite DBEmitterEJB is built. Cleaning and building all projects is recommended	DBEmitterImpl

___ o. As the DBEmitterImpl project is dependent on DBEmitterEJB, build the DBEmitterEJB project by right clicking on it then selecting **Build Project**

- ____p. Select Project > Build All
- 7. Review the EJB Deployment Descriptor
 - ____a. In the Project Explorer, expand EJB Projects > DBEmitterEJB and double click on the Deployment Descriptor : DBEmitterEJB to open it in an editor

_ b. In the EJB Deployment Descriptor Editor, ensure that the **Source** tab is selected



__ c. The following are the default values specified in the EJB deployment descriptor. These can be customized by editing the source projects and exporting a new EAR archive.

<env-entry-name>emitterFactoryJNDI</env-entry-name>
<env-entry-value>iiop://localhost:2809/com/ibm/events/configuration/emitter/Default</env-entry-value>

Note: The default emitter factory specified for the "**emitterFactoryJNDI**" environment entry uses port 2809. Check your server's BOOTSTRAP_ADDRESS to ensure that is the correct port to use. Otherwise, import the source projects into WebSphere Integration Developer and make the necessary modifications to the EJB deployment descriptor.

<env-entry-name>dataSourceJNDI</env-entry-name>
<env-entry-value>jdbc/db2emitter</env-entry-value>

Note: The JNDI name for the data source "dataSourceJNDI" environment entry is jdbc/db2emitter. Configure the DB2 Emitter data source with this JNDI name.

<env-entry-name>schedulerJNDI</env-entry-name>
<env-entry-value>sched/DB2Poller</env-entry-value>

Note: The JNDI name for the scheduler "schedulerJNDI" environment entry is sched/DB2Poller. Configure the Scheduler with this JNDI name.

<env-entry></env-entry>
<pre><description>The JNDI name for the emitter factory to use for event emission</description></pre>
<pre> <env-entry-name>emitterFactoryJNDI</env-entry-name></pre>
<env-entry-type>java.lang.String</env-entry-type>
<env-entry-value>iiop://localhost:2809/com/ibm/events/configuration/emitter/Default</env-entry-value>
<env-entry></env-entry>
<description>The JNDI name for the database data source</description>
<env-entry-name>dataSourceJNDI</env-entry-name>
<env-entry-type>java.lang.String</env-entry-type>
<env-entry-value>jdbc/db2emitter</env-entry-value>
<session id="TaskScheduler"></session>
<ejb-name>TaskScheduler</ejb-name>
<home>com.ibm.websphere.startupservice.AppStartUpHome</home>
<remote>com.ibm.websphere.startupservice.AppStartUp</remote>
<pjb-class>com.ibm.wbimonitor.samples.db2emitter.TaskSchedulerBean</pjb-class>
<session-type>Stateless</session-type>
<transaction-type>Container</transaction-type>
<env-entry></env-entry>
<pre><description>The JNDI name for the scheduler</description></pre>
<env-entry-name>schedulerJNDI</env-entry-name>
<env-entry-type>java.lang.String</env-entry-type>
<env-entry-value>sched/DB2Poller</env-entry-value>

____ d. Close the EJB Deployment Descriptor Editor.

- ____ 8. Export the **DBEmitter** ear file.
 - ____a. Select **DBEmitter** project in the Enterprise Applications.
 - ____b. Right-click on the DBEmitter project to open the context menu. Select Export > EAR file



____ c. In the Export dialog, select DBEmitter for EAR Project and Browse to choose a Destination for the EAR.

🚯 Export		×
EAR Export Export Enter	prise Application project to the local file system.	
EAR project:	DBEmitter	•
Destination:	C:\LabFiles602\DB2Emitter\DBEmitter.ear	Browse
Export so	urce files	
🗌 Overwrite	existing file	
🗌 Include pr	oject build paths and meta-data files	
If you sel depender this optio	ect this option, the exported EAR maintains project names and extences, which are useful for later importing the EAR into binary project n and later import the EAR, only binary projects are created.	rnal classpath ts. If you select
	Einish	Cancel

___ d. Press Finish button

Part 2: Create a DB2 emitter database

In this part, the DB2 Emitter database (**For example:** DBEMITT) is created, and three DDL scripts are ran to create an Application Table (**For example:** CustomerOrder), the Event and Log Tables, a procedure and three TRIGGER statements for CREATE, DELETE & UPDATE for the Application table (**For example:** CustomerOrder).

___1. Creating Application, Event and Log tables

Table 1: Application Table (CustomerOrder)

ORDERNUMBER	INT	PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1)
CUSTOMERNAME	VARCHAR	
COUNTRY	VARCHAR	
CITY	VARCHAR	
PRODUCTNUMBER	VARCHAR	
QUANTITY	INT	
ORDERPRICE	FLOAT	

Events happen here in the Application Table (**CustomerOrder**) with a record created, updated or deleted (**C/U/D**). Based on the state (**C/U/D**) of this record a trigger is invoked

Table 2: EVENTTABLE

SID	BIGINT	PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1
TYPE	VARCHAR	The type of event which the EventFormatterFactory knows.
TRIGGER	CHAR	The kind of trigger. (Create / Update / Delete)
KEYTYPE	VARCHAR	The type of key which the EventFormatter knows.
KEYVALUE	VARCHAR	The value of key which the EventFormatter knows
CREATETIMESTAMP	TIMESTAMP	Time when an event posted.

Events on the Application Table (**CustomerOrder**) are stored as records in the EVENTTABLE by triggers. The trigger inserts a record with the primary key and additional information to the EVENTTABLE. The additional information includes the trigger type (**C/U/D**), and event type, and the creation timestamp.

SID	BIGINT	The primary key.
ТҮРЕ	VARCHAR	The kind of event which the EventFormatterFactory knows.
TRIGGER	CHAR	The kind of trigger. (Create / Update / Delete)
KEYTYPE	VARCHAR	The type of key which the EventFormatter knows.
KEYVALUE	VARCHAR	The value of key which the EventFormatter knows
CREATETIMESTAMP	TIMESTAMP	Time when an event posted.
RESULT	CHAR	The result of emission (Success / Fail)
LOGTIMESTAMP	TIMESTAMP	Time when an event logged.

Table 3: LOGTABLE

The results of the event emission are logged into the Log Table.

2. Creating the TRIGGER Statement

In the TRIGGER Statement, a statement for copying necessary data from the **Application Table** (CUSTOMERORDER) and inserting a record populated by using the data into the EVENTTABLE is needed. The Type of the key column of the Application Table should be specified.



In this case, the TRIGGER statements for Create/Update/Delete would be as follows:

Example1. TRIGGER statement for **CUSTOMERORDER** table

-- CREATE TRIGGER statement

CREATE TRIGGER CUSTOMERCREATE AFTER INSERT ON **CUSTOMERORDER** REFERENCING NEW AS NEWROW FOR EACH ROW MODE DB2SQL BEGIN ATOMIC INSERT INTO EVENTTABLE(TYPE, TRIGGER, KEYTYPE, KEYVALUE, CREATETIMESTAMP) VALUES('CustomerOrderTEST', 'C', 'INT', RTRIM(CHAR(NEWROW. ORDERNUMBER)), CURRENT TIMESTAMP); END;

-- UPDATE TRIGGER statement

CREATE TRIGGER CUSTOMERUPDATE AFTER UPDATE ON **CUSTOMERORDER** REFERENCING OLD AS OLDROW FOR EACH ROW MODE DB2SQL BEGIN ATOMIC INSERT INTO

EVENTTABLE(TYPE, TRIGGER, KEYTYPE, KEYVALUE, CREATETIMESTAMP) VALUES('CustomerOrderTEST', 'U', 'INT', RTRIM(CHAR(OLDROW. ORDERNUMBER)), CURRENT TIMESTAMP); END;

-- DELETE TRIGGER statement

CREATE TRIGGER CUSTOMERDELETE AFTER DELETE ON **CUSTOMERORDER** REFERENCING OLD AS OLDROW FOR EACH ROW MODE DB2SQL BEGIN ATOMIC INSERT INTO EVENTTABLE(TYPE, TRIGGER, KEYTYPE, KEYVALUE, CREATETIMESTAMP) VALUES('CustomerOrderTEST', 'D', 'INT', RTRIM(CHAR(OLDROW. ORDERNUMBER)), CURRENT TIMESTAMP); END;

The **bold** description above represents customization points for you to implement your own triggers.

CUTOMERORDER : The Application table Name

CustomerOrderTEST : The event type which is registered to the EmitterFactory or known by EventFormatterFactory

C/U/D : The Trigger type (For example: For Creation event set 'C')

INT : The key type. In this case, ORDERNUMBER column is INT

RTRIM (CHAR (NEWROW. ORDERNUMBER): The key value. If the type of key column is not 'CHAR', you need to cast the key value to CHAR.

Note: The DDL scripts for creating the CUSTOMERORDER tables and the triggers are provided in **<LABFILES>\DB2Emitter\SetUpFiles**

- __ 3. Creating the DB2Emitter Database
 - ____a. Open DB2 CommandLine Processor (**Open start > Programs > IBM DB2 > Command Line** Tools > CommandLine Processor)
 - ____b. Create new database using the CommandLine Processor

For example: create database DBEMITT

🚥 DB2 CLP - db2setcp.bat DB2SETCP.BAT DB2.EXE	_ 🗆 🗡
prompt. For example: db2 => connect to sample db2 => bind sample.bnd	
For general help, type: ?. For command help, type: ? command, where command can be the first few keywords of a database manager command. For example: ? CATALOG DATABASE for help on the CATALOG DATABASE command ? CATALOG for help on all of the CATALOG commands.	
To exit db2 interactive mode, type QUIT at the command prompt. Outs: interactive mode, all commands must be prefixed with 'db2'. To list the current command option settings, type LIST COMMAND OPTIC	ide ONS.
For more detailed help, refer to the Online Reference Manual.	
db2 => create database DBEMITT DB200001 The CREATE DATABASE command completed successfully. db2 => _	

Note: If the DB2Emitter database is created on a remote machine, then catalog the database:

catalog tcpip node \${DB_NODE} remote \${DB_HOST} server 50000

catalog db DBEMITT as DBEMITT at node \${DB_NODE}

- ____ c. Close the CommandLine Processor
- ____d. Open DB2 Command Window (**Open start > Programs > IBM DB2 > Command Line Tools >** Command Window)
- ____e. Connect to **DBEMITT** database using the db2 administrator username and password

For example: db2 connect to DBEMITT user db2admin using xxxxxx (password)



____f. Run createCustomerTable.ddl

For example: db2 -tf C:\Labfiles602\DB2Emitter\SetUpFiles\createCustomerTable.ddl

🖼 DB2 CLP	- 🗆 🗵
C:\IBM\DB2\SQLLIB\BIN>db2 -tf C:\Labfiles602\DB2Emitter\SetUpFiles\createCus	tome 🔺
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.CUSTOMERORDER" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	
C:\IBM\DB2\SQLLIB\BIN>	-

Note: If the DDL scripts are being run for the first time, an error preceding every create table command will be seen. This is due to the DROP statements being run before the create statements. You can safely ignore these errors.

____g. Run createEventAndLogTable.ddl

For example: db2 -tf C:\Labfiles602\DB2Emitter\SetUpFiles\createEventAndLogTable.ddl

📾 DB2 CLP	×
C:\IBM\DB2\SQLLIB\BIN>db2	
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.EVENTTABLE" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.LOGTABLE" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.LOGPROC" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	-

___h. Run createCustomerTrigger.ddl

For example: db2 -tf C:\Labfiles602\DB2Emitter\SetUpFiles\createCustomerTrigger.ddl

🔤 DB2 CLP	l ×
C:\IBM\DB2\SQLLIB\BIN>db2 -tf C:\Labfiles602\DB2Emitter\SetUpFiles\createCustom	e 🔺
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.CUSTOMERCREATE" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.CUSTOMERUPDATE" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0204N "DB2ADMIN.CUSTOMERDELETE" is an undefined name. SQLSTATE=42704	
DB20000I The SQL command completed successfully.	_

____i. Close the DB2 Command Window

Part 3: Configure data source for DB2 emitter database

In this part, the Process Server is configured with a data source for the DBEmitter application to do the database transactions with the DBEMITT database, configure J2C Authentication data for the data source to authenticate with the DBEMITT database.

- _____1. Configure J2C Authentication
 - ____a. Log on to the Administrative Console
 - ____ b. In the left pane locate Security > Global security and in the right pane expand JAAS Configuration and click on J2C Authentication data under the Authentication category.
 - _____ c. In the following screen click **New** to create a new Authentication Alias
 - ____ d. Enter the following General properties:
 - 1) Alias : DB2EmitAlias
 - 2) User ID : db2admin (or the db2 user you are using)
 - 3) Password : **xxxxxx** (DB2 password you are using)
 - ____e. Click **OK** and **Save** to the Master Configuration.
- _____2. Create a new JDBC Provider
 - _____a. Locate **Resources > JDBC Providers** and select the radio button next to **Server** to ensure that the JDBC Provider is created at the server scope
 - ____b. Click **New** to create new JDBC Provider
 - ____ c. Set the following for the General Properties:
 - 1) Database type: DB2
 - 2) Provider type: DB2 Universal JDBC Driver Provider
 - 3) Implementation type: XA data source

IDBC providers	2 🗆
<u>JDBC providers</u> > New Choose a type of JDBC provider to create.	
Configuration	
General Properties Step 1: Select the database type DB2	-
Step 2: Select the provider type DB2 Universal JDBC Driver Provider	
Step 3: Select the implementation type XA data source	
Next Cancel	

- ___ d. Click Next
- ____e. In the following screen enter **DB2Emitter Provider (XA)** in Name field. Click **OK** and **Save** to the master configuration

New Delete					
Select	elect Name 🗘 Description 🗘				
	Cloudscape JDBC Provider	Cloudscape 51 embedded JDBC2- compliant Provider			
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)			
	DB2Emitter Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.			
	Event DB2 JDBC Provider	DB2 Universal JDBC Driver Provider (XA) for the Common Event Infrastructure			
	Scheduler Cloudscape JDBC Provider (XA)	Cloudscape 51 embedded JDBC2- compliant Provider			
Total 5					

- ____3. Create a new Data source. Now that a JDBC provider had been created, a data source data will be created
 - ____a. Locate Resources > JDBC Providers > DB2Emitter Provider (XA) and click on Data sources under Additional properties category

Additional Properties
Data sources
Data sources
(Version 4)

- ____b. Click on the **New** button to create a new data source
- ____ c. Enter the following General Properties:
 - 1) Name: DB2 Emitter DataSource
 - JNDI name: jdbc/db2emitter (Note, this is the default JNDI name specified as an environment entry in the EJB deployment descriptor)
 - 3) Disable "Use this Data Source in container managed persistence (CMP)"
 - 4) Description: JDBC DataSource for DB2 Sample Event Emitter
 - 5) Component-managed authentication alias: **{NodeName}/DB2EmitterAlias** (the name of the authentication alias previously created in Step 1)
 - 6) Database name: **DBEMITT** (the name of the database previously created in Step 5)
- ____ d. Click **OK** and **Save** to the master configuration
- ____e. Test the Connection to the **DBEMITT** database

Note: If test connection fails as follows then you must set a WebSphere variable:



Locate WebSphere variables in the administrative console under the Environment section.

Set **DB2UNIVERSAL_JDBC_DRIVER_PATH** in the name field and the DB2 library path in the value field (**For example:** C:\IBM\SQLLIB\java), then click Apply

The scope of this variable depends on the scope of the JDBC Provider you configured to.

4. The DataSource configuration for the DBEMITT database is complete

Part 4: Configure a scheduler service and deploy the DB2Emitter EAR

In this part, a Scheduler Service is created for the DB2Emitter to invoke the EmissonController at a specified interval.

Follow the steps below to create a cloudscape database for the scheduler service; configure a DataSource for the same and finally a scheduler for the DB2Emitter

- 1. Create a Cloudscape database for the Scheduler Service
 - ___a. Launch cview.bat
 - ___ b. Explore to <WPS_HOME>/cloudscape/bin/embedded and double-click cview.bat to launch the Cview window

2 <mark>7</mark> Cview		- 🗆 🗵
File Edit View Help		
System	System Connection Preferences Information Properties	
	OK Cancel H	elp

- ____ c. From the main menu, select File → New → Database
- ____d. In the name field, enter <WPS_HOME>/cloudscape/databases/SKDLR

Where as <WPS_HOME> is C:\IBM\WebSphere\ProcServer

For example: - C:\IBM\WebSphere\ProcServer\cloudscape\databases\SKDLR

New Database	×
Database Localization Encryption Imp	ort
Name: C:\/BM\/VebSphere\ProcServer\clou	udscape\databases\SKDLR
Director	′у
ок	Cancel Help

___e. Click OK

____f. Verify that the database is created and is listed on the left pane of the Cview window

2 <mark>7</mark> Cview	
File Edit View Help	
System C UBMWVebSphere/ProcServer/cloudscape/databases/SKDLR Tables C Views C Stored Statements C Triggers C Jar Files C Aliases	Action New Delete Database Statistics Properties SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL C:\UBM\WebSphere\ProcServer\cloudscape\databases\SKDLR SQL OK Cancel Help

- ____ g. Close the Cview window (File \rightarrow Exit)
- 2. Create a DataSource for the Scheduler Service
 - ____a. Launch the WebSphere Process Server Administrative Console and logon to it

For example: - http://localhost:9060/ibm/console/

____b. In the left pane, locate Resources → JDBC Providers, set the scope to Servers and click Apply

🚰 WebSphere Administrative Console -	Microsoft Internet Explorer	
<u>Eile Edit View Favorites Tools H</u> e	elp	27
Address 🙆 http://localhost:9060/ibm/conso	le/secure/logon.do	🔽 🄁 Go 🛛 Links 🌺
Welcome admin Logout Supp	oort Help	
 Welcome Guided Activities Servers Applications Resources JMS Providers JDBC Providers Resource Adapters Asynchronous beans Schedulers 	JDBC providers JDBC providers are used by the installed applications to access data from databases. Scope: Cell=aimcp097N01C, Node=aimcp097Node Cell : aimcp097N01C → C Node : aimcp097Node Node : aimcp097Node Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help	Help Field help The name of the server that is selected Page help More information about this page
Cache instances Object pool managers Fone	(Apply)	Local intranet

- ___ c. Click the **New** button to create a new JDBC Provider
 - 1) Set Cloudscape in database type
 - 2) Set Cloudscape JDBC Provider in provider type
 - 3) Set XA data source in implementation type

JDBC providers	? -
<u>JDBC providers</u> > New	
Choose a type of JDBC provider to create.	
Configuration	_
General Properties	
Step 1: Select the database type Cloudscape	
Step 2: Select the provider type	
Cloudscape JDBC Provider	
Step 3: Select the implementation type XA data source	
Next Cancel	

___ d. Click Next

- _____e. In the following screen, enter the **Name** filed to **Scheduler Cloudscape JDBC Provider (XA)**, Click **OK** and **Save** to the master configuration
- _____f. You must the new the Cloudscape JDBC Provider listed as shown below:

New Delete				
Select Name 🛟 Description 🗘				
	Cloudscape JDBC Provider	Cloudscape 51 embedded JDBC2- compliant Provider		
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)		
	Event DB2 JDBC Provider DB2 Universal JDBC Driver Provider (XA) for the Common Event Infrastructure			
Cloudscape 51 embedded JDBC2- (XA) Cloudscape 51 embedded JDBC2- compliant Provider				
Total 4				

- ____g. Now that a JDBC provider had been created, create a data source
- ___ h. Locate Resources > JDBC Providers > Scheduler Cloudscape JDBC Provider (XA) and click on Data sources under Additional properties

Additional Properties

- Data sources
- Data sources (Version 4)
- ____i. In the following screen, click the **New** button to create a new data source
- ____j. Under General properties, enter the following:

1) Name : SKDLR DataSource

- 2) JNDI name: jdbc/skdlr
- 3) Disable "Use this Data Source in container managed persistence (CMP)"
- 4) Description: JDBC DataSource for SKDLR Database
- 5) Database name: C:\IBM\WebSphere\ProcServer\cloudscape\databases\SKDLR

(<WPS_HOME>\cloudscape\databases\SKDLR)

- ____k. Click **OK** and **Save** to the master configuration
- ___I. Test the connection to the scheduler database and ensure that a successful message is resulted
- ____3. Create a scheduler for the DB2 Emitter:
 - _____a. Locate **Resources > Schedulers** and the set the scope to **Servers**

🖉 WebSphere Administrative Co	isole - Microsoft Internet Explorer
<u>File E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u> oc	ls Help 🥂
Address 🕘 http://localhost:9060/ibn	/console/secure/logon.do 🗾 🔁 Go 🛛 Links 🌺
Welcome admin Logout	Support Help
= Welcome	Schedulers
🗄 Guided Activities	Scheduler s
🗄 Servers	Schedulers are persistent and transactional timer services that can run business logic. Each scheduler ru
	independently and has a programming interface accessible from J2EE applications using the Java Namin Directory Interface (JNDI). You can also manage a scheduler using a Java Management Extensions (JMX
🖯 Resources	See the scheduler documentation in the information center for details on how to configure and use a sch
🗄 JMS Providers	Scope: Cell=aimcp097N01C, Node=aimcp097Node
JDBC Providers	C Cell : aimcp097N01C Scope specifies the level at which the resource definition is visible.
Resource Adapters	the scope settings help
🗄 Asynchronous beans	→ O Node : aimcp097Node
Schedulers	
🗄 Cache instances	Server : server1
Object pool managers	
🗄 Remote Artifacts	
Mail Providers	
é	la l

- ____b. Click **New** to create a new Scheduler configuration
- ____ c. In the Scheduler > New screen, enter the following information:
 - 1) Name: DB2Poller
 - 2) JNDI name: **sched/DB2Poller** (This is the default specified as an environment entry in the EJB deployment descriptor)
 - 3) Description: Scheduler for DB2 Sample Emitter's Event Table Poller
 - 4) Data source JNDI name: jdbc/skdlr (select from the drop down list)
 - 5) Table prefix: DB2EMTR_
 - 6) Poll interval: 30
 - 7) Work managers: DefaultWorkManager

General Properties	
* Scope	
cells:aimcp097N01C:nodes:aimc	:p097Node:servers:server1
* N	
* Name	
DD2Foller	
* JNDI name	
sched/DB2Poller	
Description	
Scheduler for DB2 Sample Emitter's Event Table Poller	
Category	
* Data source JNDI name	
Japa skalk	<u> </u>
Data source alias	
(none)	•
* Table prefix	
DB2EMTR_	
* Dell internal	
* Poli Interval	conde
30	
* Work managers	
DefaultWorkManager 💽	
🗖 Use administration roles	
Apply OK Reset Cancel]

____ d. Click OK and Save to the master configuration

____e. Select check box next to DB2Poller and click the Create Tables button

New Delete Verify tables Create tables Drop tables						
Select	elect Name 💸 🛛 JNDI name 🔅 🔹 Data source JNDI name 🔅 🛛 Table prefix 🔅 🛛 Poll interval 🔅 🛛 Work managers 🔅					
	AppScheduler	AppScheduler	jdbc/WPSDB WSCH_ 10 AppSchedulerWorkManager			
	DB2Poller sched/DB2Poller jdbc/skdlr DB2EMTR_ 30 DefaultWorkManager					
Total 2						

- _____f. Ensure that the tables are created successfully
- _____4. The Scheduler configuration is complete
- _____5. Deploy the DB2Emitter EAR file.

Note: Ensure that the EAR is copied to the file system local to the application server where you would like to deploy

- ____a. Logon to the Administration console. In the left pane, select **Applications >Install New Application**
- ____b. Click Browse to Specify Path on the Local file system in the right panel and click Next

🚈 WebSphere Administrative Console - Microsoft Internet Explorer	
<u>File Edit View Favorites Tools H</u> elp	1
Address 🙆 http://localhost:9060/ibm/console/secure/logon.do	Go Links »
Welcome admin Logout Support Help	IBM.
Welcome Enterprise Applications Enterprise for the application installation	Close page
	Field I For fie
Enterprise Applications Install New	select marke cursor
Application Specify path ScA Modules Monitor Models Remote file system	Page <u>More</u> <u>this p</u>
Immediate Specify path Immediate Browse	
E Security Securit	
Environment Context root Used only for standalone Web modules	
Integration Application (.war files)	
System administration Next Cancel	
Done	anet //

- ____ c. Progress through all the 6 steps of "Install New Application" screens, accepting the defaults
- ____d. In Step 6, the **Summary** screen, click the **Finish** button. The install will take several moments
- ____e. After the installation is complete, ensure that the message "Application DBEmitter installed successfully" is displayed.

Application DBEmitter installed successfully.

To start the application, first save changes to the master configuration.

Save to Master Configuration

To work with installed applications, click the "Manage Applications" button.

Manage Applications

IBM WebSphere Business Monitor 6.0.2 – Lab Exercise

____f. Click "Save to Master Configuration", and finally click Save on the "Save" screen.

Enterprise Applications



__6. Restart the Process Server

Part 5: Testing DB2 emitter

- 1. Now that the Process Server configuration is complete and DB2Emitter application is deployed, it is time to do a simple test for CEI event emissions.
 - ____a. Ensure that the Process Server and the DB2Emitter application are running.
 - ____ b. Open the DB2 Command Line Processor. (Start > Programs > IBM DB2 > Command Line Tools > Command Line Processor)
 - ____ c. Run connect to DBEMITT using the DB2 administrator user and password

For example: connect to DBEMITT user db2admin using xxxxxx (or the user/password being used)

____d. Run an insert statement for the CUSTOMERORDER table. For example: insert into CUSTOMERORDER(CUSTOMERNAME, COUNTRY, CITY, PRODUCTNUMBER, QUANTITY, ORDERPRICE) values('Jane Doe', 'USA', 'Austin', 'P123', 2, 2999.50)

📾 DB2 CLP - db2setcp.bat DB2SETCP.BAT DB2.EXE	_ 🗆 🗙
For general help, type: ?. For command help, type: ? command, where command can be the first few keywords of a database manager command. For example: ? CATALOG DATABASE for help on the CATALOG DATABASE command ? CATALOG for help on all of the CATALOG commands.	
To exit db2 interactive mode, type QUIT at the command prompt. Outside interactive mode, all commands must be prefixed with 'db2'. To list the current command option settings, type LIST COMMAND OPTIONS.	
For more detailed help, refer to the Online Reference Manual.	
db2 => connect to DBEMITT user db2admin using db2admin	
Database Connection Information	
Database server = DB2/NT 8.2.6 SQL authorization ID = DB2ADMIN Local database alias = DBEMITT	
db2 => insert into CUSTOMERORDER(CUSTOMERNAME, COUNTRY , CITY, PRODUCTNUMBE ANTITY, ORDERPRICE) values('Jane Doe', 'USA', 'Austin', 'P123', 2 , 2999.50 DB20000I The SQL command completed successfully. db2 => _	}, QU }> ▼

Note: Immediately after the insert statement is run, the inserted data is moved to 'EVENTTABLE' by the trigger. DB2Emitter refers to the 'EVENTTABLE' at regular intervals using the Scheduler.

The following is the CUSTOMERORDER table with a record inserted (created)

🏪 Open Table - C	Copen Table - CUSTOMERORDER								
AIMCP3X6 - DB2 - D	BEMITT - DB2ADMIN.C	USTOMERORDE	R						
ORDERNUMBER 🔶	CUSTOMERNAME 🖨	COUNTRY 😂	CITY ≑	PRODUCTNUMBER ⇔	QUANTITY 🔶	ORDERPRICE	Add Row		
	Jane Doe	USA	Austin	P123	2	2,999.5			

The result of the event emission is logged to the LOGTABLE as shown below:

AIMCP3X6 - DB2 - DBEMITT - DB2ADMINLOGTABLE SID \$ TYPE \$ KEYTYPE \$ KEYVALUE \$ CREATETIMESTAMP \$ RESULT \$ 1.OGTIMESTAMP \$ Add Row 1 Mar 24, 2006 10:20:5[S] Mar 24, 2006 10:20[S] Mar 24, 2006 10:20[S] Delete Row	🍢 Open Table - LOGTABLE						
SID \$ TYPE TRIGGER \$ KEYTYPE \$ KEYVALUE \$ CREATETIMESTAMP \$ RESULT \$ LOGTIMESTAMP \$ Add Row 1 Mar 24, 2006 10:20:5[S] Mar 24, 2006 10:20:5[S] Mar 24, 2006 10:20:5[S] Delete Row	AIMCP3X6 - DB2 - DBEMITT - DB2ADMIN.LOGTABLE						
1CustomerOrderTESTC INT 1 Mar 24, 2006 10:20:5S Mar 24, 2006 10:2		Add Row					
	UCustomerOrderTESTC INT 1 Mar 24, 2006 10:20:5IS Mar 24, 2006 10:2	Delete Row					

Note: If emission is successful, a successful record (RESULT='S') is inserted to 'LOGTABLE'

If an Exception occurs, a failed record (RESULT='F') is inserted to 'LOGTABLE'

If a Runtime Exception or an Error occurred, the record is rolled back

 e. Run an update statement for the CUSTOMERORDER table.
 (For example: update CUSTOMERORDER SET ORDERPRICE = 9999.90 where ORDERNUMBER = 1)



The following is the representation of data in CUSTOMERORDER and LOGTABLE after an update statement is processed.

🏪 Open Table - (USTOMERORDER						×
AIMCP3X6 - DB2 - I	DBEMITT - DB2ADMIN.(ER				
ORDERNUMBER≑	CUSTOMERNAME 🖨	COUNTRY ⇔	CITY 👙	PRODUCTNUMBER 😂	QUANTITY 👙	ORDERPRICE 🔶	Add Row
	Jane Doe	USA	Austin	P123	2	(9,999.9)	Delete Row

	Ope	n Table - LOGTABL	E						×
AIM	ICP3	X6 - DB2 - DBEMITT -	DB2ADMIN.LC	GTABLE					
SID) ⇔	TYPE ⇔	TRIGGER ≑	KEYTYPE 👙	KEYVALUE ≑	CREATETIMESTAMP ⇔	RESULT 🔶	LOGTIMESTAMP 🖨	<u>A</u> dd Row
	1	CustomerOrderTEST	С	INT	1	Mar 24, 2006 10:20:5	S	Mar 24, 2006 10:2	
	2	CustomerOrderTEST	U	INT	1)	Mar 24, 2006 11:00:2	S	Mar 24, 2006 11:0	Delete Row

_ f. Run a delete statement for the CUSTOMERORDER table. example: delete from CUSTOMERORDER where ORDERNUMBER = 1)

(For

📾 DB2 CLP - db2setcp.bat DB2SETCP.BAT DB2.EXE	_ 🗆 🗡
db2 => insert into CUSTOMERORDER(CUSTOMERNAME, COUNTRY, CITY, PRODUCTNUMBER	, QU 🔺
HNIIIY, URDERFRICE) values('Jane Doe', 'OSH', 'Hustin', 'P123', 2 , 2999.50) DB20000I The SQL command completed successfully.	
db2 => update CUSTOMERORDER SET ORDERPRICE = 9999.90 where ORDERNUMBER = 1	
DB200001 The SQL command completed successfully. db2 => db2 delete from CUSTOMERORDER uberg ORDERNUMBER = 1	
SQL0104N An unexpected token "db2" was found following "BEGIN-OF-STATEMENT"	_
Expected tokens may include: "SELECT". SQLSTATE=42601	
DB20000I The SQL command completed successfully.	
db2 =>	•

The following is the representation of data updated in the CUSTOMERORDER and LOGTABLE after a delete statement is issued.

₩Ор	en Table - Cl	JSTOMERO	RDER						×	<
AIMCR	AIMCP3X6 - DB2 - DBEMITT - DB2ADMIN.CUSTOMERORDER									
ORDE	RNUMBER 👙	CUSTOMER	NAME 🔶 📗	COUNTRY	\$	CITY	\$	PRODUCTNUMBEF	<u>A</u> dd Row	
									Delete Row	1
Ч. Оре	en Table - LOGTA	BLE							1	×
AIMCP3	3X6 - DB2 - DBEMIT	T - DB2ADMIN.L	OGTABLE							
SID 🖨	TYPE	⇔ TRIGGER⇔	KEYTYPE¢	KEYVALUE 🖨	CREAT	ETIMESTAMP⇔	RESULT 🖨		Add Row	1
1	CustomerOrderTES	sт с	INT	(1)	Mar 24	, 2006 10:20:5	S	Mar 24, 2006 10:21:05		1
2	CustomerOrderTES	ST U	INT	1	Mar 24	, 2006 11:00:2	S	Mar 24, 2006 11:00:26		
3	CustomerOrderTES	st D	INT	11	Mar 24	, 2006 11:21:4	S	Mar 24, 2006 11:21:50		

____g. To view the events emitted in a CBE browser, logon to the Process Server's Administrative console, Locate Integration Applications > Common Base Event Browser click on the Get Events and then All Events.

🚰 WebSphere Administrative Consol	le - Microsoft	Internet Explorer				_ 8 ×
<u>File Edit View Favorites Tools</u>	Help					1
🔇 Back 🔹 🕤 🕣 💌 😰 🐔 🔎 Se	earch 🛛 🔆 Fav	orites 🐵 🕼 🗧 🗧 -				
Address () http://localhost:9061/ibm/cor	nsole/secure/lo	jon.do			🔹 🛃 Go	Links »
Welcome AdminDu Logout	Support	Help			000	TEM.
1. CANCELLE 1. 1	163 12				000	
Welcome	WebSphere	software CBE Event Browser				1f
				AL OOL		=:=0
III Servers	Get Events	Help				
Applications	1		Soloct Action			<u> </u>
H Resources	Event Views][00]		
E Security	(011))	Select ^ Creation Time ^ Name ^	Priority ^ Severity ^	Server ^ Sut	o-component ^	Situa
Environment	Events	© 2006-03-28720:45:39.516Z CustomerOrde		db2s	yscs.exe	Availat
Integration Applications	BPEL	O <u>2006-03-28T20:51:25.156Z</u> CustomerOrde	<	db2s	yscs.exe	Availak
Failed Event Manager	Process Fuents	O 2006-03-28T20:53:10.359Z CustomerOrde		db2s	yscs.exe	Availak
Relationship Manager	User	Page 1 of 1 Total: 3 Filt	ered: 3 Displayed: 3 S	Selected: 1		
Browser	Data					-
System administration	Events					•
Monitoring and Tuning	Server Events	extendedDataElement / TRIGGER	REATE			
	30 - 36 	extendedDataElement / OrderNumber 1				
	Number of	extendedDataElement / CustomerName J	ine Doe			
El Service integration	events: 3	extendedDataElement / Country	SA			
H UDDI		extendedDataElement / City A	ustin			
		extendedDataElement / ProductNumber P	23			_
		extendedDataElement / Quantity 2				
		extendedDataElement / OrderPrice 2	999.5			
		reporterComponentId				
		sourceComponentId / component d	o2.exe			
		sourceComponentid / subComponent d	o2syscs.exe			
		sourceComponentid / componentidType d	o2syscs.exe			
		sourceComponentId / instanceId				
		sourceComponentId / application D	32			
		sourceComponentId / executionEnvironment				1
		source.omponentia / location	COS		Local interact	<u> </u>
					Local intrariet	

Part 6: Solution

If you would prefer not to create the emitter from scratch, then you can follow this section to import a project interchange file which has the updated source. Using the project interchange that is provided, you can import it to WebSphere Integration Developer, then Build All projects and export the EAR.

If using the solution, skip Part 1, complete this section, and then continue with Part 2.

- _____1. Open WebSphere Integration Developer
 - ____a. Select Start > Programs > IBM WebSphere > Integration Developer V6.0.2 > WebSphere Integration Developer V6.0.2
 - ____ b. Workspace Launcher window will be displayed. Click the Browse... button and select your workspace directory. (For example: C:\LabFiles602\DB2Emitter\workspace)
 - ___ c. Click **OK**.
 - _____d. Close the welcome window by clicking the arrow in the top right corner of the welcome window
 - ____e. By default WebSphere Integration Developer opens in **Business Integration** perspective. You need to change it to **J2EE** perspective. To do this click on the top right corner of the WebSphere Integration Developer and choose **Other** if **J2EE** is not listed here.
 - ____f. Select Project > Build Automatically, to turn off automatic builds
 - 2. Import the solution DB2Emitter project interchange into WebSphere Integration Developer
 - ____a. Select **File > Import** from the main menu.
 - ____b. From the Import window select Project Interchange and click Next
 - ____ c. The Import Project Interchange Contents window will be opened.
 - _____d. Click Browse... and select <LABFILES>\DB2Emitter\DB2Emitter_Solution_PI.zip as the source .zip file.

For example: C:\LabFilesCEM61\DB2Emitter\DB2Emitter_Solution_PI.zip

____e. Click Select All then click Finish

🚯 Import Project Ir	nterchange Contents	×
Import Projects Import Projects from	a zip file.	Ţ
From zip file: Proiect location root:	C:\LabFiles602\DB2Emitter\DB2Emitter_Solution_PI.zip	Browse
CEIEmitter		
Select All Deselect	t All Select Referenced Select Referenced < Back	Cancel

- _____f. Refresh the projects and clean by clicking on **Project** > **Clean** and then choose "**Clean All Projects**" in the dialog
- ___ g. Click Project > Build All to build all the projects
- ____h. At this point you might see an error in the Problems tab
- ____i. As the DBEmitterImpl project is dependent on DBEmitterEJB, build the DBEmitterEJB project by right clicking on it.
- ____j. Finally in the Project Explorer, right click on **DBEmitter** under Enterprise Applications to choose **Export > EAR file**.
- ____k. Provide the Destination you would like to save the EAR.
- ___I. Close WebSphere Integration Developer.
- ____ m. Continue the lab with **Part 2** of this document.

What you did in this exercise

You imported the sample DB2 Emitter source into WebSphere Integration Developer and modified it to match the DB2 Emitter's CUSTOMERORDER data. Then you exported the DBEmitter EAR to WebSphere Process Server V6.0.2 that has a CEI SDK installed.

You Deployed the Application to WebSphere Process Server V6.0.2

You created a new DataSource for the Db2 Emitter database.

You created a cloudscape database for the Scheduler, created a DataSource for it and then created a Scheduler service.

You updated records in the Application Table (CUSTOMERORDER) to test for the Event Emissions to the CEI server

Finally you used the Process Server's Common Event Browser to view the emitted events.