WebSphere® IBM Branch Transformation Toolkit for WebSphere Studio

IBM

Version 5.2

Migration guide from BTT version 5.1 to version 5.2

Note!
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 33.

This edition applies to Version 5, Release 2, Modification 0, of *IBM Branch Transformation Toolkit for WebSphere Studio* (5724-H82) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send to the following address:

IBM China Software Development Lab
Branch Transformation Toolkit Product
Diamond Building, ZhongGuanCun Software Park, Dongbeiwang West Road No.8,
ShangDi, Haidian District, Beijing 100094 P. R. China

Include the title and order number of this book, and the page number or topic related to your comment.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Migration guide from version 5.1 to version 5.2

The migration tool provided by Branch Transformation Toolkit version 5.2 helps users of earlier versions move up to the full power and versatility of the IBM® WebSphere® Application Server and IBM WebSphere Process server. It provides the migration path to allow an enterprise to reuse much of its existing toolkit application code base and still take advantage of the WebSphere J2EE environment.

IBM recognizes that an enterprise cannot always upgrade its application software all at once. Because the enterprise can decide how much or how little toolkit functionality to retain, version 5.2 allows the enterprise to upgrade to a full J2EE environment incrementally. If desired, with version 5.2, the enterprise can bypass toolkit functionality entirely and access J2EE components directly.

This tool is to migrate BTT 5.1 definition to BTT 5.2 base definition. It generates the corresponding program codes and the BTT version 5.2 tooling artifact. There are two ways for the migration:

- Single Migration: You can migrate the BTT 5.1 components one by one manually by using the migration tool. The migration tool will migrate the entities to BTT 5.2 such as dse.ini, data model, formatter, server operations, screen flow, process flow, and generate the artifacts for the tools of BTT 5.2 such as Graphical Builder, BPEL and Struts.
- Batch Migration: You can migrate a batch of components automatically using your own migration application. The batch migration application will migrate the entities and generate the related tooling artifacts. The process can only run on RAD and WID.

It provides you interfaces to help you migrate the BTT 5.1 application to BTT 5.2 one by one. It can also provide the capability for you to build your own batch migration application by using a set of definition files. Furthermore, the migration tool can delegate the migration process to the plug-in custom code to perform the customer unique extension.

This migration guide describes the migration considerations when you migration to version 5.2 and the migration tools that help you roll out the migrations. Note that the descriptions do not present all of the possible architecture variations available in each version.

Tasks

For the tasks related to migration from version 5.1 to version 5.2, see the following:

Migrating your applications to version 5.2

In BTT version 5.2, most of the usage and interfaces of the components are kept as same as BTT version 5.1. The flowing components need the migration work:

- Struts Extension: wsifAction in struts configuration file has been replaced by JSR109 enabled struts actions.
- Business Process Extension: The BPEL standard has been changed in WebSphere Process Server
- Service Infrastructure

Migration preparation

Before you start migrating your application from BTT version 5.1 to BTT version 5.2, you need to prepare the development environment for the migration:

- For the application without BPEL, you need to install Rational® Application Developer v6.0.1 or above
- For the application with BPEL, you need to install WebSphere Integration Developer v6.0.1 or above

1. Import your BTT version 5.1 application
2. Replace BTT version 5.1 CHA projects and the related jar files by those of BTT version 5.2. Here we take BTTHTMLSample migration as an example.
   a. Migrate your Enterprise Application from J2EE 1.3 to J2EE 1.4
   b. Remove the CHA related projects from the workspace, including BTTCHAEJB and BTTCHAWEB
   c. Import BTT version 5.2 CHA EJB JAR file into the workspace. The file name is BTTCHAEJB. The EAR project should be the project of the BTT version 5.1 application, for example, BTTHTMLSample.
   d. Copy `com.ibm.btt.cs.invoker.base.BeanInvokerRegistryMapper.properties` from `bttinvoker.jar` in the JavaSource folder of BTTHTMLSampleWeb project, because the properties file doesn't contain any invoker information for this sample in `bttinvoker.jar` of the BTT version 5.2 package.
   e. Replace BTT version 5.1 jar files in the Enterprise Application project and the Web project by BTT version 5.2 jar files.

      Note: `bttfmt.jar` and `bttsvcinfra.jar` are not needed in the BTT version 5.2 package. You can remove them.
   f. Add `bttbase.jar` and `bttOpStepAdaptor.jar` in the build path of BTTCHAEJB project
   g. Create a new database for BTT version 5.2 CHA, because the table schema of CHA in BTT version 5.2 is changed.
   h. Turn on the services of startup bean and the workarea from the administration console, because by default, these two services are disabled in WebSphere Application Server v6.x and WebSphere Process Server 6.x.

You may encounter some compilation error in the projects – BTTHTMLSampleEJB and BTTHTMLSampleWeb. They will be solved in "Migrate application code" on page 3

Now you can start the migration.

Migrate definition file

The HTMLSample is taken as an example to illustrate the migration process.

1. For the server side dse.ini file, some information needs to be changed. Add the following information:

```
<kColl id="settings">
........
........
<!--===================-->
<!--WorkAreaDefintioin-->
<!--===================-->
<kColl id="WorkArea">
        <field id="usingWorkArea" value="true"/>
        <field id="location" value="iiop://localhost:2809/"/>
        <field id="jndiName" value="java:comp/websphere/UserWorkArea"/>
```

```
</kColl>
<!--==============================-->
<!--For clustering consideration-->
<!--==============================-->
<kColl id="EJB">
        <field id="InitialContextFactory" value="com.ibm.websphere.naming.WsnInitialContextFa
        <field id="location" value="iiop://localhost:2809/"/>
</kColl>

<kColl id="cha-server">
<!--==========================================-->
<--      CHA Service Parameters             -->
<!--==========================================-->

........
<field id="EJBProviderURL" value="iiop://localhost:2809"/><!--new line -->
........
```

2. Some information is not necessary in BTT version 5.2. Remove the following information:

```
<kColl id="settings">
..
 <kColl id="CHAFormatterServiceClient">
   <field id="CHAFormatterServiceProxyClass"
value="com.ibm.btt.formatter.client.CHAFormatterServiceWSIFEjbProxy"/>
   <field id="WSIFEjbProxyWSDLPath"
value="c:/com/ibm/btt/formatter/server/CHAFormatterServiceEJBService.wsdl"/>
 </kColl>
 ..
</kColl>
   ..
```

Migrate application code

The HTMLSample is taken as an example to illustrate how to migrate the application code.

The migration of application code has two steps:

1. Solve the compilation error
2. Modify the definition for the presentation layer application

Solve the compilation error:

1. Error in BTTHTMLSampleEJB project: The compilation error will be in `SignInBean.java`:

```
public void ejbRemove()  {
  super.ejbRemove();
 }
public void ejbCreate() throws BTTSAEException {
  super.ebjCreate();
 }
```

   - For the error in `ejbRemove()`, you can remove the statement `super.ejbRemove()`, or you can remove this method directly
   - For the error in `ejbCreate()`, you can remove this method directly.

2. Error in BTTHTMLSampleWeb project: All the compilation errors in this project are related to the type mismatching in `throw new DSETypeException(DSETypeException.critical, msg, msg, null)`. You can solve the errors by changing the statement to `throw new DSETypeException(DSETypeException.critical, msg, msg, (String)null)`. Besides, there will also be compilation errors in `SignInBean.java`:

```
public void ejbRemove()  {
  super.ejbRemove();
 }
public void ejbCreate() throws BTTSAEException {
  super.ebjCreate();
 }
```

Modify the definition file for presentation layer application:

Put your short description here; used for first paragraph and abstract.

1. Remove `/jsp:useBean` statement from JSP files. These files are `Frameheader.jsp` and `creditCardsWelcome.jsp`
2. Add the following statement below into all of the Struts configuration files:
   ```
   <struts-config>
   ........
        <controller processorClass="com.ibm.btt.struts.base.BTTRequestProcessor"/>
   ```
3. Modify `web.xml`
   a. In `web.xml`, there is a path definition error in RAD/WID development environment:
      ```
      <init-param>
            <param-name>config</param-name>
            <param-value>WEB-INF/struts-config.xml</param-value>
      </init-param>
      ```
      Modify `<param-value>WEB-INF/struts-config.xml</param-value>` to `<param-value>/WEB-INF/struts-config.xml</param-value>`
   b. There two optional parameters in `web.xml`:
      - directmapping:
        ```
        <init-param>
              <param-name>directmapping</param-name>
              <param-value>ture</param-value>
        </init-param>
        ```
        This parameter is optional and the default is false if you don't specify it. But, if you specify the parameter and the value is true, you need to modify each of the formbean class name in the struts configuration file by the following statement:
        ```
        <form-beans>
          <form-bean name="paymentForm" type="com.ibm.btt.struts.base.BTTActionForm">
          </form-bean>
        </form-beans>
        ```
        Then the application will not prepare the formbean class and the BTT Struts Extension will handle the mapping from the http request data to the BTT context automatically.
      - tokenProcessor:
        ```
        <init-param>
              <param-name>tokenProcessor</param-name>
              <param-value>com.customer.TokenProcessor</param-value>
        </init-param>
        ```
      It is not necessary to define it in the `web.xml` file and the default value is `com.ibm.btt.struts.utils.BTTTokenProcessor`

Now you complete all the migration work for the application. You can deploy the application and execute the transaction in RAD 6.0.1 and WID 6.0.1

Example: migrate your JavaSample

Below is an example of how to migrate your JavaSample. Perform the following steps:

1. Remove BTTCHAEJB, BTTCHAEJBWEB, BTTFormatterEJB, BTTServicesInfraWeb, and bttsvcinfra.
2. Right click on BTTJavaSample and select **migrate** → **J2EE Migrate Wizard**, and click **Next** in the widow that pops up. Follow the wizard till you finish the migration of J2EE.
3. In DummyJournal.java, modify

   ```
   import com.ibm.btt.services.jdbcjournalservice.Journal;
   import com.ibm.btt.services.jdbcjournalservice.JournalService;
   import com.ibm.btt.services.jdbcservicesinfra.DSESQLException;
   ```

   to

   ```
   import com.ibm.btt.services.jdbc.DSESQLException;
   import com.ibm.btt.services.jdbc.journal.Journal;
   import com.ibm.btt.services.jdbc.journal.JournalService;
   ```
4. Click **BTTJavaSampleEJB** → **EJBmodule** → **com.ibm.btt.samples.services**, and delete DummyJournalImpl.java
5. Click **BTTJavaSampleWeb** → **WebContent** → **WEB-INF**, and delete the tld directory.
6. Replace server and client define files, including: dsedata.xml, dsectxt.xml, dsefmts.xml, dsesrvce.xml and dsetype.xml from BTT version 5.1 to BTT version 5.2.
7. Replace all jars in the BTT JavaSample and replace all jars in /BTTJavaSample/Web/WebContent/WEB-INF/lib
8. Modify ejb-jar.xml in the /BTTJavaSampleEJB/ejbModule/META-INF:
   a. Modify `<env-entry-value>StartupJavaSessionAction</env-entry-value>` to `<env-entry-value>startup</env-entry-value>`
   b. Modify `<env-entry-value>CustomerSearch</env-entry-value>` to `<env-entry-value>customerSearch</env-entry-value>`
   c. Modify `<env-entry-value>AccountStatement</env-entry-value>` to `<env-entry-value>accountStatement</env-entry-value>`
   d. Modify `<env-entry-value>Deposit</env-entry-value>` to `<env-entry-value>deposit</env-entry-value>`
   e. Modify `<env-entry-value>Withdrawal</env-entry-value>` to `<env-entry-value>withdrawal</env-entry-value>`
   f. Modify `<env-entry-value>EndSessionAction</env-entry-value>` to `<env-entry-value>endSessionAction</env-entry-value>`
9. Modify class in /BTTJavaSampleEJB/ejbModule/ com.ibm.btt.samples.appl.businessAccountStatement.java:

   Modify

   ```
   public Hashtable execute(BTTSystemData sysData, String uid)
     throws BTTSAEException, Exception, java.rmi.RemoteException;
   ```

   to

   ```
   public String execute(String request)
     throws BTTSAEException, Exception, java.rmi.RemoteException;
   ```

   In AccountStatementBean.java, remove `import com.ibm.btt.services.jdbcjournalservice.*;`, and modify:

   ```
   public class AccountStatementBean extends StatelessSingleAction
   ```

   to

   ```
   public class AccountStatementBean extends BaseJavaSampleBean
   ```

and remove `super.ejbCreate()` and `super.ejbRemove();`:

```
public void ejbCreate() throws BTTSAEException {
  super.ejbCreate();
 }
public void ejbRemove() {
  super.ejbRemove();
 }
```

Modify:

```
public Hashtable execute(BTTSystemData sysData, String uid) throws BTTSAEException, Exception {

  initialize(sysData);

  Hashtable result = new Hashtable();
  IndexedCollection ic = null;
  int i,j = 0;

  Context sessionCtx = null;

  //if instanceID != null , get sessionCtx
  if( getInstanceId() != null )
   sessionCtx = Context.getContextByInstanceID(getInstanceId());

  //if operationCtx has no parent and sessionCtx is not null, do chain
  if(getContext().getParent() == null && sessionCtx != null)
   getContext().chainTo(sessionCtx);

  //write input paramater into current context
  setValueAt("AccountNumber",uid);

  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  result.put("TrxReplyCode",getContext().getValueAt("TrxReplyCode"));

  ic = (IndexedCollection)getContext().getElementAt("accountStatementDetails");
  j = ic.size();
  result.put("ICollSize",new Integer(j).toString());

  for(i=0;i<j;i++{
   result.put("OpnDate"+i,ic.getValueAt(i+".OpnDate"));
   result.put("OpnDescription"+i,ic.getValueAt(i+".OpnDescription"));
   result.put("OpnAmount"+i,ic.getValueAt(i+".OpnAmount"));
   result.put("OpnBalance"+i,ic.getValueAt(i+".OpnBalance"));
  }

result.put("TrxErrorMessage",getContext().getValueAt("TrxErrorMessage"));

  close();

  return result;
 }
```

to:

```
public String execute(String request) throws BTTSAEException, Exception {

  initialize();

  //extract data into operation context
  FormatElement fmt = new FormatElement();
  fmt.setName(getName()+"ReqFmt");
  fmt.unformat(request,getContext());
  //genflow
  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  //generate the response data
  fmt.setName(getName()+"RepFmt");
  String result = fmt.format(getContext());

  //release operation context
  close();

  return result;
 }
```

Remove `executeJournalHostRequestDataStep(),executeSendHostStep(),executeJournalHostReplyDat` and `initialize()` method.

In CustomerSearch.java:

Modify`public Hashtable execute(BTTSystemData sysData, String uid)` to `public String execute(String request)`

In CustomerSearchBean.java, remove import `com.ibm.btt.services.jdbcjournalservice.*;`, and modify:

`public class CustomerSearchBean extends StatelessSingleAction` to `public class CustomerSearchBean extends BaseJavaSampleBean`, and remove`super.ejbCreate()` and `super.ejbRemove();`:

```
public void ejbCreate() throws BTTSAEException {
  super.ejbCreate();
 }
public void ejbRemove() {
  super.ejbRemove();
 }
```

Modify

```
public Hashtable execute(BTTSystemData sysData, String uid) throws BTTSAEException,Exception

  initialize(sysData);

  Hashtable result = new Hashtable();
  IndexedCollection ic = null;
  int i,j = 0;

  Context sessionCtx = null;

  //if instanceID != null , get sessionCtx
  if( getInstanceId() != null )
   sessionCtx = Context.getContextByInstanceID(getInstanceId());

  //if operationCtx has no parent and sessionCtx is not null, do chain
  if(getContext().getParent() == null && sessionCtx != null)
   getContext().chainTo(sessionCtx);

  System.out.println(sessionCtx);

  //write input paramater into current context
  setValueAt("CustomerId",uid);

  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();
  result.put("TrxReplyCode",getContext().getValueAt("TrxReplyCode"));
  result.put("CustomerName",getContext().getValueAt("CustomerName"));

  ic = (IndexedCollection)getContext().getElementAt("accounts");
  j = ic.size();
  result.put("ICollSize",new Integer(j).toString());

  for(i=0;i<j;i++){
   result.put("AccountNumber"+i,ic.getValueAt(i+".AccountNumber"));
   result.put("Type"+i,ic.getValueAt(i+".Type"));
   result.put("Name"+i,ic.getValueAt(i+".Name"));
   result.put("Balance"+i,ic.getValueAt(i+".Balance"));
  }

  result.put("TrxErrorMessage",getContext().getValueAt("TrxErrorMessage"));

  close();

  return result;
 }
```

To:

```
public String execute(String request) throws BTTSAEException,Exception {
  //initialize
  initialize();

  //extract data into operation context
  FormatElement fmt = new FormatElement();
  fmt.setName(getName()+"ReqFmt");
  fmt.unformat(request,getContext());

  //genflow
  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  //generate the response data
  fmt.setName(getName()+"RepFmt");
  String result = fmt.format(getContext());

  //release operation context
  close();

  return result;
 }
```

Remove `executeJournalHostRequestDataStep(),executeSendHostStep(),executeJournalHostReplyDataS` and `initialize()` method.

In Deposit.java, modify `public Hashtable execute(BTTSystemData sysData, String uid,String date,String amount)` to `public Context execute(Context ctx)`, and add `import com.ibm.btt.base.Context;`

In DepositBean.java, remove `import com.ibm.btt.services.jdbcjournalservice.*;`, and modify `public class DepositBean extends com.ibm.btt.server.bean.StatelessSingleAction` to `public class DepositBean extends BaseJavaSampleBean`, and remove `super.ejbCreate()` and `super.ejbRemove();`:

```
public void ejbCreate() throws BTTSAEException {
  super.ejbCreate();
 }
public void ejbRemove() {
  super.ejbRemove();
 }
```

Modify:

```
public Hashtable execute(BTTSystemData sysData, String uid,String date,String amount) throws

  initialize(sysData);

  Hashtable result = new Hashtable();
  IndexedCollection ic = null;
  int i,j = 0;

  Context sessionCtx = null;

  //if instanceID != null , get sessionCtx
  if( getInstanceId() != null )
   sessionCtx = Context.getContextByInstanceID(getInstanceId());

  //if operationCtx has no parent and sessionCtx is not null, do chain
  if(getContext().getParent() == null && sessionCtx != null)
   getContext().chainTo(sessionCtx);

  //write input paramater into current context
  //write input paramater into current context
  setValueAt("AccountNumber",uid);
  setValueAt("Date",date);
  setValueAt("Amount",amount);

  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  result.put("TrxReplyCode",getContext().getValueAt("TrxReplyCode"));
  result.put("AccountBalance",getContext().getValueAt("AccountBalance"));
  result.put("TrxErrorMessage",getContext().getValueAt("TrxErrorMessage"));

  close();

  return result;
 }
```

To:

```
public Context execute(Context ctx) throws BTTSAEException, Exception {

  setContext(ctx);

  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  //release context and clear operation
  setContext(null);
  close();

  return ctx;
 }
```

Remove
`executeJournalHostRequestDataStep(),executeSendHostStep(),executeJournalHostReplyDataS`
`and initialize()` method.

In EndSessionAction.java, modify `public Hashtable execute(BTTSystemData sysData)` to `public void execute()`.

In EndSessionActionBean.java, remove `super.ejbCreate()` and `super.ejbRemove();`:

```
public void ejbCreate() throws BTTSAEException {
  super.ejbCreate();
 }
public void ejbRemove() {
  super.ejbRemove();
 }
```

Modify:

```
public Hashtable execute(BTTSystemData sysData) throws BTTSAEException,Exception {

  initialize(sysData);

  Hashtable result = new Hashtable();
  Context sessionCtx ;

  //if instanceID != null
  if( getInstanceId() != null ){
   // Removes the context and its parent (the parent session context)
   sessionCtx = Context.getContextByInstanceID(getInstanceId());
   if(sessionCtx instanceof Context)
    sessionCtx.prune();
  }

  close();

  return result;
 }
```

To:

```
public void execute() throws BTTSAEException,Exception {

  //get session context
  doSessionPropagation();

  //if instanceID != null
  if( getInstanceId() != null ){
   Context sessionCtx = Context.getContextByInstanceID(getInstanceId());
   if(sessionCtx instanceof Context)
    sessionCtx.prune();
  }

 }
```

Remove `initialize(BTTSystemData sysData)` method

In StartupJavaSessionAction.java, add `import com.ibm.btt.base.Context;`, and modify `public Hashtable execute(BTTSystemData sysData,String wksContext,String wksParentContext)` to `public Context execute(Context ctx)`

In StartupJavaSessionActionBean.java, remove `super.ejbCreate()` and `super.ejbRemove();`:

```
public void ejbCreate() throws BTTSAEException {
  super.ejbCreate();
 }
public void ejbRemove() {
  super.ejbRemove();
 }
```

Modify:

```
public Hashtable execute(BTTSystemData sysData, String wksContext,String wksParentContext) th

  initialize(sysData);

  Hashtable ha = new Hashtable();
  try {

   setupSessionContext();
   ha.put("InstanceID", getInstanceId());

   close();

  } catch (BTTSAEException ex) {
   ex.printStackTrace();
   throw new DSEInvalidRequestException(DSEException.critical,
            getClass().getName(),
            "Not able to create the session context " + " in " + getName() + " \n" + ex.toStr
  }

  return ha;
 }
```

To:

```
public Context execute(Context ctx) throws BTTSAEException, Exception {
  try {
   //setup session context
   setupSessionContext();

   //set the instance back to presentation layer
   ctx.setValueAt("instanceId",getInstanceId());

   close();

   return ctx;
  } catch (BTTSAEException ex) {
   ex.printStackTrace();
   throw new DSEInvalidRequestException(DSEException.critical,       getClass().getName(),"No

getName() + " \n" + ex.toString());
  }
 }
```

Remove `initialize(BTTSystemData sysData)` method.

In Withdrawal.java, modify `public Hashtable execute(BTTSystemData sysData, String uid,String date,String amount)` to `public String execute(String request)`

In WithdrawalBean.java, remove `import com.ibm.btt.services.jdbcjournalservice.*;`, and modify `public class WithdrawalBean extends com.ibm.btt.server.bean.StatelessSingleAction` to `public class WithdrawalBean extends BaseJavaSampleBean`, and remove `super.ejbCreate()` and `super.ejbRemove();`:

```
public void ejbCreate() throws BTTSAEException {
  super.ejbCreate();
 }
public void ejbRemove() {
  super.ejbRemove();
 }
```

Modify:

```
public Hashtable execute(BTTSystemData sysData, String uid,String date,String amount) throws

  initialize(sysData);

  Hashtable result = new Hashtable();
  IndexedCollection ic = null;
  int i,j = 0;

  Context sessionCtx = null;

  //if instanceID != null , get sessionCtx
  if( getInstanceId() != null )
   sessionCtx = Context.getContextByInstanceID(getInstanceId());

  //if operationCtx has no parent and sessionCtx is not null, do chain
  if(getContext().getParent() == null && sessionCtx != null)
   getContext().chainTo(sessionCtx);

  //write input paramater into current context
  setValueAt("AccountNumber",uid);
  setValueAt("Date",date);
  setValueAt("Amount",amount);

  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  result.put("TrxReplyCode",getContext().getValueAt("TrxReplyCode"));
  result.put("AccountBalance",getContext().getValueAt("AccountBalance"));
  result.put("TrxErrorMessage",getContext().getValueAt("TrxErrorMessage"));

  close();

  return result;
 }
```

To:

```
public String execute(String request) throws BTTSAEException, Exception {
  //create context and chains to session ctx
  initialize();

  //extract data into operation context
  FormatElement fmt = new FormatElement();
  fmt.setName(getName()+"ReqFmt");
  fmt.unformat(request,getContext());

  //genflow
  executeJournalHostRequestDataStep();

  executeSendHostStep();

  executeJournalHostReplyDataStep();

  //generate the response data
  fmt.setName(getName()+"RepFmt");
  String result = fmt.format(getContext());

  //release operation context
  close();

  return result;
 }
```

Remove `executeJournalHostRequestDataStep(),executeSendHostStep(),executeJournalHostReplyDataS` and `initialize()` method.

10. Add BaseJavaSampleBean.java in /BTTJavaSampleEJB/ejbModule/ com.ibm.btt.samples.appl.business BaseJavaSampleBean.java:

```
package com.ibm.btt.samples.appl.business;

import javax.resource.cci.Connection;
import javax.resource.cci.ConnectionFactory;
import javax.resource.cci.Interaction;
import javax.resource.cci.InteractionSpec;

import com.ibm.btt.formatter.client.FormatElement;
import com.ibm.btt.samples.business.sna.lu0.DummyLu0ConnectionSpec;
import com.ibm.btt.samples.business.sna.lu0.DummyLu0InteractionSpec;
import com.ibm.btt.samples.business.sna.lu0.DummyLu0Record;
import com.ibm.btt.server.bean.StatelessSingleAction;
import com.ibm.btt.services.jdbc.journal.Journal;


public class BaseJavaSampleBean extends StatelessSingleAction {

 /**
  * Writes a record into the journal.
  */
 public void executeJournalHostRequestDataStep() throws Exception {

  Journal journal;
  setValueAt("HostBuff",((FormatElement) getFormat(getName()+"SendFmt")).format(getContext())

  // writes to the journal using the appropriate format
  journal = (Journal)getService("Journal");
  assignService("test",journal);
  Journal j =  (Journal) getService("test");
  journal.addRecord(getContext(),"preSendJournalFmt");//$NON-NLS-1$
  System.out.println("Journal HostRequest"+getName()+" ok");
  return;
 }

 /**
  * Sends data to the host using a synchronous method
  * and waits for the reply.
  */
 public void executeSendHostStep() throws Exception {

  javax.naming.Context initialContext = null;
  ConnectionFactory connectionFactory = null;

  if (initialContext == null) {
   initialContext = new javax.naming.InitialContext();
   connectionFactory = (ConnectionFactory) initialContext.lookup("snalu0");
  }
  // START TRANSACTION
  long begin = System.currentTimeMillis();


  // For testing Component-managed authentication.
  DummyLu0ConnectionSpec lu0ConnectionSpec = new DummyLu0ConnectionSpec();
  lu0ConnectionSpec.setUserName("sna");
  lu0ConnectionSpec.setPassword("sna");
  Connection connection = connectionFactory.getConnection(lu0ConnectionSpec);
  System.out.println("connection created...");


  // Beginning of testing SYNC_SEND_RECEIVE
  Interaction interaction = connection.createInteraction();
  System.out.println("interaction created...");
  DummyLu0InteractionSpec interactionSpec = new DummyLu0InteractionSpec();

  DummyLu0Record in = new DummyLu0Record();
  DummyLu0Record out = new DummyLu0Record();
  //getRequestData();
  in.setData((String)getValueAt("HostBuff"));
  System.out.println("data to host: : "+in.getData());

  interactionSpec.setInteractionVerb(InteractionSpec.SYNC_SEND_RECEIVE);
  interactionSpec.setExecutionTimeout(10000);

  interaction.execute(interactionSpec, in, out);

  System.out.println("data from host: : "+out.getData());

  interaction.close();
  System.out.println("interaction closed...");
  connection.close();
  System.out.println("connection closed...");

  com.ibm.btt.formatter.client.FormatElement fromHost = (com.ibm.btt.formatter.client.FormatElem
  fromHost.unformat(out.getData(),getContext());
  System.out.println("SendHost ok");

 }

 /**
  * Writes the data received from the host into the journal,
  * applying the appropriate format.
  */
  public void executeJournalHostReplyDataStep() throws Exception {

  Journal journal;
  journal = (Journal)getService("Journal");
  journal.addRecord(getContext(), "afterRecJournalFmt");//$NON-NLS-1$
  System.out.println("JournalHostReply"+getName()+" ok");
  return;

 }
}
```

11. Modify some classes in the /BTTJavaSampleWeb/com/ibm/btt/samples/appl/presentation.

    In JavaAccountStatementInvoker.java, modify:

```
import java.util.Map;
import java.util.Hashtable;
import javax.ejb.EJBHome;
import com.ibm.btt.base.BTTSystemData;
import com.ibm.btt.base.Context;
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidArgumentException;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.*;
import com.ibm.btt.cs.servlet.CSConstants;
import com.ibm.btt.samples.appl.business.*;
/**
 * This class implements a business processor invoker.
 * @copyright(c) Copyright IBM Corporation 2004,2005
 */
public class JavaWithdrawalInvoker extends BeanInvokerImpl implements BeanInvokerForJavaRequest{
 public Object executeEJB() throws Exception{
  //** Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Withdrawal bean = (Withdrawal) getBeanInvokerProxy();;
  BTTSystemData sys = getSystemData();
  Map result = bean.execute( getSystemData(),
                             (String) getEjbParameter("AccountNumber"),
                             (String)getEjbParameter("Date"),
                             (String) getEjbParameter("Amount"));
  return result;
 }
 public void parseRequestData(String requestData)
```

```
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
            //** Prepare session data in ejb parameters
   getEjbParameters().put(Constants.SESSION_ID, getSystemData().getSessionId());
   getEjbParameters().put(CSConstants.DATAAPPLICATIONIDKEY, getSystemData().getSubsessionId()
   //** Get AccountNumber value
   Tokenizer tokens = getDelimitedTokenizer(requestData);
   BeanInvokerFormatter formatter = getFormatter();
   String s =   formatter.unformatString((String)tokens.nextToken("#"),null);
   getEjbParameters().put("AccountNumber", s);
   //** Get Date value
   java.util.Date aDate = formatter.unformatDate((String)tokens.nextToken("#"),true,"ymd",tru
   getEjbParameters().put("Date", aDate.toString());
   //** Get Amount value
   //Double amt = (Double) formatter.unformatFloat((String)tokens.nextToken("#"),4,null);
   Float amt = (Float) formatter.unformatFloat((String)tokens.nextToken("#"));
   getEjbParameters().put("Amount", amt.toString());
   //** Get BranchId value
//   s = formatter.unformatString((String)tokens.nextToken("#"),null);
//   getEjbParameters().put("BranchId", s);
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  Map haResult = (Map)ejbResult;
  BeanInvokerFormatter formatter = getFormatter();
  String responseString = "";
  try {
   responseString = formatter.formatString((String)haResult.get("TrxReplyCode"), null);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("AccountBalance"), null);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("TrxErrorMessage"), null);
   responseString = formatter.addDelimiter(responseString, "#");
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
  return responseString;
 }
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  //** 1: Get EJBHome Object
  WithdrawalHome home = (WithdrawalHome)getHomeObject();
  //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manuall
  Withdrawal bean = null;
  try{
      bean = (Withdrawal)home.create();
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
  return bean;
 }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Obj
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObj
 }
}
```

To:

```
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.samples.appl.business.AccountStatement;
import com.ibm.btt.samples.appl.business.AccountStatementHome;

public class JavaAccountStatementInvoker
 extends BeanInvokerImpl
 implements BeanInvokerForJavaRequest {
 /**
  * @see com.ibm.btt.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
   getEjbParameters().put("request",requestData);
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
   }
 /**
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerImpl#executeEJB()
  */
 public Object executeEJB() throws Exception {
  AccountStatement bean = (AccountStatement)getBeanInvokerProxy();
  //call remote EJB mthod. The parameters can be retrieved from parameters hashtable
  String request = (String)getEjbParameter("request");
  return bean.execute(request);
  }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  return ejbResult;
 }
 /**
  * @see com.ibm.btt.cs.invoker.base.BeanInvoker#createBeanInvokerProxy()
  */
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  try{
  //Get EJBHome Object
  AccountStatementHome home = (AccountStatementHome)getHomeObject();
    //Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  AccountStatement bean = null;
  bean = (AccountStatement)home.create();
  return bean;
    } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
  }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Object
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObject

 }

}
```

In JavaCustomerSearchInvoker.java, modife:

```
import java.util.Hashtable;
import com.ibm.btt.cs.invoker.base.*;
import com.ibm.btt.base.*;
import com.ibm.btt.clientserver.*;
import com.ibm.btt.cs.servlet.*;
import com.ibm.btt.samples.appl.business.*;
/**
 * Java Invoker for CustomerSearch
 * @copyright(c) Copyright IBM Corporation 2004,2005
 */
public class JavaCustomerSearchInvoker
 extends BeanInvokerImpl
 implements BeanInvokerForJavaRequest {
 /**
  * @see com.ibm.btt.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
                  //** Prepare session data in ejb parameters
   getEjbParameters().put(Constants.SESSION_ID, getSystemData().getSessionId());
   getEjbParameters().put(CSConstants.DATAAPPLICATIONIDKEY, getSystemData().getSubsessionId()
     //** Get CustomerId value
   Tokenizer tokens = getDelimitedTokenizer(requestData);
   BeanInvokerFormatter formatter = getFormatter();
   String s =  formatter.unformatString((String)tokens.nextToken("#"), "");
      getEjbParameters().put("CustomerId", s);
      //** Code Generated end here
     } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
 }
 /**
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerImpl#executeEJB()
  */
 public Object executeEJB() throws Exception {
  CustomerSearch bean = (CustomerSearch)getBeanInvokerProxy();
  Hashtable result = new Hashtable();
  String customerId = null;
  //call remote EJB mthod. The parameters can be retrieved from parameters hashtable
  //if the EJB method result is NOT hashtable, customer should put the result into Hashtable 
  customerId = (String)getEjbParameter("CustomerId");
  result = (Hashtable)bean.execute(getSystemData(),customerId);
    return result;
 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  Hashtable haResult = (Hashtable)ejbResult;
  BeanInvokerFormatter formatter = getFormatter();
  int i,j;
  String responseString = "";
  String formatAs = "";
  try {
   responseString = formatter.formatString((String)haResult.get("TrxReplyCode"),formatAs);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("CustomerName"),formatAs);
   responseString = formatter.addDelimiter(responseString, "#");
      j = Integer.parseInt((String)haResult.get("ICollSize"));
   for(i=0;i<j;i++){
       responseString += formatter.formatString((String)haResult.get("AccountNumber"+i),forma
    responseString = formatter.addDelimiter(responseString, "@");
    responseString += formatter.formatString((String)haResult.get("Type"+i),formatAs);
    responseString = formatter.addDelimiter(responseString, "@");
    responseString += formatter.formatString((String)haResult.get("Name"+i),formatAs);
    responseString = formatter.addDelimiter(responseString, "@");
    responseString += formatter.formatString((String)haResult.get("Balance"+i),formatAs);
    responseString = formatter.addDelimiter(responseString, "@");
   }
      responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("TrxErrorMessage"),formatAs)
   responseString = formatter.addDelimiter(responseString, "#");
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
  return responseString;
 }
 /**
  * @see com.ibm.btt.cs.invoker.base.BeanInvoker#createBeanInvokerProxy()
  */
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  //Get EJBHome Object
  CustomerSearchHome home = (CustomerSearchHome)getHomeObject();
    //Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  CustomerSearch bean = null;
    try{
      bean = (CustomerSearch)home.create();
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
    return bean;
 }
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObject
 }
}
```

To:

```
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.samples.appl.business.CustomerSearch;
import com.ibm.btt.samples.appl.business.CustomerSearchHome;

public class JavaCustomerSearchInvoker
 extends BeanInvokerImpl
 implements BeanInvokerForJavaRequest {
 /**
  * @see com.ibm.btt.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String requestData) throws DSEInvalidRequestException, DSEObjectNo
  try {
   getEjbParameters().put("request",requestData);
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
 }

 /**
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerImpl#executeEJB()
  */
 public Object executeEJB() throws Exception {
  CustomerSearch bean = (CustomerSearch)getBeanInvokerProxy();
  //call remote EJB mthod. The parameters can be retrieved from parameters hashtable
  String request = (String)getEjbParameter("request");
  //result = (Hashtable)bean.execute(getSystemData(),customerId);
  return bean.execute(request);
  }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  return ejbResult;
 }
 /**
  * @see com.ibm.btt.cs.invoker.base.BeanInvoker#createBeanInvokerProxy()
  */
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  try{
  //Get EJBHome Object
  CustomerSearchHome home = (CustomerSearchHome)getHomeObject();
  //Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  CustomerSearch bean = null;
       bean = (CustomerSearch)home.create();
       return bean;
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
 }
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObj
   }
}
```

In JavaDepositInvoker.java, modify:

```
import java.util.Map;
import java.util.Hashtable;
import javax.ejb.EJBHome;
import com.ibm.btt.base.BTTSystemData;
import com.ibm.btt.base.Context;
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidArgumentException;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.*;
import com.ibm.btt.cs.servlet.CSConstants;
import com.ibm.btt.samples.appl.business.*;
/**
 * This class implements a business processor invoker.
 * @copyright(c) Copyright IBM Corporation 2004,2005
 */
public class JavaDepositInvoker extends BeanInvokerImpl implements BeanInvokerForJavaRequest{
 public Object executeEJB() throws Exception{
  //** Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Deposit bean = (Deposit) getBeanInvokerProxy();;
  BTTSystemData sys = getSystemData();
  Map result = bean.execute( getSystemData(),
           (String) getEjbParameter("AccountNumber"),
                             (String)getEjbParameter("Date"),
                             (String) getEjbParameter("Amount")
                             );
  return result;
 }
  public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {

            //** Prepare session data in ejb parameters
   getEjbParameters().put(Constants.SESSION_ID, getSystemData().getSessionId());
   getEjbParameters().put(CSConstants.DATAAPPLICATIONIDKEY, getSystemData().getSubsessionId()
     //** Get AccountNumber value
   Tokenizer tokens = getDelimitedTokenizer(requestData);
   BeanInvokerFormatter formatter = getFormatter();

   String s =   formatter.unformatString((String)tokens.nextToken("#"),null);
   getEjbParameters().put("AccountNumber", s);
   //** Get Date value
   java.util.Date aDate = formatter.unformatDate((String)tokens.nextToken("#"),true,"ymd",tru
   getEjbParameters().put("Date", aDate.toString());
      //** Get Amount value
   Float amt = (Float) formatter.unformatFloat((String)tokens.nextToken("#"));
   getEjbParameters().put("Amount", amt.toString());
      //** Get BranchId value
//   s = formatter.unformatString((String)tokens.nextToken("#"),null);
//   getEjbParameters().put("BranchId", s);
    } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  Map haResult = (Map)ejbResult;
  BeanInvokerFormatter formatter = getFormatter();
    String responseString = "";
  try {
   responseString = formatter.formatString((String)haResult.get("TrxReplyCode"), null);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("AccountBalance"), null);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("TrxErrorMessage"), null);
   responseString = formatter.addDelimiter(responseString, "#");
    } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
  return responseString;
 }
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
    //** 1: Get EJBHome Object
  DepositHome home = (DepositHome)getHomeObject();
    //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Deposit bean = null;
    try{
      bean = (Deposit)home.create();
   } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
    return bean;
 }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Object
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObject
  }
}
```

To:

```
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.Context;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.formatter.client.FormatElement;
import com.ibm.btt.samples.appl.business.Deposit;
import com.ibm.btt.samples.appl.business.DepositHome;

public class JavaDepositInvoker extends BeanInvokerImpl implements BeanInvokerForJavaRequest{
 public Object executeEJB() throws Exception{
  //** Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Deposit bean = (Deposit) getBeanInvokerProxy();;
  Context ctx = (Context) getEjbParameter("Context");
    ctx = bean.execute(ctx);
  return ctx;
 }
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
     //create operation context
      Context ctx = new Context("depositServerCtx",false);
   ctx.chainTo(Context.getContextByInstanceID(getSystemData().getInstanceId()));
      //unformat value into context
      FormatElement fmt = new FormatElement();
      fmt.setName("depositReqFmt");
      fmt.unformat(requestData, ctx);
      getEjbParameters().put("Context", ctx);
    } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
   String responseString = "";
  try {
   FormatElement fmt = new FormatElement();
   fmt.setName("depositRepFmt");
   responseString = fmt.format((Context) ejbResult);
    } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
  return responseString;
 }
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  try{
  //** 1: Get EJBHome Object
  DepositHome home = (DepositHome)getHomeObject();
    //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manua
  Deposit bean = null;
  bean = (Deposit)home.create();
  return bean;
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
   }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Obj
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObj
  }
}
```

In JavaEndSessionInvoker.java, modify:

```
import java.util.Hashtable;
import com.ibm.btt.cs.invoker.base.*;
import com.ibm.btt.cs.invoker.cache.*;
import com.ibm.btt.base.*;
import com.ibm.btt.clientserver.*;
import com.ibm.btt.cs.servlet.*;
import com.ibm.btt.samples.appl.business.*;
import com.ibm.btt.sm.CSSessionHandler;
/**
 * Java Invoker for EndSession
 * @copyright(c) Copyright IBM Corporation 2004,2005
 */
public class JavaEndSessionInvoker extends BeanInvokerImpl
 implements BeanInvokerForJavaRequest {
 /**
  * Constructor for JavaEndSessionInvoker.
  */
 public JavaEndSessionInvoker() {
  super();
 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {

            //** Prepare session data in ejb parameters
```

```
    getEjbParameters().put(Constants.SESSION_ID, getSystemData().getSessionId());
    getEjbParameters().put(CSConstants.DATAAPPLICATIONIDKEY, getSystemData().getSubsessionId());
    //** Code Generated end here

   } catch (Exception e) {
    throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
   }
  }
  /**
   * Removes the session entry in the
   * sessions table and the context hierarchy created in the initial
   * operation (EndSessionServerOp).
   * The operation context of this operation will be automatically
   * removed by the client server mechanism during the close() process.
   */
  public Object executeEJB() throws Exception {
   String sessionCtxName = "javaSessionCtx";
//  // Removes the session entry in the sessions table
   CSSessionHandler.removeSession(getSystemData().getSessionId());
   //  Context.removeSession((String)getEjbParameter(Constants.SESSION_ID));//$NON-NLS-1$
//  // Removes the context and its parent (the parent session context)
//  Context.getContextNamed(sessionCtxName).prune();
     EndSessionAction bean = (EndSessionAction)getBeanInvokerProxy();
   Hashtable result = bean.execute(getSystemData());
     //** Clear Bean Invoker Proxy Cache
   BeanInvokerProxyCache proxyCache = (BeanInvokerProxyCache)InvokerCacheFactory.getInstance(Invo
   proxyCache.clear((String)getEjbParameter(Constants.SESSION_ID));
   return result;
  }
  /**
   * @see com.ibm.btt.cs.invoker.BeanInvoker#createBeanInvokerProxy()
   */
  public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
   //** 1: Get EJBHome Object
   EndSessionActionHome home = (EndSessionActionHome)getHomeObject();

   //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manually
   EndSessionAction bean = null;
   try{
       bean = (EndSessionAction)home.create();
   } catch(Exception e){
    throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
   }
     return bean;
  }
  /**
   * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
   */
  public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
   Hashtable haResult = (Hashtable)ejbResult;
   String responseString = "";
   return responseString;
  }
  /*
   * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Object
   */
  public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObject

  }
}
```

To:

```
import javax.servlet.http.HttpSession;
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.cs.invoker.cache.BeanInvokerProxyCache;
import com.ibm.btt.cs.invoker.cache.InvokerCacheFactory;
import com.ibm.btt.samples.appl.business.EndSessionAction;
import com.ibm.btt.samples.appl.business.EndSessionActionHome;
import com.ibm.btt.sm.CSSessionHandler;

public class JavaEndSessionInvoker extends BeanInvokerImpl
 implements BeanInvokerForJavaRequest {
 private Object sessionObject = null;
 /**
  * Constructor for JavaEndSessionInvoker.
  */
 public JavaEndSessionInvoker() {
  super();
 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
 }
 /**
  * Removes the session entry in the
  * sessions table and the context hierarchy created in the initial
  * operation (EndSessionServerOp).
  * The operation context of this operation will be automatically
  * removed by the client server mechanism during the close() process.
  */
 public Object executeEJB() throws Exception {
  // Removes the session entry in the sessions table
  EndSessionAction bean = (EndSessionAction)getBeanInvokerProxy();
   bean.execute();
    CSSessionHandler.removeSession((HttpSession)sessionObject);
    //** Clear Bean Invoker Proxy Cache
  BeanInvokerProxyCache proxyCache = (BeanInvokerProxyCache)InvokerCacheFactory.getInstance(I
  proxyCache.clear(getSystemData().getSessionId());
  return null;
 }
 /**
  * @see com.ibm.btt.cs.invoker.BeanInvoker#createBeanInvokerProxy()
  */
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  try{
  //** 1: Get EJBHome Object
  EndSessionActionHome home = (EndSessionActionHome)getHomeObject();
    //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manua
  EndSessionAction bean = null;
  bean = (EndSessionAction)home.create();
    return bean;
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
   }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  return "";
 }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Obj
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObj
  this.sessionObject = sessionObject;
 }
}
```

In JavaStartupSessionInvoker.java, modify:

```
import java.util.Hashtable;
import javax.ejb.EJBHome;
import javax.ejb.EJBObject;
import javax.servlet.http.HttpSession;
import com.ibm.btt.base.BTTSystemData;
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerFormatter;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.cs.invoker.base.Tokenizer;
import com.ibm.btt.cs.servlet.CSConstants;
import com.ibm.btt.samples.appl.business.StartupJavaSessionAction;
import com.ibm.btt.samples.appl.business.StartupJavaSessionActionHome;
import com.ibm.btt.sm.CSSessionHandler;
import com.ibm.btt.sm.SessionEntry;
/**
* Java Invoker for StartupSession
 * @copyright(c) Copyright IBM Corporation 2004,2005
 */
public class JavaStartupSessionInvoker extends BeanInvokerImpl implements BeanInvokerForJavaRequ
 private Object sessionObject = null;
 /**
  * Customer should add thier code here to access the Single Action EJB
  */
 public Object executeEJB() throws Exception{
  Hashtable result = new Hashtable();
  //** 1: Call back-end server to logon first
  StartupJavaSessionAction bean = (StartupJavaSessionAction) getBeanInvokerProxy();;
  BTTSystemData sys = getSystemData();
  result = bean.execute(getSystemData(),"", "");
  // Get instnace ID
  String instanceID =  (String)result.get("InstanceID");
  //** 2: If logon success, Create Session Entry, and then put into SessionManagement
  SessionEntry se = new SessionEntry();
  se.setSessionId(getSystemData().getSessionId());
  se.setHttpSession((HttpSession)sessionObject);
  se.setCurrentContext(instanceID);
  se.setIpAddress((String)getEjbParameter("ipAddress"));
  se.setTID((String)getEjbParameter("TID"));
  CSSessionHandler.addSession(se);
  result = getEjbParameters();
  return result;
 }
 /**
  * @see com.ibm.dse.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
       //** Prepare session data in ejb parameters
   getEjbParameters().put(Constants.SESSION_ID, getSystemData().getSessionId());
   getEjbParameters().put(CSConstants.DATAAPPLICATIONIDKEY, getSystemData().getSubsessionId());
   //** Code Generated begin here
   //***** Parse RequestData according to the format definition in client side
   /*--
   <fmtDef id="startupReqFmt">
     <record>
       <fString dataName="TID"/>
       <delim delimChar="#"/>
       <fString dataName="WKSContext"/>
       <delim delimChar="#"/>
       <fString dataName="WKSParentContext"/>
       <delim delimChar="#"/>
       <fString dataName="permanentConnectionForEvents"/>
```

```
        <delim delimChar="#"/>
        <fString dataName="ipAddress"/>
        <delim delimChar="#"/>
        <fInteger dataName="eventsPort"/>
        <delim delimChar="#"/>
      </record>
    </fmtDef> --*/
      //** Get TID value
    Tokenizer tokens = getDelimitedTokenizer(requestData);
    BeanInvokerFormatter formatter = getFormatter();
    String s =  formatter.unformatString((String)tokens.nextToken("#"), null);
    getEjbParameters().put("TID", s);
    //** Get WKSContext value
    s = formatter.unformatString((String)tokens.nextToken("#"), null);
    getEjbParameters().put("WKSContext", s);
       //** Get WKSParentContext value
    s = formatter.unformatString((String)tokens.nextToken("#"), null);
    getEjbParameters().put("WKSParentContext", s);
      //** permanentConnectionForEvents
    s = formatter.unformatString((String)tokens.nextToken("#"), null);
    getEjbParameters().put("permanentConnectionForEvents", s);
    //** Get ipAddress value
    s = formatter.unformatString((String)tokens.nextToken("#"), null);
    getEjbParameters().put("ipAddress", s);
      //** Get eventsPort value
    s = formatter.unformatString((String)tokens.nextToken("#"), null);
    getEjbParameters().put("eventsPort", s);
    //** Code Generated end here

   } catch (Exception e) {
    e.printStackTrace();
    throw new DSEInvalidRequestException(CSConstants.COMPID, "", e.getMessage());
   }
    }
  /**
   * @see com.ibm.dse.cs.BeanInvoker#processRespondData()
   */
  public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
   Hashtable haResult = (Hashtable)ejbResult;
   BeanInvokerFormatter formatter = getFormatter();
     String responseString = "";
   try {
    responseString = formatter.formatString((String)haResult.get("TID"), null);
    responseString += formatter.addDelimiter(responseString, "#");
    responseString += formatter.formatString((String)haResult.get("WKSContext"), null);
    responseString = formatter.addDelimiter(responseString, "#");
    responseString += formatter.formatString((String)haResult.get("WKSParentContext"), null);
    responseString = formatter.addDelimiter(responseString, "#");
    responseString += formatter.formatString((String)haResult.get("permanentConnectionForEvent
    responseString = formatter.addDelimiter(responseString, "#");
    responseString += formatter.formatString((String)haResult.get("ipAddress"), null);
    responseString = formatter.addDelimiter(responseString, "#");
    responseString += formatter.formatString((String)haResult.get("eventsPort"), null);
    responseString = formatter.addDelimiter(responseString, "#");
   } catch (Exception e) {
    e.printStackTrace();
   }
   return responseString;
  }
  /**
   * @see com.ibm.btt.cs.invoker.BeanInvoker#createBeanInvokerProxy()
   */
  public Object createBeanInvokerProxy() throws DSEInvalidRequestException {
     //** 1: Get EJBHome Object
   StartupJavaSessionActionHome home = (StartupJavaSessionActionHome)getHomeObject();
    //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manual
   StartupJavaSessionAction bean = null;
```

```
    try{
   bean = (StartupJavaSessionAction)home.create();
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
    return bean;
   }

 /* (non-Javadoc)
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Object
  */
 public void setSessionObject(Object sessionObject)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  this.sessionObject = sessionObject;
 }
}
```

To:

```
import java.io.Serializable;
import javax.servlet.http.HttpSession;
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.Context;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.cs.servlet.CSConstants;
import com.ibm.btt.formatter.client.FormatElement;
import com.ibm.btt.samples.appl.business.StartupJavaSessionAction;
import com.ibm.btt.samples.appl.business.StartupJavaSessionActionHome;
import com.ibm.btt.sm.CSSessionHandler;
import com.ibm.btt.sm.SessionEntry;

public class JavaStartupSessionInvoker extends BeanInvokerImpl implements BeanInvokerForJavaRequ
 private Object sessionObject = null;
 /**
  * Customer should add thier code here to access the Single Action EJB
  */
 public Object executeEJB() throws Exception{
  //** 1: Call back-end server to logon first
  StartupJavaSessionAction bean = (StartupJavaSessionAction) getBeanInvokerProxy();
  Context ctx = bean.execute((Context) getEjbParameters().get("context"));
    //** 2: If logon success, Create Session Entry, and then put into SessionManagement
  SessionEntry se = new SessionEntry();
  se.setCurrentContext((Serializable) ctx.getValueAt("instanceId"));
  se.setIpAddress((String)ctx.getValueAt("ipAddress"));
  se.setTID((String)ctx.getValueAt("TID"));
  CSSessionHandler.addSession((HttpSession)sessionObject,se);
  return ctx;

 }
 /**
  * @see com.ibm.dse.cs.BeanInvoker#parseRequestData(Object)
  */
 public void parseRequestData(String request)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
      //create logon local context
   Context ctx = new Context("startupServerCtx",false);
   //extract data into local context
   FormatElement fmt = new FormatElement();
   fmt.setName("startupReqFmt");
   fmt.unformat(request,ctx);
   getEjbParameters().put("context",ctx);
     } catch (Exception e) {
   throw new DSEInvalidRequestException(CSConstants.COMPID, "", e.getMessage(), e);
  }
  }
 /**
  * @see com.ibm.dse.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  String responseString = "";
  try {
   FormatElement fmt = new FormatElement();
   fmt.setName("startupReqFmt");
   responseString = fmt.format((Context) ejbResult);
     } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }
  return responseString;
 }
 /**
  * @see com.ibm.btt.cs.invoker.BeanInvoker#createBeanInvokerProxy()
  */
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException {
  try{
    //** 1: Get EJBHome Object
  StartupJavaSessionActionHome home = (StartupJavaSessionActionHome)getHomeObject();
    //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manua
  StartupJavaSessionAction bean = null;
  bean = (StartupJavaSessionAction)home.create();
  return bean;
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }

 }
 /* (non-Javadoc)
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Obj
  */
 public void setSessionObject(Object sessionObject)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  this.sessionObject = sessionObject;
}
}
```

In JavaWithdrawalInvoker.java, modify:

```
import java.util.Map;
import java.util.Hashtable;
import javax.ejb.EJBHome;
import com.ibm.btt.base.BTTSystemData;
import com.ibm.btt.base.Context;
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidArgumentException;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.*;
import com.ibm.btt.cs.servlet.CSConstants;
import com.ibm.btt.samples.appl.business.*;
/**
 * This class implements a business processor invoker.
 * @copyright(c) Copyright IBM Corporation 2004,2005
 */
public class JavaWithdrawalInvoker extends BeanInvokerImpl implements BeanInvokerForJavaReque
 public Object executeEJB() throws Exception{
  //** Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Withdrawal bean = (Withdrawal) getBeanInvokerProxy();;
  BTTSystemData sys = getSystemData();
  Map result = bean.execute( getSystemData(),
                            (String) getEjbParameter("AccountNumber"),
                            (String)getEjbParameter("Date"),
                            (String) getEjbParameter("Amount"));
  return result;
 }
 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
                  //** Prepare session data in ejb parameters
   getEjbParameters().put(Constants.SESSION_ID, getSystemData().getSessionId());
   getEjbParameters().put(CSConstants.DATAAPPLICATIONIDKEY, getSystemData().getSubsessionId());
      //** Get AccountNumber value
   Tokenizer tokens = getDelimitedTokenizer(requestData);
   BeanInvokerFormatter formatter = getFormatter();
       String s =   formatter.unformatString((String)tokens.nextToken("#"),null);
   getEjbParameters().put("AccountNumber", s);
     //** Get Date value
   java.util.Date aDate = formatter.unformatDate((String)tokens.nextToken("#"),true,"ymd",true,"
   getEjbParameters().put("Date", aDate.toString());
     //** Get Amount value
   //Double amt = (Double) formatter.unformatFloat((String)tokens.nextToken("#"),4,null);
   Float amt = (Float) formatter.unformatFloat((String)tokens.nextToken("#"));
   getEjbParameters().put("Amount", amt.toString());
     //** Get BranchId value
//   s = formatter.unformatString((String)tokens.nextToken("#"),null);
//   getEjbParameters().put("BranchId", s);
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
  }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  Map haResult = (Map)ejbResult;
  BeanInvokerFormatter formatter = getFormatter();
   String responseString = "";
  try {
   responseString = formatter.formatString((String)haResult.get("TrxReplyCode"), null);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("AccountBalance"), null);
   responseString = formatter.addDelimiter(responseString, "#");
   responseString += formatter.formatString((String)haResult.get("TrxErrorMessage"), null);
   responseString = formatter.addDelimiter(responseString, "#");
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
  return responseString;
 }
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
    //** 1: Get EJBHome Object
  WithdrawalHome home = (WithdrawalHome)getHomeObject();
   //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Withdrawal bean = null;
   try{
      bean = (Withdrawal)home.create();
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage());
  }
    return bean;
 }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Object
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObject
   }
}
```

To:

```
import com.ibm.btt.base.Constants;
import com.ibm.btt.base.DSEInvalidRequestException;
import com.ibm.btt.base.DSEObjectNotFoundException;
import com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest;
import com.ibm.btt.cs.invoker.base.BeanInvokerImpl;
import com.ibm.btt.samples.appl.business.Withdrawal;
import com.ibm.btt.samples.appl.business.WithdrawalHome;

public class JavaWithdrawalInvoker extends BeanInvokerImpl implements BeanInvokerForJavaReque
 public Object executeEJB() throws Exception{
  //** Create Bean Proxy. If this bean is not existed in pool, then create new one manually
  Withdrawal bean = (Withdrawal) getBeanInvokerProxy();;
   String result = bean.execute((String) getEjbParameter("request"));
   return result;
 }

 public void parseRequestData(String requestData)
  throws DSEInvalidRequestException, DSEObjectNotFoundException {
  try {
   getEjbParameters().put("request",requestData);
  } catch (Exception e) {
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }

 }
 /**
  * @see com.ibm.btt.cs.BeanInvoker#processRespondData()
  */
 public Object processRespondData(Object ejbResult) throws DSEInvalidRequestException {
  return ejbResult;
 }
 public Object createBeanInvokerProxy() throws DSEInvalidRequestException{
  try{
  //** 1: Get EJBHome Object
  WithdrawalHome home = (WithdrawalHome)getHomeObject();
   //** 2: Create Bean Proxy. If this bean is not existed in pool, then create new one manual
  Withdrawal bean = null;
      bean = (Withdrawal)home.create();
     return bean;
  } catch(Exception e){
   throw new DSEInvalidRequestException(Constants.COMPID, "", e.getMessage(), e);
  }

 }
 /*
  * @see com.ibm.btt.cs.invoker.base.BeanInvokerForJavaRequest#setSessionObject(java.lang.Obj
  */
 public void setSessionObject(Object sessionObject) throws DSEInvalidRequestException, DSEObj
 }
}
```

Reference

For the references related to migration from version 5.1 to version 5.2, see the following:

Migrate services

BTT version 5.1 service compatible

BTT version 5.2 adopts the BTT version 4.3 architecture. It is enhanced to support EJB/Web Service interface and WebSphere pool management.

BTT version 5.1 supports 3 invocation ways: Local Java™, EJB and WSDL. The first two invocation ways are kept in BTT version 5.2, while WSDL invocation is not supported. The BTT version 5.1 JournalService and TableService are replaced by BTT version 4.3 architecture and invocation ways. The definition and usage are still compatible, but the customer extension may not be compatible.

How to migrate to BTT version 5.2 new service architecture

If you use pool service, you need to update your code manually. For example, your service implement BTTPoolable in version 5.1, then you need to implement WebSphere's PoolableObject interface.

BTT version 5.1 service architecture requires you to define a set of properties files and a Factory class and implementation class. In BTT version 5.2, it only requires your service object to implement Externalizable interface and define it in dsesvrce.xml. The getService series function is still workable at both SAE layer and BP layer.

In common cases, you can invoke version 5.1 Journal and Table service by getService() way. You don't need to change the code. But if you invoke them in a special way or make their own customer extensions, some extra work may be needed.

Migrate to use BTT version 5.2 services

In BTT version 5.2, the JDBC Table service and Electronical Journal service are still provided for the application. These two services are based on the new service architecture of BTT version 5.2. If the application needs these two services, you need to do the following:

1. Modify the service definition in dse.ini
   - For JDBC Table: Change the class name by `com.ibm.btt.services.jdbc.tablemapping.JDBCTable`
   - For Electronic Journal: Change the class name by `com.ibm.btt.services.jdbc.journal.JDBCJournal`
2. Modify the application code to comply with BTT version 5.2 specification: In BTT version 5.2, to get a new service instance is changed by `com.ibm.btt.base.Service.readObject`.
3. Use objectPool feature of WebSphere Application Server and WebSphere Portal Server instead of the generic pool of BTT: In BTT version 5.2, the object pool feature is provided by the application server. If the application needs to use the BTT version 5.2 service architecture with the pool mechanism, the pool configuration is changed to be complied with the objectPool of WAS and WPS.

Migrate presentation layer

After you finished the steps in "Migrating your applications to version 5.2" on page 1, you may still need to do some extra work:

- Struts configuration file – wsifAction: Since the BPEL architecture and implementation are changed totally. WSIF is not supported any more. The wsifAction in struts configuration file should be replaced by JSR109 enabled struts action. The following is the wsifAction sample:

```
<action path="/StateA"
  className="com.ibm.btt.struts.config.BTTActionConfig"
  name="stateAForm"
  type="com.ibm.btt.struts.wsif.WSIFAccessAction"
  context="branchServer"
  wsdlFile="com/ibm/btt/struts/sample/service/EchoJavaService.wsdl"
  ns="http://service.sample.struts.btt.ibm.com/EchoJavaService/"
  service="EchoService"
  pns="http://service.sample.struts.btt.ibm.com/Echo/"
  pname="Echo"
  mapToFormat="toEchoFormat"
  mapFromFormat="fromEchoFormat"
  opName="echo"
  inName="echoRequest"
  outName="echoResponse">
              .
              .
</action>
```

The new BPEL struts action should be the normal BTTBaseAction extensions within BPEL JSR109 facade invocation code. The following is the sample:

```
<action path="/StateA"
  className="com.ibm.btt.struts.config.BTTActionConfig"
  name="stateAForm"
  type="com.customer.jsr109.StateAAction"
  context="branchServer">
</action>
```

Note: The new BPEL action is not generic. It's tightly-coupled with BPEL service name and interface. In each BPEL action, you must put the invocation code to each BPEL service. The following is the sample code in BPEL action:

```
TransferServiceProxy transferProxy = new TransferServiceProxy();
transferProxy.transfer(param1, param2?);
```

The class name TransferServiceProxy is generated by BPEL engine automatically when developing the BPEL service.

- Struts Configuration file – WSIFMessage Mapping: The WSIF (BPEL 1.0) is leveraged on WSIFMessage Mapping file to transfer parameter from presentation layer to BPEL. In BTT version 5.2, all WSIFMessage mapping should be removed. The following is BTT Struts WSIFMessage Mapping definition sample:

```
<formats>
      <format id="toEchoFormat" className="">
              <wsif-map dataElement="userName" partKey="message"/>
      </format>
      <format id="fromEchoFormat" className="">
              <wsif-map dataElement="txCode" partKey="result"/>
              <wsif-map dataElement="txMessage" partKey="result"/>
      </format>
        .
        .
<formats>
```

Migrate Single Action EJB

BTT version 5.1 Single Action EJB programming model is still kept. BTT version 5.2 Single Action EJB provides more programming models, such as shared context mode, pass-through mode. For example, the Graphical Builder generated EJB has these kind of interfaces for your reference.

Note: Since WorkArea is always available in WAS6, it is recommended that you enable WorkArea in BTT defintion.

Migrate BPEL

The BPEL standard changed a lot from WBISF5.1 in BTT version 5.1 to WPS6 in BTT version 5.2. BTT BPEL extension programming model changed from strong type to shared mode. You need to create new BPEL and copy the logic. Yet the code may still be compatible and the BTT super classes such as BaseSnippet is still available for you to extend. To avoid logical problems, you need to reconsider the definition file such as BPEL file and logic.s

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Trademarks and service marks