IBM WebSphere Sample Adapter for Enterprise Information System Simulator 7.0.0.0 - Generation and Implementation on WID 7.0.0

Quick Start Scenarios
Note: Before using this information and the product it supports, read the information in the “Notices” section, at the end of this document.

This edition applies to version 7, release 0, and modification 0 of IBM WebSphere Adapter for Enterprise Information System Simulator and to all subsequent releases and modifications, until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2008. US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

TABLE OF CONTENTS
1.0 Overview

1.1 Introduction

IBM WebSphere® Sample Adapter for Enterprise Information System Simulator is a Sample Adapter used to demonstrate the IBM WebSphere® Adapter Toolkit capability to customers and business partners. WebSphere Sample Adapter for Enterprise Information System Simulator simulates the basic functions of the IBM WebSphere® Adapters and related Enterprise Information System. The IBM WebSphere® Adapter Toolkit can reduce the effort for the users, who need to build a custom adapter and configure the WebSphere Adapters to the related Enterprise Information System.

1.2 Architecture

IBM WebSphere® Sample Adapter for Enterprise Information System Simulator consists of two components: JCA-Compliant Adapter and EIS Mocker.

- **JCA-Compliant Adapter**: Based on WebSphere Adapter Foundation Classes, the JCA-compliant adapter can simulate the basic functions of the existing WebSphere Adapters.
- **EIS Mocker**: A SWT-based tool which can simulate the behaviors of Enterprise Information System.
2.0 JCA-Compliant Adapter

2.1 Supported business objects
The JCA-compliant adapter supports the business objects of the following WebSphere Adapters:
- WebSphere® Adapter for PeopleSoft
- WebSphere® Adapter for Siebel
- WebSphere® Adapter for OracleEBS
- WebSphere® Adapter for JDE
- WebSphere® Adapter for SAP

2.2 Supported server
The following servers are supported:
- WebSphere® Process Server
- WebSphere® Application Server

2.3 Outbound operations
The supported outbound operations are:
- Create
- Delete
- Update
- Retrieve
- ApplyChanges

Specify the following properties for outbound processing:

<table>
<thead>
<tr>
<th>Proper Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>Localhost</td>
<td>The IP or hostname of the system where the EIS Mocker runs.</td>
</tr>
<tr>
<td>Port</td>
<td>9000</td>
<td>The port number which the EIS Mocker uses.</td>
</tr>
</tbody>
</table>

2.4 Inbound operations
The supported inbound operations are:
- Create
- Delete
- Update

Specify the following properties for inbound processing:

<table>
<thead>
<tr>
<th>Proper Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
### 2.5 EMD tooling support

The EMD is provided to generate the needed artifact files. When running the Visual Test Adapter EMD, the property "**XSD Files Folder**" needs to be filled to specify the folder which contains the business object xsd files.

![Image of EMD tooling setup](image)

### 3.0 Creating Workspace, Project and Generating Adapter Classes using WID 7.0.0

The following sections describe the steps for creating a workspace, creating a new connector project and then generating the Adapter classes.

#### 3.1 Creating Workspace

1. Start IBM WebSphere Integration Developer.
2. Choose **Start > Program Files > IBM WebSphere Integration Developer > WebSphere Integration Developer V7.0.** This displays the Workspace Launcher dialog.

![Workspace Launcher Dialog](image)

3. Enter the new workspace directory **WSA_EISS** (WebSphere Sample Adapter for Enterprise Information System Simulator) along with the path, or click **Browse** to select a location for your project in the **Workspace** field.

4. The WID displays the new workspace as shown below:

![Workspace Display](image)

5. Choose **File > New > Project** option to select a wizard.
6. In the Select a wizard dialog, expand the Java EE folder and choose **Connector Project**, and click **Next**. This starts the New Connector Project wizard and displays the Connector Project dialog.
3.2 Creating a new Connector Project

For the project, we will use the name WSA_EISS. The project was initially known as VisualTestAdapter. Hence you will find names VTA in the project files generated and displayed in the screenshots.

1. In the Connector Project dialog, enter the new project name WSA_EISS (WebSphere Sample Adapter for Enterprise Information System Simulator) in
Set **Target Runtime** as WebSphere Process Server v7.0. Select Configuration as IBM WebSphere Adapter from the list box for **Configuration**. Uncheck the **Add to EAR** option. Click **Next**. This displays the **Source Folder** dialog.

**NEED TO DO**

2. Click **Next**.

This displays the J2C Resource Adapter Properties dialog.
3. Type the values for the fields in the J2C Resources Adapter Properties dialog as shown below.

4. Click Next. This displays the Generation Options dialog.
3.3 Generating Adapter Classes

1. Choose the outbound, inbound, Data binding, and EMD classes that need to be generated by checking on the check boxes provided for each of the processes.

   a. For Generating Outbound Adapter classes do not choose any of the check boxes in the Generation Options dialog. You may choose these options if the adapter you are building requires the support provided by these options. Each option has been explained as and when the
user chooses them.

b. For Generating Inbound Adapter classes the Event Polling Support is selected by default.
WebSphere software

New Connector Project

Generation Options
Select the specific components to generate

Adapter specification: IBM WebSphere Resource Adapter

Available components:
- Generate outbound adapter classes
- Generate inbound adapter classes
- Data binding classes for SCA
- Generate Enterprise Metadata Discover

Component properties:
- Connection pooling
  - Event polling support
  - Callback event support

Generate inbound adapter classes:
Creates code for the methods which must be implemented to produce a resource adapter capable of receiving events from an ERP.

Connection pooling:
The code created will support connection pooling.

Event polling:
This generates code that must be implemented to create a resource adapter capable of...
c. Check the other two options as shown below. These are for generating EMD classes and Data Binding classes.

2. Click **Finish**. This displays the Open Associated Perspective? dialog.

3. Click **No** for not changing the perspective.
The WAT generates the code and builds the workspace. The project has to be created without errors. The result should be as shown below.

You can now start implementing the business flow by modifying the java classes that have been generated by the IBM WebSphere Adapter Toolkit. For the Visual Test Adapter classes, the methods that have been modified are described in Section 5.

4.0 Implementing WebSphere Sample Adapter for Enterprise Information System Simulator Classes

The WebSphere Sample Adapter for Enterprise Information System Simulator classes have been modified to achieve the business flow as explained in the Architecture section. The detailed description of each class and their methods that have been modified after generating the classes using IBM WebSphere Adapter Toolkit are given below.
4.1 Package name com.ibm.j2ca.vta

4.1.1 Filename VTAResourceAdapter.java
Class name VTAResourceAdapter
Method Name getResourceAdapterMetadata()
Description This method will construct one instance of VTAWSAResourceAdapterMetadata class, and then return it.
Following is a code snippet to construct one instance of VTAWSAResourceAdapterMetadata class:

```java
if (raMetaData == null) {
    raMetaData = new VTAWSAResourceAdapterMetadata("IBM WebSphere Adapter for Visual Test", "IBM", "7.0", true);
}
return raMetaData;
```

Method Name createEventStore()
Description This method will construct one instance of VTAEEventStoreWithXid class, and then return it.
Following is a code snippet to construct one instance of VTAEEventStoreWithXid class:

```java
EventStoreWithXid eventStore = null;
try{
    eventStore = new VTAEEventStoreWithXid((VTAAActivationSpecWithXid)activationSpec, this);
} catch(Exception e) {
    throw new CommException(e.getMessage(), e);
}
return eventStore;
```

4.1.2 Filename VTAStructuredRecord.java
Class name VTAStructuredRecord
Method Name getNext()
Description This method will initialize the business object value according to the data retrieved from EIS side.
As the following code snippet shows:

```java
DataMap dataMap = (DataMap) this.getEISRepresentation();
if(dataMap.getName().endsWith("BG")){// it is a BG, then get the root BO.
    String bgName = dataMap.getName();
    String bgVerb = (String)dataMap.get(VTAConstants.VERB);
    this.setBGVerb(bgVerb);
```
String boName = bgName.substring(0, bgName.length()-2);
dataMap = dataMap.getDataMap(boName);
}
Type metadata = this.getMetadata();
OutputCursor topCursor = (OutputCursor)this.getTopLevelCursor();

if(this.getOperationName().equalsIgnoreCase(WBIInteractionSpec.EXISTS_OP)){
    //Boolean value = new Boolean(true);
    Boolean value = (Boolean)this.getEISRepresentation();
    topCursor.startObject();
    OutputAccessor accessor = (OutputAccessor)topCursor.getAccessors().get(0);
    accessor.setBoolean(value.booleanValue());
    topCursor.completeObject();
    return true;
}
if(this.getOperationName().equalsIgnoreCase(WBIInteractionSpec.RETRIEVE_ALL_OP)){
    return false;
}
topCursor.startObject();
VTAUtil.copyDataMapToRecord(dataMap, topCursor, metadata, this.getLogUtils());
topCursor.completeObject();
return true;

4.2  Package Name com.ibm.j2ca.vta.emd.description

4.2.1 Filename   VTADescription.java

Class name VTADescription

Method Name populateSchemaDefinitions()

Description
This method will populate the business object schema definition.
As the following code snippet shows:

try {

prepareChildSchemaFiles();
VTAMetadataObject object = (VTAMetadataObject) getMetadataObject();
// Extract the content after the tag for ComplexType> This is needed
// to add the annotation for object level ASI
String complexTypeInfo = object.getContent().substring(0, 
object.getContent().indexOf(">") + 1) + 
System.getProperty("line.separator");
String content;
String header = object.getHeader();
int encoding = header.indexOf("encoding");
if (encoding != -1) {
    String encodingName = header.substring(encoding + 10, 
    header.indexOf("\"", encoding + 10));
    if (!encodingName.equalsIgnoreCase("UTF-8")) {
        header = header.substring(0, encoding + 10) + "UTF-8" + 
        header.substring(header.indexOf("\"", encoding + 10));
    }
}
```java
} content = "header + includeXSDs.toString() + 
getSerializedImports(this.getImportNameSpaces()) + 
System.getProperty("line.separator") + 
getSerializedAnnotationForEMD(getASISchemaName()) 
+ complexTypeInfo + 
getSerializedMetadata(this.getMetadataForBusinessObject()) + 
object.getContent().substring(object.getContent().indexOf(">") + 1);
// Add defn to schema definition list
URI uri = new URI(this.getRelativePath() + getBOName() + ".xsd");
put(object.getTargetNameSpace(), uri, content);
}
} catch (Exception e) {
    throw new MetadataException("Unable to create Data Description from Object selected", e);
}

Method Name prepareChildSchemaFiles ()
Description

This method will populate the child business object schema definition recursively.
As the following code snippet shows:

```java
MethodName prepareChildSchemaFiles ()
Description

This method will populate the child business object schema definition recursively.
As the following code snippet shows:
```java
MetadataObjectResponse response = 
getMetadataObject().getChildren(null);
VTAMetadataObject child = null;
Iterator i = response.getObjectIterator();
while (i.hasNext()) { 
    child = (VTAMetadataObject) i.next();
    if (child.getBOType() != null) {
        if (child.getTargetNameSpace().equals(((VTAMetadataObject) 
this.getMetadataObject()).getTargetNameSpace()) &&
        child.isPartOfSameXSD()) {
            includeXSDs .append("<" + child.getAliasTag() + "include 
        schemaLocation=" + child.getBOName() + ".xsd" />
        }
    }
    boolean found = false;
    for (int j = 0; j < selectedSet.length; j++) {
        VTAMetadataObject selectedObj = (VTAMetadataObject) 
((VTAMetadataImportConfiguration) selectedSet[j]).getMetadataObject();
        if (selectedObj.getLocation().equals(child.getLocation())) {
            found = true;
            break;
        }
    }
    if (found)
        continue;
        // Generate the schemas for grand and following level children
        VTADataDescription desc = new VTADataDescription(child, 
selectedSet);
        desc.setRelativePath(this.getRelativePath());
        desc.setAliasTag(child.getAliasTag());
        desc.prepareChildSchemaFiles();
```
SchemaDefinition[] schemas = desc.getSchemaDefinitions();
for (int j = 0; j < schemas.length; j++) {
    SchemaDefinition defn = schemas[j];
    put(defn.getNamespace(), defn.getLocation(),
        defn.getContent());
} // If not found continue to add the defn of the child object
String complexTypeInfo = child.getContent().substring(0, child.getContent().indexOf(">") + 1) +
    System.getProperty("line.separator");
String header = child.getHeader();
int encoding = header.indexOf("encoding");
if (encoding != -1) {
    String encodingName = header.substring(encoding + 10,
        header.indexOf("", encoding + 10));
    if (!encodingName.equalsIgnoreCase("UTF-8"))
        header = header.substring(0, encoding + 10) + "UTF-8" +
            header.substring(header.indexOf("", encoding + 10));
} String content = header + desc.getIncludeXSDs() +
    getSerializedImports(this.getImportNameSpaces()) +
    System.getProperty("line.separator")
    + getSerializedAnnotationForEMD(getASISchemaName()) + complexTypeInfo +
    getSerializedMetadata(this.getMetadataForBusinessObject()) +
    child.getContent().substring(child.getContent().indexOf(">") + 1);
try {
    uri = null;
    uri = new URI(this.getRelativePath() + child.getBOName() +
        ".xsd");
} catch (URISyntaxException use) {
    throw new MetadataException(use.getMessage(),use);
} put(child.getTargetNameSpace(), uri, content);
} else {
    VTADataDescription boDesc = new VTADataDescription(child,
        selectedSet);
    boDesc.setRelativePath(this.getRelativePath());
    boDesc.setAliasTag(child.getAliasTag());
    boDesc.populateSchemaDefinitions();
    SchemaDefinition[] schemas = boDesc.getSchemaDefinitions();
    for (int j = 0; j < schemas.length; j++) {
        SchemaDefinition defn = schemas[j];
        put(defn.getNamespace(), defn.getLocation(),
            defn.getContent());
    }
}
This method will initialize the instance of the VTADescription class and return. As the following code snippet shows:

```java
VTAMetadataObject metadataObj = (VTAMetadataObject) spec.getMetadataObject()
String location = "/";
VTADescription dataDesc = new VTADescription(metadataObj, selection);
dataDesc.setMetadataObject(metadataObj);
dataDesc.setName(namespace, metadataObj.getDisplayName());
dataDesc.setRelativePath(location);

String artifactType = ((VTAMetadataSelection) selection).getSelectedArtifact();
if (artifactType.equalsIgnoreCase(MetadataConfigurationType.GENERATED_DATA)) {
    dataDesc.setTopLevel(false);
dataDesc.populateSchemaDefinitions();
dataDesc.setDataBindingGeneratorClassName("com.ibm.j2ca.vta.emd.run
dataDesc.setGenericDataBindingClassName(null);
} else if (artifactType.equalsIgnoreCase(MetadataConfigurationType.GENERATED_DATA)) {
    dataDesc.setTopLevel(false);
dataDesc.setGenericDataBindingClassName("com.ibm.j2ca.extension.dataexchange.b
}
else {// WMB
    dataDesc.setTopLevel(false);
dataDesc.populateSchemaDefinitions();
dataDesc.setGenericDataBindingClassName("com.ibm.j2ca.vta.VTASTructu
}
return dataDesc;
```

4.3 Package Name com.ibm.j2ca.vta.emd.discovery

4.3.1 Filename VTAMetadataObject.java

Class name VTAMetadataObject
Method Name createImportConfiguration()
Description This method will construct an instance of VTAMetadataImportConfiguration class and return it. As the following code snippet shows:

```java
VTAMetadataImportConfiguration conf = new VTAMetadataImportConfiguration(this);
conf.setLocation(this.getLocation());
this.setMetadataImportConfiguration(conf);
return conf;
```

4.3.2 Filename VTAMetadataSelection.java

Class name VTAMetadataSelection
Method Name createSelectionProperties()
Description This method will create the selection properties.
As the following code snippet shows:

```java
WBIPropertyGroupImpl pg = null;
try {
    pg = new WBIPropertyGroupImpl("Selection Properties");
    pg.setDisplayName("SelectionProperties");
    pg.setDescription("SelectionProperties");

    MetadataConfigurationType[] types =
    WBIMetadataDiscoveryImpl.getMetadataConfiguration();

    for(int i=0;i<types.length;i++)
        if (types[i].equals(MetadataConfigurationType.GENERIC_RECORDS)
            ||
            types[i].equals(MetadataConfigurationType.GENERATED_RECORDS)
                ||
            types[i].equals(MetadataConfigurationType.GENERATED_DATA_BINDING)
                ||
            types[i].equals(MetadataConfigurationType.GENERIC_DATA_BINDING)) {
            artifactType = types[i].toString();
        }
    if (types[i].equals(MetadataConfigurationType.INBOUND_SERVICE)
        ||
        types[i].equals(MetadataConfigurationType.OUTBOUND_SERVICE)) {
            serviceType = types[i].toString();
    }

    WBIMultiValuedPropertyImpl operationProp = new
    WBIMultiValuedPropertyImpl("Operations", String.class); //$NON-NLS-1$
    operationProp.setDisplayName("Operations");
    operationProp.setDescription("Operations");
    operationProp.setRequired(true);
    String[] operations = null;

    if (serviceType.equals(MetadataConfigurationType.INBOUND_SERVICE.toString())) {
        operations = new String[3];
        operations[0] = "Create";
        operations[1] = "Update";
        operations[2] = "Delete";
    }

    if (artifactType.equals(MetadataConfigurationType.GENERATED_DATA_BINDING.toString())){
        if (is WPS)
            operations = operationsWPS;
        else
            operations = operationsNonWPS;
    }

    operationProp.setValidValues(operations);
    for(int idx=0;idx<operations.length;idx++)
        operationProp.addValue(operations[idx]);
    pg.addProperty(operationProp);

    //Copy the applied properties to the new instance
    if (getAppliedSelectionProperties() != null)
        EMDUtil.copyValues(getAppliedSelectionProperties(), pg);
} catch (MetadataException e) {
```
4.3.3 Filename VTAMetadataTree.java

Class name VTAMetadataTree
Method Name listMetadataObjects()
Description
This method will list all the available Metadata Objects and add them to the MetadataTree.
As the following code snippet shows:

```java
WBIMetadataObjectResponseImpl response = new WBIMetadataObjectResponseImpl();
ArrayList objects = new ArrayList();

String[] fileNames = getFileNames(this.xsdFolderPath);
String directory = new File(this.xsdFolderPath).getAbsolutePath();
if (!directory.endsWith(File.separator)) {
    directory = directory + File.separator;
}
String fileLocation = null;
ArrayList objectsRegular = null;
if (fileNames != null && fileNames.length > 0) {
    for (int i = 0; i < fileNames.length; i++) {
        try {
            fileLocation = directory + fileNames[i];
            BOSchema schema = new BOSchema(fileLocation, WBIBiDiConstants.EMPTY_STR);
            Hashtable types = schema.getBOTypes();
            File file = new File(fileLocation);
            FileInputStream fs = new FileInputStream(file);
            int size = fs.available();
            byte[] bytes = new byte[size];
            fs.read(bytes);
            objectsRegular = prepareSelectionObjectList(bytes, types, fileLocation);
            objects.addAll(objectsRegular);
        } catch (Exception e) {
            if (helper.getLogUtils().isTraceEnabled(Level.FINE)) {
                helper.getLogUtils().trace(Level.SEVERE, "VTAMetadataTree", "listMetadataObjects", "Error in creating top level objects", e);
                throw new MetadataException("Unable to load objects " + e.getMessage(), e);
            }
        }
    }
}
```

Method Name: prepareSelectionObjectList()

Description:
This method will parse the XML file and prepare the business objects defined in the file.

As the following code snippet shows:

```java
response.setObjects(objects);
if (helper.getLogUtils().isTraceEnabled(Level.FINE))
return response;

ArrayList objects = new ArrayList();
String dataContent = new String(bytes);
try {
    int encoding = dataContent.indexOf("encoding");
    if (encoding != -1) {
        String encodingName = DTFUtils.getSubstring(dataContent, encoding + 10, dataContent.indexOf("\"", encoding + 10));
        if (encodingName.length() > 0) {
            dataContent = new String(bytes, encodingName);
        }
    }
    // Find the targetNamespace
    int targetNSIndex = dataContent.indexOf("targetNamespace=");
    String targetNameSpace = DTFUtils.getSubstring(dataContent, targetNSIndex + 17, dataContent.indexOf("\"", targetNSIndex + 18));
    // preprocess to remove comments
    while (true) {
        int index = dataContent.indexOf("<!--");
        if (index != -1) {
            int endIndex = dataContent.indexOf("-->", index);
            dataContent = DTFUtils.getSubstring(dataContent, 0, index) + dataContent.substring(endIndex + 4);
        } else 
            break;
    }
    // Check if the file contains complexTypes
    int compIndex = dataContent.indexOf("complexType");
    if (compIndex == -1)
        throw new MetadataException("Invalid XSD document, missing header information");
    int index = DTFUtils.getSubstring(dataContent, 0, compIndex).lastIndexOf("<");
    String header = DTFUtils.getSubstring(dataContent,
```

```java
```
String matchTag = DTFUtils.getSubstring(dataContent, index + 1, compIndex);

int footerIndex = dataContent.lastIndexOf("</" + matchTag + "complexType>");
// Store the footer to be added to each individual complexType defn
String footer = dataContent.substring(footerIndex + 16 + matchTag.length());

// Extract the contents of the document starting from first complexType to last complexType
String document = DTFUtils.getSubstring(dataContent, index, footerIndex + 16 + matchTag.length());
while (true) {
    // parse the document backwards for "complexType"
    index = document.lastIndexOf("<" + matchTag + "complexType");
    if (index < 0)
        break;
    String token = document.substring(index);
    int endingIndex = token.indexOf(">");
    String complexTypeString = DTFUtils.getSubstring(token, 0, endingIndex);
    int nameIndex = complexTypeString.indexOf("name=");
    if (nameIndex == -1) {
        document = DTFUtils.getSubstring(document, 0, index);
        continue;
    }
    String name = DTFUtils.getSubstring(complexTypeString, nameIndex + 6, token.indexOf(""", nameIndex + 7));
    VTAMetadataObject object = new VTAMetadataObject(this, helper);
    object.setDisplayName(name);
    object.setHeader(header);
    object.setAliasTag(matchTag);
    //object.setContentType(contentType);
    //object.setCharSet(charSet);
    object.setBOName(name);
    object.setLocation(targetNameSpace + name);
    object.setXsdName(fileLocation);
    treeNodes.put(object.getLocation(), object);
    VTAMetadataImportConfiguration conf = new VTAMetadataImportConfiguration(object);
    conf.setLocation(object.getLocation());
    object.setMetadataImportConfiguration(conf);
object.setContent(System.getProperty("line.separator") + token + footer);
object.setTargetNameSpace(targetNameSpace);

// Get the BOType to associate
// This is needed to link the parent child relationships
BOType type = (BOType) types.get(name);
if (type != null) {
    object.setBOType(type);
    if (type.getChildren() != null && type.getChildren().size() > 0) {
        object.setHasChildren(true);
    }
    object.setSelectableForImport(true);
    objects.add(object);
}

// Continue till we find complexType defns
if (index > 1) {
    document = DTFUtils.getSubstring(document, 0, index);
} else {
    break;
} else {
    document = DTFUtils.getSubstring(document, 0, index);
    continue;
}
}

} catch (Exception e) {
    throw new MetadataException(e.getMessage());
}

if (helper.getLogUtils().isTraceEnabled(Level.FINE))

return objects;

4.4 Package Name com.ibm.j2ca.vta.inbound

4.4.1 Filename VTAEventStoreWithXid.java

Class name VTAEventStoreWithXid
Method Name getRecordForEvent()
Description

This method will retrieve the corresponding record from EIS according to the specified event. As the following code snippet shows:

```java
VTAStructuredRecord record = new VTAStructuredRecord();
try {
```
String objectName = event.getEventType();
String opName = ((VTAEvent) event).getOperationName();
DataMap dataMap = (DataMap) dataObjectStore.getDataMapForEvent(event.getEventId());

record.setRecordName(objectName);
record.setOperationName(opName);
record.setEISRepresentation(dataMap);
record.setLogUtils(logUtils);
}

} catch (RemoteException re) {
    logUtils.log(LogLevel.FATAL, LogUtilConstants.ADAPTER_RBUNDLE, VTAConstants.VTAEVENTSTOREXID,
                       "getRecordForEvent", "6805", new Object[] { event.getEventId() }); //$NON-NLS-1$
    throw new CommException("The corresponding data could not be retrieved for an event.", re);
} catch (Exception e) {
    logUtils.log(LogLevel.FATAL, LogUtilConstants.ADAPTER_RBUNDLE, VTAConstants.VTAEVENTSTOREXID,
                       "getRecordForEvent", "6805", new Object[] { event.getEventId() }); //$NON-NLS-1$
    throw new ResourceException("The corresponding data could not be retrieved for an event.", e);
}

if (logUtils.isTraceEnabled(Level.FINE))
    return record;

Method Name: getEvents()
Description: This method will retrieve the available events from EIS according the specified event status.
As the following code snippet shows:

ArrayList events = new ArrayList();
if (pollQuantity == 0)
    return events;
try {
    ArrayList raEvents = dataObjectStore.getEvents();
    for (int i = 0; i < raEvents.size(); i++) {
        EisEvent metaData = (EisEvent) raEvents.get(i);

        VTAEvent event = new VTAEvent();
        event.setEventId(metaData.getEventId());
        event.setEventType(metaData.getEventType());
        event.setEventStatus(metaData.getStatus());

        event.setOperationName(metaData.getOperation());
        if (metaData.getStatus() == status){
            events.add(event);
        }
    }
} catch (Exception e) {
    logUtils.log(LogLevel.FATAL, LogUtilConstants.ADAPTER_RBUNDLE, VTAConstants.VTAEVENTSTOREXID,
                      "getRecordForEvent", "6805", new Object[] { event.getEventId() }); //$NON-NLS-1$
    throw new ResourceException("The corresponding data could not be retrieved for an event.", e);
}
Method Name: getPendingTransactions()

Description:
This method will retrieve the events whose Xid field is not empty.
As the following code snippet shows:

```java
if (dataObjectStore == null) {
    throw new ResourceException("The connection to the EIS Mocker is invalid.");
}
ArrayList xids = new ArrayList();
ArrayList raEvents = null;
try {
    raEvents = dataObjectStore.getEvents();
} catch (RemoteException re) {
    logUtils.log(LogLevel.SEVERE, LogUtilConstants.ADAPTER_RBUNDLE, VTAConstants.VTAEVENTSTOREXID, "getPendingTransactions", "6802");//$NON-NLS-1$
    throw new CommException("The polling of events from the event store cannot be done.", re);
}
for (int iev = 0; iev < raEvents.size(); iev++) {
    EisEvent metaData = (EisEvent) raEvents.get(iev);
    String xidStr = metaData.getXid();
    if (xidStr != null && !xidStr.equals("") && !xidStr.equals("0")) {
        Xid xid = new XidImpl(xidStr);
        xids.add(xid);
    }
}
int size = xids.size();
Xid xidArray[] = new XidImpl[size];
for (int ixid = 0; ixid < size; ixid++) {
    xidArray[ixid] = (Xid) xids.get(ixid);
}
```
if (logUtils.isTraceEnabled(Level.FINE))
return xidArray;

Method Name: getEventForXid()
Description: This method will retrieve the corresponding event from EIS according to the specified event xid.
As the following code snippet shows:

```java
VTAEvent event = new VTAEvent();
ArrayList raEvents = null;
boolean found = false;
try {
    raEvents = dataObjectStore.getEvents();
    for (int idx=0; idx<raEvents.size(); idx++) {
        EisEvent metaData = (EisEvent) raEvents.get(idx);
        if (metaData.getXid().equals(xid.toString())) {
            event.setEventId(metaData.getEventId());
            event.setEventType(metaData.getEventType());
            event.setEventStatus(metaData.getStatus());
            event.setOperationName(metaData.getOperation());
            found = true;
            break;
        }
    }
} catch (RemoteException re) {
    logUtils.log(LogLevel.FATAL, LogUtilConstants.ADAPTER_RBUNDLE, VTAConstants.VTAEVENTSTOREXID, "getEventForXid", "6806", new Object[] {xid.toString()}); //$NON-NLS-1$
    throw new CommException("The corresponding event could not be retrieved for an xid.", re);
}
if (!found) {
    logUtils.log(LogLevel.FATAL, LogUtilConstants.ADAPTER_RBUNDLE, VTAConstants.VTAEVENTSTOREXID, "getEventForXid", "6806", new Object[] {xid.toString()}); //$NON-NLS-1$
    throw new ResourceException("The corresponding event could not be retrieved for an xid.");
}
if (logUtils.isTraceEnabled(Level.FINE))
return event;
```
4.5 Package Name com.ibm.j2ca.vta.outbound

4.5.1 Filename VTAInteraction.java

Class name VTAInteraction
Method Name execute()

Description
This method will process the input record, and then return the result as output record.
As the following code snippet shows:

```java
Record output = null;
try {
    WBIInteractionSpec wbiSpec = (WBIInteractionSpec) ispec;
    VTAStructuredRecord recordInput = null;
    if (inputRecord instanceof VTAStructuredRecord) {
        recordInput = (VTAStructuredRecord) inputRecord;
    } else if (inputRecord instanceof DataObjectRecord) {//is SDO1
        DataObjectRecord record = (DataObjectRecord) inputRecord;
        recordInput = new VTAStructuredRecord();
        DEFactorySDO binding = new DEFactorySDO();
        DataObject object = record.getDataObject();
        //Since it would be a BG get the root property
        commonj.sdo.Property prop = WPSServiceHelper.getRootBusinessObjectProperty(object.getType());
        //get the corresponding dataObject from BG
        DataObject inputObject = object.getDataObject(prop);
        binding.setBoundObject(inputObject);
        //initialize the record
        Object[] array = {inputObject};
        recordInput.initializeInput(binding, array);
        recordInput.setNamespace(object.getType().getURI());
        recordInput.setRecordName(object.getType().getName());
        //set the verb as operationName for backward compatability
        //recordInput.setOperationName(object.getString("verb"));
        //CR91999 - process the BG case.
        recordInput.setIsBG(true);
        recordInput.setBGVerb(object.getString(VTAConstants.VERB));
    } else {//is JavaBean
        recordInput = new VTAStructuredRecord();
        DEFactoryJavaBean binding = new
```

DEFactoryJavaBean();
```
binding.setBoundObject(inputRecord);
    Object[] array = {inputRecord};
    recordInput.initializeInput(binding, array);
}
recordInput.setLogUtils(logUtils);
CommandForCursor command = commandMgr.produceCommands(recordInput, wbiSpec.getFunctionName());
    //CR91999 - process the BG case.
    VTABaseCommand vtaCmd = (VTABaseCommand)command;
    vtaCmd.setFuncName(wbiSpec.getFunctionName());
    if (recordInput.getIsBG()) {
        vtaCmd.setIsBG(true);
        vtaCmd.setBGVerb(recordInput.getBGVerb());
    }
    interpreter.execute(command);
    VTAStructuredRecord outputRecord = new VTAStructuredRecord();
    outputRecord.setEISRepresentation(command.getEisRepresentation());
    outputRecord.setMetadata(recordInput.getMetadata());
    outputRecord.setOperationName(wbiSpec.getFunctionName());
    outputRecord.setLogUtils(logUtils);
    output = outputRecord;
    if (logUtils.isTraceEnabled(Level.FINE))
        logUtils.traceMethodExit(VTAConstants.VTAINTERACTION, "execute");
} catch (CommException ce) {
    throw ce;
} catch (ResourceException re) {
    throw re;
} catch (Exception e) {
    logUtils.trace(Level.FINEST, VTAConstants.VTAINTERACTION, "execute","The execute call on the VTAInteraction instance has failed.", e);
    throw new ResourceException("The execute call on the VTAInteraction instance has failed.", e);
}
    return output;

### 4.5.2 Filename  VTAManagedConnection.java

**Class name**  VTAManagedConnection  
**Method Name**  getWBIConnection()  
**Description**  This method will establish a connection to the EIS.  
As the following code snippet shows:
```java
if (rmiSession == null) {
    String host = factory.getHostname();
    String port = factory.getPortnumber();
    // log the properties
    if (logUtils.isTraceEnabled(Level.FINE)) {
        logUtils.trace(Level.FINEST, VTACl biconstants.VTAEVENTSTOREXID, "VTAEventStoreWithXid", "The host name is " + host);
        logUtils.trace(Level.FINEST, VTACl biconstants.VTAEVENTSTOREXID, "VTAEventStoreWithXid", "The port is " + port);
    }
    if (host!=null && host.length()>0 && port != null && port.length()>0){
        try {
            rmiSession = new RMIClient(host, Integer.parseInt(port));
        } catch (Exception e) {// fail to establish the connection.
            logUtils.log(LogLevel.FATAL, LogUtilConstants.ADAPTER_RBUNDLE, VTACl biconstants.VTAMANAGEDCONNECTION, "getWBIConnection", "6011");//$NON-NLS-1$
            throw new CommException("The adapter failed to establish the connection with EIS Mocker.",e);
        }
    } else {
        // throw new ResourceException("The property value of host or port is invalid.");
    }
}

if (logUtils.isTraceEnabled(Level.FINE))
    logUtils.traceMethodExit(VTACl biconstants.VTAMANAGEDCONNECTION, "getWBIConnection");

return new VTACl biconnection(this);
```

4.6 Package Name com.ibm.j2ca.vta.outbound.commands

4.6.1 FilenameVTABaseCommand.java

Class name VTABaseCommand

Method Name execute()

Description

This method will send the input data to EIS, and retrieve the result data. As the following code snippet shows:

```java
logUtils = getLogUtils();
if (logUtils.isTraceEnabled(Level.FINE))
    logUtils.traceMethodEntrance(VTACl biconstants.VTABASECOMMAND,
```
"execute";

try{
    client = (RMIObjectClient) this.getConnection();
    dataObjStore = client.getDataObjectStore();
} catch(RemoteException re){
    logUtils.trace(Level.FINEST,
    VTAConstants.VTABASECOMMAND, "execute","The Adapter failed to establish
    session with EIS Mocker",re);
    throw new CommException("The Adapter failed to
    establish session with EIS Mocker",re);
}

try{
    DataMap inputDataMap = null;
    if(this.getIsBG()){
        String boName = metadata.getName();
        String bgName = boName + "BG";
        inputDataMap = new DataMap(bgName);
        DataMap boDataMap = inputDataMap.createDataMap(boName, false);
        inputDataMap.set(VTAConstants.VERB,
        this.bgVerb);
        VTAUtil.copyRecordToDataMap(inputObject, metadata, boDataMap,logUtils);
    } else{
        inputDataMap =
        VTAUtil.convertRecordToDataMap(inputObject, metadata, this,logUtils);
    }
    DataMap outputDataMap =
    dataObjStore.sendDataMap(inputDataMap,this.getFuncName());

    this.setEisRepresentation(outputDataMap);

    if(WBIInteractionSpec.EXISTS_OP.equalsIgnoreCase(operationName)){
        this.setEisRepresentation(new Boolean(true));
    }
} catch (RemoteException re) {
    logUtils.trace(Level.FINEST,
    VTAConstants.VTABASECOMMAND, "execute","The Adapter failed to send data
to EIS Mocker",re);
    throw new CommException("The Adapter failed to send
data to EIS Mocker",re);
} catch (Exception e) {
    throw new ResourceException("The execute call on the
    VTABaseCommand instance has failed",e);
}

if(logUtils.isTraceEnabled(Level.FINE))
    logUtils.traceMethodExit(VTAConstants.VTABASECOMMAND,
    "execute");
4.7 Package Name com.ibm.j2ca.vta.util

4.7.1 Filename VTAUtil.java
Class name VTAUtil
Method Name convertRecordToDataMap()
Description
This method will convert the Record to DataMap.
As the following code snippet shows:

```java
String boName = metadata.getName();
DataMap outDataMap = new DataMap(boName);
try {
    copyRecordToDataMap(inputObject, metadata,
    outDataMap, logUtils);
} catch (ResourceException re) {
    throw new DESPIException(re.getMessage());
}
return outDataMap;
```

Method Name copyDataMapToRecord()
Description
This method will copy the data from DataMap to Record.
As the following code snippet shows:

```java
for (Iterator e = metadata.getPropertyIterator();
e.hasNext(); ) {
    Property prop = (Property) e.next();
    String propName = prop.getName();
    Object value = inDataMap.get(propName);
    if (value == null) {
        continue;
    } else {// is Child BO
        if (prop.isContainment()) {// N-Card
            Type childMetadata = prop.getType();
            List childList =
            inDataMap.getList(propName);
            OutputCursor childCursor =
            (OutputCursor) outputObject.getChildCursor(propName);
            for (int j = 0; j < childList.size(); j++) {
                DataMap childDataMap = (DataMap)
                childList.get(j);
                childCursor.startObject();
                copyDataMapToRecord(childDataMap,
                childCursor, childMetadata, logUtils);
                childCursor.completeObject();
            }
        } else {// 1-Card
            Type childMetadata = prop.getType();
            OutputCursor childCursor =
            (OutputCursor) outputObject.getChildCursor(propName);
            DataMap childDataMap = (DataMap) value;
```

```java
```
Method Name: copyRecordToDataMap()
Description:
This method will copy the data from Record to DataMap. As the following code snippet shows:

```java
for (Iterator e = metadata.getPropertyIterator();
e.hasNext();) {
    Property prop = (Property) e.next();
    String propName = prop.getName();
    if (prop.isContainment()) { // is Child BO
        InputCursor childCursor = (InputCursor)inputObject.getChildCursor(propName);
        childCursor.reset();
        while (childCursor.getNext()) {
            DataMap childDataMap = outDataMap.createDataMap(propName, prop.isMany());
            copyRecordToDataMap(childCursor, prop.getType(), childDataMap, logUtils);
        }
    } else { // is simple type
        InputAccessor accessor = (InputAccessor) inputObject.getAccessor(propName);
        if (accessor != null && accessor.isSet()) {
            if (prop.isMany()) { // N-Card
                List valueList = (List) accessor.getObject();
                ArrayList newList = new ArrayList();
                for (int idx = 0; idx < valueList.size(); idx++) {
                    newList.add(valueList.get(idx));
                }
                outDataMap.set(propName, newList);
            } else { // 1-Card
                outDataMap.set(propName, accessor.getObject());
            }
        }
    }
}
```
5.0 Configuring the adapter on WID/WPS 7.0 with the WSA_EISS project

This section provides the step by step process to configure the adapter for Outbound processing and Inbound processing in WID/WPS environment using the WSA_EISS project created as described in sections 4 and 5. The same may be achieved using a RAR file which is available in the same location. The steps for using the RAR file are given in the QSS document for Deployment and testing. The following steps are for configuring the adapter directly from the project space without creating a RAR.

5.1 Configuring the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration required for outbound processing.

To generate the business objects and related artifacts, perform the following steps:

1. In the Business Integration perspective, select File > New > External Service.
2. In the **External Service** window, select **Adapters > Unlisted Adapter** and click **Next**.

3. In the **Select an Adapter** panel, select **WebSphere Sample Adapter for Enterprise Information System Simulator > WSA_EISS** and click
Next.

New External Service

Select an Adapter
Select the version of the adapter that you want to use.

- WebSphere Sample Adapter for Enterprise Information System Simulator (IBM 7.0.0.0)
- WSA_EISS
4. In the **Processing Direction** panel, select **Outbound** and click **Next**.
5. In the **Discovery Configuration** panel, click **Browse** to specify the path of the **XSD files folder** and click **Next**.
6. In the **Object Discovery and Selection** panel, select **WbiCustomerCiBG** as the **Selected objects** and click **Next**.
7. In the **Configure Composite Properties** panel, click **Next**.
8. In the **Service Generation and Deployment Configuration** panel, select Other and click **Next**.
9. In **Service Location Properties** panel, click **Finish**.

This generates the **VTAOutboundInterface** component as shown below:
The interface has to be created without errors. Now this is the completion of the EMD run for outbound. For actual deploying the adapter on runtime environment and sending requests and verifying the results, refer to the QSS document for Deployment and testing WebSphere Sample Adapter for Enterprise Information System Simulator available in the same location.

5.2 Configuring the adapter for inbound processing

Run the external service wizard to specify business objects, services, and configuration that have to be used for inbound processing.

To generate the business objects and related artifacts, perform the following steps:
1. In the **Business Integration** perspective, select **File > New > External Service**.
2. In the **External Service** panel, select **Adapters > Unlisted Adapter**, and click **Next**.

3. In the **Select an Adapter** panel, select **WebSphere Sample Adapter for Enterprise Information System Simulator -> WSA_WISS**, and click
Select an Adapter

Select the version of the adapter that you want to use.

- [ ] WebSphere Sample Adapter for Enterprise Information System Simulator (IBM: 7.0.0.0)
- [x] WSA_EISS

Next.
4. In the **Processing Direction** panel, select **Inbound**, and click **Next**.
5. In the **Discovery Configuration** panel, use **Browse** to specify the path of the **XSD files folder**, and click **Next**.

6. In **Object Discovery and Selection** panel, select **WbiCustomerCiBG** as the **Selected objects**, and click **Next**.
7. In the **Configure Composite Properties** panel, click **Next**.
8. In **Service Generation and Deployment Configuration** panel, select Other, and click **Next**.
9. In the **Service Location Properties** panel, click **Finish**.
10. The **VTAInboundInterface** component will be generated as shown below:
The interface has to be created without errors. Now this is the completion of the EMD run for inbound. For actually deploying the adapter on runtime environment and sending requests and verifying the results, refer to the QSS document for Deployment and testing Visual Test Adapter available in the same location.
6.0 Building the RAR file on WID7.0

A Resource Adapter aRchive (RAR) is an archive file format defined in the J2EE Connector Architecture (JCA) specification. A RAR file is the valid format for deployment of resource adapters on application servers. Note that although the file extension of RAR is used (to signify Resource Adapter aRchive), this use of the file extension RAR should not be confused with the RAR file extension that indicates use of RAR compression. J2EE RAR files may also be called connectors.

The RAR file for the WebSphere Sample Adapter for Enterprise Information System Simulator can be created by following the steps given below. The RAR file is used to deploy the adapter on a different environment by importing the same.

6.1 Add the required jar files to the project workspace

1. Open the WID workspace where the WebSphere Sample Adapter for Enterprise Information System Simulator Project is in place. Click on File > Import as shown below.
2. Choose **File System** from the options as shown below and click **Next**.
3. Click **Browse** and choose the folder where the jar files (Ex: DESPI.jar, commonj.connector.jar) are located. Then check all the check boxes for the jar files that have to be bundled with the RAR file on the right side of the panel. Choose the **connector module** as the destination folder. Click **Finish**. All the jar files will be added to the Project workspace and these jar files will be bundled with the RAR file which can be created according to the steps given in the following section.
6.2 Creating a RAR file for Visual Test Adapter 7.0

1. Open the WID workspace where the Visual Test Adapter Project is in place. Click on **File > Export** as shown below.
2. Choose **RAR file** option from Java EE as shown below. Click **Next**.
3. Choose the folder where the rar file has to be created. Provide a name for the rar file. Use the other default options as given below. Then click **Finish**.
The RAR archive will be shown below with the jar files and the WebSphere Sample Adapter for Enterprise Information System Simulator project files (connector module and META_INF files)
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other 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 grant 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 106-0032, 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 the 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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation Department
2Z4A/SOM1 294 Route 100
Somers, NY 10589-0100 U.S.A.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Programming interface information**

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtains the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published.

Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Screens are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.*

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project ([http://www.eclipse.org](http://www.eclipse.org)).