

IBM® WebSphere® Commerce Award T&C Reference Application

IBM® WebSphere® Commerce
Business Edition

Version 5.4

Last Updated: August 23, 2002

Introduction:

This reference application demonstrates an extended WebSphere Commerce 5.4 BE contract model. This functionality would implement an award business process and allow the shopper to collect and redeem award points. The application is designed and packaged on top of WCS 5.4 BE and the shopping flow is built on top of ToolTech store.

Prerequisite Environment:

Operating System	Windows NT / 2000
Commerce Server	IBM WebSphere Commerce Business Edition 5.4
Application Server	WAS 4.x
WCS Database	IBM DB2 (with required fix-patches for WCS 5.4)

- No customization has been done on WCSContract EJB group.
- ToolTech store has been published.

Installing the Reference Application:

At a high level the installation includes the following tasks:

- Unpackages the driver assets into their appropriate directories.
- Updates the commerce database.
- Install and start the enterprise application.

Unpackages the driver assets into their appropriate directories

1) Back up following files:

- *drive:* \WebSphere\AppServer\installedApps\WC_Enterprise_App_instanceName.ear\properties\com\ibm\commerce\tools\contract\properties\ContractRB_en_US.properties
- *drive:* \WebSphere\AppServer\installedApps\WC_Enterprise_App_instanceName.ear\wctools.war\javascript\tools\contract\Contract.js
- *drive:* \WebSphere\AppServer\installedApps\WC_Enterprise_App_instanceName.ear\wctools.war\javascript\tools\contract\Extensions.js
- *drive:* \WebSphere\AppServer\installedApps\WC_Enterprise_App_instanceName.ear\wctools.war\tools\contract\ContractSummary.jsp

- *drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instanceName.ear\wcstore*
s.war\ToolTechStoreDirectoryName\Shoppingcart.jsp
- *drive:\WebSphere\CommerceServer\xml\trading\B2BTrading.dtd*
- *drive:\WebSphere\CommerceServer\xml\tools\contract\ContractNotebook.xml*

2) Open file <AwardTC_RefApp>\wcs\schema\db2\tooltech.sql, replace 'ToolTech Contract number 4567' with '<your published ToolTech store archive name> Contract number 4567'

3) Go to directory <AwardTC_RefApp>\driver\ear\wcstores.war, rename directory 'ToolTech' to your published ToolTech store directory name.

4) Zip contents of directory <AwardTC_RefApp>\driver\ear (i.e, properties, wcstores.war, wctools.war) to AwardTCEar.zip

Unzip AwardTCEar.zip to directory

drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instanceName.ear

5) Zip contents of directory <AwardTC_RefApp>\driver\wcs (i.e., schema, temp, xml) to AwardTCWCS.zip

Unzip AwardTCWCS.zip to directory *drive:\WebSphere\CommerceServer*

Updates the commerce database

1) Open a db2 command window, connect to target database, load general award schema data by running:

db2 -tvf *drive:\WebSphere\CommerceServer\schema\db2\award.sql*

2) On the same db2 command window, load ToolTech specific schema data by running:

db2 -tvf *drive:\WebSphere\CommerceServer\schema\db2\tooltech.sql*

3) At a DOS command prompt, navigate to the following directory:

drive:\WebSphere\CommerceServer\bin

Load access control policies by running: (all in one line)

acpload <dbname> <dbuser> <dbpswd>

drive:\WebSphere\CommerceServer\xml\policies\xml\award_ac_policy.xml

Install and start the enterprise application

This includes following 6 steps. Appendix B of Programming Guide has a detailed description of each step. To keep this reference application completed, we will also list all the necessary steps here.

- 1) Exporting the current WebSphere Commerce enterprise application
- 2) Exporting configuration information for enterprise beans
- 3) Assembling the new enterprise beans into existing enterprise application
- 4) Stopping and removing the current enterprise application
- 5) Importing the new enterprise application
- 6) Starting the new enterprise application

1) Exporting the current WebSphere Commerce enterprise application

To export the current enterprise application from the WebSphere Application Server, do the following:

1. Ensure that you have the following directory on the target WebSphere Commerce Server:

drive:\WebSphere\CommerceServer\working

If you do not already have this directory, create it now.

2. Open the WebSphere Application Server Administration Console.

3. Expand **WebSphere Administrative Domain**.

4. Expand **Enterprise Application**.

5. Right-click your WebSphere Commerce application. For example, expand the **demo** application and select **Export Application**.

6. In the **Export directory** field, enter the following directory:

drive:\WebSphere\CommerceServer\working

2) Exporting configuration information for enterprise beans

To export this configuration information, do the following:

1. Copy the was.export.app.xml file from the following directory on the target WebSphere Commerce Server:

drive:\WebSphere\CommerceServer\xml\config

into the following directory on the target WebSphere Commerce Server:

drive:\WebSphere\CommerceServer\working

2. Open the was.export.app.xml file in a text editor. In this file replace all occurrences of \$Enterprise_Application_Name\$ with WebSphere Commerce Enterprise Application – *instanceName* where *instanceName* is the name of your WebSphere Commerce instance (for example, demo). Save this file.

Note: The value that you are inserting must match the information for your instance that is displayed in the WebSphere Application Server Administration Console.

3. At a command prompt, navigate to the following directory:

drive:\WebSphere\CommerceServer\working

4. Invoke the XMLConfig tool to perform a partial export by entering the following command:
xmlConfig -export OutputFile.xml -partial was.export.app.xml -adminNodeName *wasHostName*
where

wasHostName is the name of the node in the WebSphere Application Server that contains your current enterprise application.

OutputFile.xml is the name of the file that is created as a result of running this command and was.export.app.xml is file that you modified in step 2.

After you have exported the configuration information for each of the beans contained in the current enterprise application, you must add in a new stanza that describes each enterprise bean that you will add to your application. In addition, you must replace a variable in the configuration file to specify the exact name of the .ear file for your enterprise application.

To make these updates to the OutputFile.xml file, do the following:

1. Navigate to the following directory on the target WebSphere Commerce Server:

drive:\WebSphere\CommerceServer\working

2. Open the OutputFile.xml file in a text editor.

3. Locate the <ear-file-name> tag and replace the value with the following:

drive:\WebSphere\CommerceServer\working\WC_Enterprise_App_instanceName.ear

4. Locate the < jar-file > tag under <ejb-module name="WCSContract"> and replace the value with the following: **CustomizedWCSContractDeployed.jar**

5. We have created three new beans: Award, AwardCollectionTC and AwardRedemptionTC. Locate the <ejb-module-binding> tag under <ejb-module name="WCSContract">, and add in a new stanza for the each new bean as following.

```
<enterprise-bean-binding name="Award_Binding">  
  <jndi-name>instanceNameAward</jndi-name>  
</enterprise-bean-binding>  
<enterprise-bean-binding name="AwardCollectionTC_Binding">  
  <jndi-name>instanceNameAwardCollectionTC</jndi-name>  
</enterprise-bean-binding>  
<enterprise-bean-binding name="AwardRedemptionTC_Binding">  
  <jndi-name>instanceNameAwardRedemptionTC</jndi-name>  
</enterprise-bean-binding>
```

where

instanceName is the name of your WebSphere Commerce instance to which you are deploying the code.

6. Save the OutputFile.xml file.

Note: The <application-server-full-name> specification cannot span more than one line. Ensure WAS e-fix PQ58038 has been applied to your system. PQ58038 fixes the additional CRLF in the value of <application-server-full-name>.

3) Assembling the new enterprise beans into current enterprise application

To assemble the new EJB group into the enterprise application, do the following:

1. Backup the current enterprise application, by doing the following on the target WebSphere Commerce Server:
 - a. At a command prompt, navigate to the following directory:
drive:\WebSphere\CommerceServer\working
 - b. Make a copy of the existing *WC_Enterprise_App_instanceName.ear* file and name it *WC_Enterprise_App_instanceName.ear.bak*.
2. Open the WebSphere Application Server Administration Console.
3. From the **Tools** menu, select **Application Assembly Tool**.
4. Open the enterprise application upon which you are going to work, by doing the following:
 - a. From the **File** menu, select **Open**.
 - b. In the **File name** field, enter:
drive:\WebSphere\CommerceServer\working\WC_Enterprise_App_instanceName.ear and click **Open**. Wait for the application to open before continuing to the next steps. This takes several minutes.
5. Click **EJB Modules**. The pane on the right displays the EJB modules in the enterprise application.
6. Click **WCSCContract**.
7. Click the General tab to view the class path information. Copy this existing class path information into a text file.
8. Right-click the EJB module and select **Delete**.
9. Right-click **EJB Modules** and select **Import**.
10. In the **File name** field, enter the following:
drive:\WebSphere\CommerceServer\temp\CustomizedWCSCContractDeployed.jar
Click **Open**, then in the Confirm Values window, click **OK**.
11. Once the CustomizedWCSCContractDeployed.jar file is imported, scroll to the WCSCContract EJB group and select this group. Information about this group is shown in the pane on the right.
12. Open the text file containing the class path information for the previous version of the EJB group. Select and copy the class path.
13. In the **Classpath** field for the modified EJB group, enter *lib/award_ejbimpl.jar*, followed by the original class path information from previous step.
14. Click **Apply**.
15. Add the implementation JAR file for your EJB group to the application, by doing the following:
 - a. Right-click the **Files** node for your enterprise application and select **Add Files**. (The **Files** node for the enterprise application is located near the bottom of the hierarchical tree. Note that there are other Files nodes for components within the enterprise application, but you must select the Files node for the whole application.)
 - b. In the Add Files window, click **Browse**.
 - c. Navigate to the following directory:
drive:\WebSphere\CommerceServer\temp
 - d. With this directory highlighted, click **Select**.
 - e. Return to the Add Files window. Notice that the contents of the temporary directory are displayed. Highlight the **lib** directory. The contents of the lib directory are displayed in the pane on the right.
 - f. In the pane on the right, select the file **award_ejbimpl.jar** and click **Add**. The file is then shown in the Selected Files pane.
 - g. Click **OK**.
16. From the **File** menu, select **Close**.
17. Wait for the file to close, then from the **File** menu, select **Open** and reopen the *WC_Enterprise_App_instanceName.ear* file.
18. We have 3 new EJBs, Award, AwardCollectionTC and AwardRedemptionTC. For each of these entity beans, configure security for your entity bean, by doing the following:
 - a. With the EJB Modules node expanded, locate and expand the WCSCContract node.
 - b. Expand **Entity Beans**.
 - c. Expand *yourEntityBean* where *yourEntityBean* is the name of your entity bean.
 - d. Click **Method Extensions**, then in the pane on the right do the following:
 - 1) Click the **Advanced** tab.
 - 2) Ensure that **Security identity** is selected.
 - 3) For each method, ensure that **Use identity of EJB server** is selected.
 - 4) Click **Apply** (if you have made any modifications).
 - e. In the left navigation pane, right-click **Security Roles** under the WCSCContract EJB group and select **New**, then do the following:

In the **Name** field, enter `WCSecurityRole` and click **Apply**. Note, if this role exists already, you do not need to perform this step.

f. In the left navigation pane, right-click **Method Permissions** under the `WCContract EJB` group and select **New**, then do the following:

- 1) In the **Method Permission Name** field, enter `WCMethodPermission`
- 2) In the **Methods** selection area, click **Add**. The Add Methods window opens.
- 3) Expand **CustomizedWCContractDeployed.jar**, then *yourEntityBean* (e.g. Award) and then expand each of the **Home** and **Remote** lists of methods.
- 4) Hold the Shift key and select all of the home methods and click **OK**.
- 5) Repeat the method selection process to add the remote methods as well (if remote methods exist).
- 6) In the Roles selection area, click **Add**, select the `WCSecurityRole` and click **OK**.
- 7) Click **Apply**.

19. From the **File** menu, select **Save**.

20. Close the Application Assembly Tool.

After this step has completed, you have created a new enterprise application that contains all of the previous logic as well as your new business logic. This is all contained in the newly modified `WC_Enterprise_App_instanceName.ear` file.

4) Stopping and removing the current enterprise application

1. Open the WebSphere Application Server Administration Console.
2. Expand **WebSphere Administrative Domain**.
3. Expand **Nodes**.
4. Expand *nodeName* (where *nodeName* is the name of your node).
5. Expand **Application Servers**.
6. Right-click your WebSphere Commerce application server. For example, right-click the **WebSphere Commerce Server - nodeName** and select **Stop**.
7. Expand **Enterprise Applications**.
8. Right-click your WebSphere Commerce application. For example, right-click the **WebSphere Commerce Enterprise Application - instanceName** application and select **Stop**.
9. Right-click your WebSphere Commerce application. For example, right-click the **WebSphere Commerce Enterprise Application - instanceName** application and select **Remove**.
10. When prompted to indicate if the application should be exported, select **No**.
11. When prompted to indicate if the application should be removed, select **Yes**.

5) Importing the new enterprise application

Invoke the XMLConfig tool to import the enterprise application into WebSphere Application Server by entering the following command:

```
xmlConfig -import OutputFile.xml -adminNodeName wasHostName
```

where

- *wasHostName* is the name of the node in the WebSphere Application Server that contains your current enterprise application.
- *OutputFile.xml* is the XML file that describes all of your enterprise beans.

6) Starting the new enterprise application

1. Open the WebSphere Application Server Administration Console.
2. Expand **WebSphere Administrative Domain**, then **Nodes**, then *nodeName*
3. Highlight the *nodeName* node.
4. Click the **Refresh selected subtree** icon.
5. Start your WebSphere Commerce application by doing the following:
 - Expand **Application Servers**.
 - Right-click your WebSphere Commerce application. For example, right-click the **WebSphere Commerce Server - instanceName** and select **Start**.

Directory Structure:

Following is directory structure for the application:

```

WCS_Ref_Apps
  AwardTC_RefApp
    driver
      ear
        properties
          com
            ibm
              commerce
                tools
                  contract
                    properties
      wcstores.war
      ToolTech
      WEB_INF
      lib
      wctools.war
      javascript
      tools
      contract
      tools
      contract
  wcs
    schema
    db2
    temp
    lib
    xml
    policies
      xml
    tools
      contract
    trading
  java
    com
      ibm
        commerce
          contract
            objimpl
      mycompany
        award

```

Directory	Description	Contents
WCS_Ref_Apps	This is the root directory.	AwardTC_RefApp directory Readme.doc
AwardTC_RefApp	This is the root directory for the sample application.	driver directory java directory
driver	This directory contains the assets that will be built into the installable archive.	ear directory wcs directory
ear	This directory contains the assets that will be copied to the installed commerce enterprise	properties directory wcstores.war directory wctools.war directory

	archive directory.	
properties	This directory contains the property file that will be copied to the installed commerce enterprise archive directory.	award_en_US.properties ContractRB_en_US.properties (in a qualified structure: com.ibm.commerce.tools.contract)
wcstores.war\ToolTech	This directory contains ToolTech store specific .jsp files	Shoppingcart.jsp Award.jsp
wcstores.war\WEB_INF\lib	This directory contains JAR file for new command and databean logic	award_cmd_databean.jar
wctools.war	This directory contains the commerce accelerator java script and .jsp files that will be copied to the installed commerce enterprise archive directory.	javascript\tools\contract\Contract.js javascript\tools\contract\Extensions.js tools\contract\AwardCollectionTC.jsp tools\contract\AwardRedemptionTC.jsp tools\contract\Contract.jsp
wcs	This directory contains assets that need to be installed in the root directory of IBM WebSphere Commerce.	schema directory temp directory xml directory
schema	Assets required populating the database.	<i>db2/award.sql</i> - This script used to populate the commerce database with configuration information for this application. <i>db2/Tooltech.sql</i> - This script used to populate the commerce database with ToolTech specific data.
temp	The directory contains the ejb deployed code and implementation code which will be assembled to the commerce enterprise application.	<i>CustomizedWCSContractDeployed.jar</i> - the JAR file that contains the deployed code for the WCSContract EJB group for this application. <i>lib\award_ejbimpl.jar</i> - the JAR file for WCS Contract EJB implementation code for this application.
xml	This directory contains XML document.	<i>policies\xml\award_ac_policy.xml</i> - the xml file that contains access control policies for the new resources introduced in this application. <i>tools\contract\contractNotebook.xml</i> - modified contract notebook (including new pages). <i>trading\B2BTrading.dtd</i> - modified contract dtd (including the new awardTC).
java	This directory contains the Java source.	Java source files are located in directories that model their package names.

Design and Implementation Overview:

Objective

To design and implement an Award T&C reference application, demonstrating an extended WCS 5.4 BE contract model.

Feature Scenario

This demo application shall allow shoppers to collect and redeem award points.

User Cases:

- A shopper comes in, places an order under a contract. When he checks out, based on how much he buys, award points are calculated under his name and current contract. Award points will be deposited to the shopper once the payment has been captured.
- A shopper comes in, places an order under a contract. When he checks out, he has a choice to pay with his award points.

Design Assumption

- No combination of multiple payment methods is supported. That is, a shopper can choose to pay with award only if his collected award points are able to cover the entire order.
- Return and refund using award points are not supported.
- Currency conversion using award points is not supported.

Design

This application is designed and packaged on top of WCS 5.4 BE, and demonstrate how a new TC and policy can be incorporated into the contract model.

The new TC is "AwardTC" and it has two subtypes - "AwardCollectionTC" and "AwardRedemptionTC". Two new EJBs are used to access data defined in the award terms and conditions. The new policy defined in the sample is "PayWithAward". Its logic is implemented by a new set of payment policy commands. New JSP pages for AwardCollectionTC and AwardRedemptionTC are created under contract user interface in WebSphere Commerce Accelerator. Some existing controller commands are customized to accommodate the support of new TC and policy during runtime. A shopper's current award points are stored in a new table called "award" and a new EJB is used to access this table.

Once installed, contract creator can use WebSphere Commerce Accelerator (or contract import command) to create and modify contracts that have the award Terms and Conditions. Shopper can then shop under the contract to collect and redeem award points.

Functional Design

1) B2Btrading.dtd

New terms and conditions are defined in B2Btrading.dtd:

```
<!ELEMENT TermCondition
(TermConditionDescription?,Participant*,CreateTime?,UpdateTime?,(PriceTC|ProductSetTC|Shipping
TC|FulfillmentTC|PaymentTC|ReturnTC|InvoiceTC|RightToBuyTC|ObligationToBuyTC|PurchaseOrder
TC|OrderApprovalTC|DisplayCustomizationTC|OrderTC|AwardTC))>
<!ELEMENT AwardTC (AwardCollectionTC|AwardRedemptionTC)>
<!ATTLIST AwardCollectionTC
value CDATA #REQUIRED
currency CDATA #REQUIRED
point CDATA #REQUIRED
>
<!ATTLIST AwardRedemptionTC
value CDATA #REQUIRED
```

currency CDATA #REQUIRED
 point CDATA #REQUIRED
 >

2) Data and Object Model

Overall, we define:

1 new table - *AWARD*

1 new TC type - *AwardTC*

2 new TC subtype - *AwardCollectionTC* and *AwardRedemptionTC*

1 new payment policies - "*Pay with Award*" which implements the 5 payment policy command interfaces to process the payment with shopper's award

Table *AWARD*:

<i>Column Name</i>	<i>Column Type</i>	<i>Column Description</i>
MEMBER_ID	BIGINT NOT NULL	Foreign key to MEMBER table. Combined with TRADING_ID to be the primary key.
TRADING_ID	BIGINT NOT NULL	Foreign key to TRADING table. Combined with MEMBER_ID to be the primary key.
POINT	INT	The current award points shopper has collected under the contract.

3) Runtime Model

- During collection time: Extend *DoDepositWithAwardCollectionCmdImpl* from *DoDepositCmdImpl*. Award points are collected in method *DoDepositWithAwardCollectionCmdImpl::performExecute()*. The collection is actually happened once the payment is captured.
- During redemption time: (In ToolTech store, all available payment methods under current contract will be displayed on the checkout page.) If a shopper chooses to pay with his award, the new set of payment policy commands will be invoked to handle the payment process. The five new payment commands are:
 - *CheckPaymentAcceptAwardCmdImpl.java*: Return true.
 - *DoPaymentAwardCmdImpl.java*: Subtract award points if the collected award points are able to pay for the entire order.
 - *DoDepositAwardCmdImpl.java*: Do nothing. Logic is covered in *DoPaymentAwardCmdImpl*
 - *DoCancelAwardCmdImpl.java*: Rollback the points deducted.
 - *DoRefundAwardCmdImpl.java*: Refund the points.

4) User Interface – Commerce Accelerator Tools

WebSphere Commerce Accelerator has been updated in order to create new contracts that include the new award terms and conditions.

- Modify the placeholder JavaScript file *Extensions.js* for the new terms and conditions.
- Create new JSP files *AwardCollectionTC.jsp* and *AwardRedemptionTC.jsp* for the user interface. Each of these pages will include a dynamic JavaScript object to capture the data from the input fields. Modify *ContractSummary.jsp* to including the new TC.
- Create two new TC DataBeans to encapsulate loading the data from Access Bean, or include the Access Bean directly on the page.
- Update the *ContractRB_en_US.properties* file to include the new resources
- Modify *ContractNotebook.xml* to include the new pages.

5) Access Control

- AwardDataBean

In this application we only allow a shopper to see his own award points, that is, only the beneficiary is allowed to display AwardDataBean. Here are the stanzas of our new access control policy.

```
<Policy Name="AllUsersDisplayAwardDataBeanResourceGroup"
  OwnerID="RootOrganization"
```

```
UserGroup="AllUsers"
ActionGroupName="DisplayDataBeanActionGroup"
ResourceGroupName="AwardDataBeanResourceGroup"
RelationName="beneficiary">
</Policy>
```

- New pages in the contract notebook in CSA
In commerce accelerator, we have added two new pages:
ContractAwardCollectionTCPanelView and ContractAwardRedemptionTCPanelView. We want them to have the same access control policy as others pages in the contract notebook, i.e., to be accessed by Sellers, SalesManagers and AccountRepresentatives.

6) Customization to WCS published store

This reference application is independent to store logic so a published store is not necessary for the implementation. However, a published *Tooltech* B2B store would help in understanding and utilizing the new functionality in the shopping flow. The modified *Shoppingcart.jsp* together with *award.jsp* will demonstrate shopping using the new payment method "pay with award" during order process.

Trademarks and Service Marks

The following are trademarks or registered trademarks of IBM Corporation in the United States and other countries:

AIX DB2 IBM VisualAge(R) for Java WebSphere

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

Windows, Windows NT and Windows 2000 are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

** © COPYRIGHT INTERNATIONAL BUSINESS MACHINES CORPORATION 2002

** ALL RIGHTS RESERVED.

Note to US Government Users -- Documentation related to restricted rights -- Use, duplication, or disclosure is subject to restriction set forth in GSA ADP Schedule Contract with IBM Corp.

End of document.