



# **BEA AquaLogic Commerce Services**

## **Getting Started Guide**

Version 6.0  
March 2008

## Copyright

Copyright © 1995-2008 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA SYSTEMS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks and Service Marks

Copyright © 1995-2008 BEA Systems, Inc. All Rights Reserved. BEA, BEA JRockit, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Interaction, BEA AquaLogic Interaction Analytics, BEA AquaLogic Interaction Collaboration, BEA AquaLogic Interaction Content Services, BEA AquaLogic Interaction Data Services, BEA AquaLogic Interaction Integration Services, BEA AquaLogic Interaction Process, BEA AquaLogic Interaction Publisher, BEA AquaLogic Interaction Studio, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Kodo, BEA Liquid Data for WebLogic, BEA Manager, BEA MessageQ, BEA Salt, BEA WebLogic Commerce Server, BEA AquaLogic Commerce Services, BEA WebLogic Communications Platform, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic Log Central, BEA WebLogic Mobility Server, BEA WebLogic Network Gatekeeper, BEA WebLogic Personalization Server, BEA WebLogic Personal Messaging API, BEA WebLogic Platform, BEA WebLogic Portlets for Groupware Integration, BEA WebLogic Real Time, BEA WebLogic RFID Compliance Express, BEA WebLogic RFID Edge Server, BEA WebLogic RFID Enterprise Server, BEA WebLogic Server Process Edition, BEA WebLogic SIP Server, BEA WebLogic WorkGroup Edition, BEA Workshop for WebLogic Platform, BEA Workshop JSP, BEA Workshop JSP Editor, BEA Workshop Struts, BEA Workshop Studio, Dev2Dev, Liquid Computing, and Think Liquid are trademarks of BEA Systems, Inc. Accelerated Knowledge Transfer, AKT, BEA Mission Critical Support, BEA Mission Critical Support Continuum, and BEA SOA Self Assessment are service marks of BEA Systems, Inc.

All other names and marks are property of their respective owners.

# Contents

<b>Getting Started with AquaLogic Commerce Services .....</b>	<b>1</b>
Installing AquaLogic Commerce Services .....	1
What AquaLogic Commerce Services installs .....	2
Options during installation .....	2
Where to find everything .....	3
Windows menus .....	3
Running AquaLogic Commerce Services on WebLogic Server .....	5
Single Sign-On .....	5
AquaLogic Commerce Services Integration with WebLogic Portal .....	6
Deploying AquaLogic Commerce Services within a WebLogic Portal Domain .....	6
Adding AquaLogic Commerce Services to a Domain .....	14
Creating a new domain using the AquaLogic Commerce Services template .....	14
Deploying to an existing domain using WLST .....	14
Further reading .....	16
Developing with WorkSpace Studio .....	16
Additional WebLogic Portal Integration Information .....	16
Implementing Unified User Profile (UUP) with WebLogic Portal .....	16
AquaLogic Commerce Services user profile property set .....	16
AquaLogic Commerce Services UUP shared library .....	16
UUP in WebLogic Portal Administration Console .....	17



# Getting Started with AquaLogic Commerce Services

This guide is designed to give you a starting point for getting up and running with AquaLogic Commerce Services. It includes the basic deployment steps to

- Installing AquaLogic Commerce Services
- Deploying SSO and UUP with AquaLogic Commerce Services
- Deploying the sample storefront within a Portal domain and integrating with Portal
- Adding AquaLogic Commerce Services to a domain

## Installing AquaLogic Commerce Services

BEA AquaLogic Commerce Services is very easy to install as it provides an automated installer which places the application in your existing BEA WebLogic Server installation.

You will need to download the installer, as well as the license file. To update the license, copy the license file `sdk_license.bea` to `<WL_HOME>`, where `<WL_HOME>` refers to the path where BEA products are installed. From a command prompt, type:

- `cd c:\bea <enter>`
- `updatelicense sdk_license.bea <enter>`

## ***What AquaLogic Commerce Services installs***

The AquaLogic Commerce Services installation includes the following:

<b>Module</b>	<b>Description</b>
AquaLogic Commerce Services Template	A WebLogic domain template for easily creating new AquaLogic Commerce Services deployments that includes a preconfigured domain with AquaLogic Commerce Services application deployed.
AquaLogic Commerce Services Sample	A sample deployment of AquaLogic Commerce Services with a preconfigured demo store
AquaLogic Commerce Services WLP Plugins	UUP Plug-in modules for integrating AquaLogic Commerce Services with WebLogic Portal to integrate with Portal user profile management
AquaLogic Commerce Services Workshop Plugin	<p>Plug-in for Workshop that enables AquaLogic Commerce Services facets to be used in Workshop. These facets will enable development and integration with AquaLogic Commerce Services in the following areas</p> <ul style="list-style-type: none"> <li>• ALCS Core: includes all AquaLogic Commerce Services core components required for integration and development with the core library.</li> <li>• ALCS Storefront: adds to your Enterprise Application the sample Storefront Web Application.</li> <li>• ALCS WebServices Server: adds to your Enterprise Application the WebServices Server that exposes many services through its SOAP and REST interfaces.</li> <li>• ALCS WebServices Client: adds to your project the client library that allows development of applications integrating with AquaLogic Commerce Services through web services.</li> <li>• ALCS Content Management Integration: adds the required libraries and dependencies to your Enterprise Application to enable integration of AquaLogic Commerce Services catalogs with the WebLogic Portal Content Repository (SPI)</li> <li>• ALCS Portlets : adds sample portlets to your WAR project that demonstrates how to leverage the content management integration, and web services integration to create portlets for WebLogic Portal</li> </ul>

## ***Options during installation***

The installation tries to find the root folder of your BEA installation to use to install AquaLogic Commerce Services, but it also provides an option to choose the installation folder should you have an unusual configuration.

The installer also allows you to customize which of the modules gets included in the installation.

## ***Where to find everything***

The installer will place AquaLogic Commerce Services in the directory selected during installation, e.g. <WL\_HOME>/commerce\_6.0, where <WL\_HOME> refers to the path where BEA WebLogic Server is installed (such as C:\BEA\WebLogic).

The following components will be installed in directories beneath this directory:

- **database-scripts** - scripts for setting up the AquaLogic Commerce Services schema in the supported databases.
- **portal-plugins** – UUP plug-in for integration with WebLogic Portal.
- **uninstall** - allows you to uninstall AquaLogic Commerce Services.
- **samples/commerce/commerceApp** - the sample application including Commerce Storefront, Commerce Manager and Commerce Integration.
- **samples/domains/commerce** - the domain for the sample application.
- **common/templates/applications/commerce-services.jar** - the template to use for your own domain. Includes the ALCS application.
- **eclipse** – facets plugin to be used with Workshop. This needs to be copied to your Workshop plugins folder.

In addition, the necessary integration files for development with AquaLogic Commerce Services will be installed in directories under <wl\_server \_release#>/**samples**.

## ***Windows menus***

The Windows installer adds an ALCS 6.0 menu to the BEA Products > Examples menu of the Start > Programs menu, containing the following links:

- Start Commerce Services Example Server
- Stop Commerce Services Example Server
- Run Commerce Manager
- Visit Storefront Example

## ***Linux symbolic links***

The Linux installer adds the following symbolic links to your home directory to make it easy to start the examples:

- startCommerceServicesExampleServer
- stopCommerceServicesExampleServer

## Getting Started with AquaLogic Commerce Services

- Run Commerce Manager
- Commerce Services Sample

## Running AquaLogic Commerce Services on WebLogic Server

Running AquaLogic Commerce Services on WebLogic Server allows you to utilize advanced functionality, such as Single Sign On (SSO) and Unified User Profile (UUP) to share customer data and facilitate logins across WebLogic applications.



### Deploying "Out of the Box"

Many first time users want to deploy AquaLogic Commerce Services directly from the installer to ensure they can do this successfully before further trying out integrations to other applications. To do this, simply run the installation package and follow the on screen instructions. Installation will copy a preconfigured domain along with a populated database to get you up and running quickly.

You can start the domain by the installed menu shortcuts or by running the startup script.

i.e. c:\bea\_home>\commerce\_6.0\samples\domains\commerce\startWebLogic.cmd

## Single Sign-On

AquaLogic Commerce Services provides Single Sign-On (SSO) integration with other WebLogic Server and WebLogic Portal applications that are running in the same domain.

You may need to add the following section to your web application's WebLogic.xml to ensure your application uses the same cookie as AquaLogic Commerce Services:

```
<session-descriptor>
  <cookie-name>JSESSIONID</cookie-name>
  <cookie-path>/</cookie-path>
</session-descriptor>
```

Or if the xml uses the wls namespace:

```
<wls:session-descriptor>
  <wls:cookie-name>JSESSIONID</wls:cookie-name>
  <wls:cookie-path>/</wls:cookie-path>
</wls:session-descriptor>
```



### Warning

Configuring Single Sign-On to use an authentication database other than the one provided by AquaLogic Commerce Services will interfere with the ability to import customer profiles provided by the Commerce Manager application.

Users will still be able to import customer profiles through the Commerce Manager, but these imports will have no effect.

For more information on multi domain SSO setup, refer to the following BEA documentation:

<http://dev2dev.bea.com/pub/a/2006/12/sso-with-saml.html>

<http://edocs.bea.com/wls/docs100/secmanage/saml.html>

## AquaLogic Commerce Services Integration with WebLogic Portal

AquaLogic Commerce Services can run stand-alone on WebLogic Server, with no requirement for WebLogic Portal. However, if you use WebLogic Portal, AquaLogic Commerce Services provides a number of valuable integrations which will enable you to get even more out of both products. These integrations include SSO and UUP (from the WLS integration), as well as a Content Management SPI to share catalog data from AquaLogic Commerce Services to WebLogic Portal Virtual Content Repository, sample portlets and development facets.

For the purposes of this Getting Started Guide, we will step you through running AquaLogic Commerce Services and WebLogic Portal on the same domain. This allows seamless Single Sign-on between AquaLogic Commerce Services and WebLogic Portal applications as well as providing WebLogic Portal with access to the AquaLogic Commerce Services Customer profile data through the Unified User Profile (UUP). However, if, in production, you would like to run AquaLogic Commerce Services and WebLogic Portal in separate domains, you can use SSO via SAML to integrate the two.

## Deploying AquaLogic Commerce Services within a WebLogic Portal Domain

The following sequence of steps will guide you through the deployment of AquaLogic Commerce Services in a new WebLogic Portal domain, and demonstrate the integration of the Portal integration points including SSO, WebServices, and included Sample Portlets.

### ***Creating a new domain***

1. Install ALCS6 and copy the license file to <BEA\_HOME>
2. This creates an ALCS WLS domain under <BEA\_HOME>/commerce\_6.0/samples/domains/commerce.
3. Run the update license script (.sh in \*nix, and ..cmd in Windows)  
<BEA\_HOME>/UpdateLicense.sh <NAME\_OF\_LICENSE>
4. Create a new Portal domain in which to run ALCS by run the configuration wizard from the start menu.
5. Create a new Portal domain called "alcs" using the JRockit JVM. You can skip all other configuration at this point.

**Modify the domain's environment**

1. Add the stringtemplate jar file needed by the promotion rules engine to the WebLogic `pre_classpath` to replace WebLogic's older libraries. Copy `<BEA_HOME>/commerce_6.0/samples/domains/commerce/lib/stringtemplate-stringtemplate-2.3b6.jar` to `$BEA_HOME/user_projects/domains/alcs/lib`
2. Edit `<BEA_HOME>\user_projects\domains\alcs\bin\setDomainEnv.cmd` by changing this line:

```
PRE_CLASSPATH=""
```

to this:

```
PRE_CLASSPATH=%DOMAIN_HOME%\lib\stringtemplate-stringtemplate-2.3b6.jar
```

**Configure the domain for SSO**

1. Copy the `<BEA_HOME>\commerce_6.0\samples\domains\commerce\jaaslogin.config` file to `<BEA_HOME>\user_projects\domains\alcs\`
2. Edit `<BEA_HOME>\user_projects\domains\alcs\bin\setDomainEnv.cmd` by appending this line to the file:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% %JAVA_PROPERTIES% -  
Djava.security.auth.login.config=jaaslogin.config -Dderby.system.home=%DOMAIN_HOME%
```

**Copy the required libraries**

1. Copy the derby database used by ALCS6 into the portal domain's root directory `<BEA_HOME>\commerce_6.0\samples\domains\commerce\commerce_sample` (copy entire folder and everything underneath) to `<BEA_HOME>\user_projects\domains\alcs`
2. ALCS requires Java Advanced Imaging (JAI) libraries. Ideally these would be installed into the JVM that's running ALCS, but a simpler alternative is to add the libraries to the domain's lib directory. ALCS also needs the derby drivers to access the sample database  
Copy all the libraries in the sample domain into your portal domain's lib directory `<BEA_HOME>/commerce_6.0/samples/domains/commerce/lib/*.*.jar` to `<BEA_HOME>/user_projects/domains/alcs/lib/`

**Configure the Web Services**

1. Turn off authentication for webservices entirely for now. Parts can be added later as needed.

Open up `acegi.xml` in your application directory

`commerceServicesWS/WEB-INF/conf/spring/security/acegi.xml`

e.g. commerce\_6.0/samples/commerce/commerceApp/CommerceServicesWS/WEB-INF/...

2. Under FILTER CHAIN, change the line to

```
VA.*\Z=channelProcessingFilter
```

And change the section under HTTP CHANNEL REQUIREMENTS to:

```
VA.*\Z=REQUIRES_INSECURE_CHANNEL
```

3. Add the following to the Web Services web.xml located under your application directory:

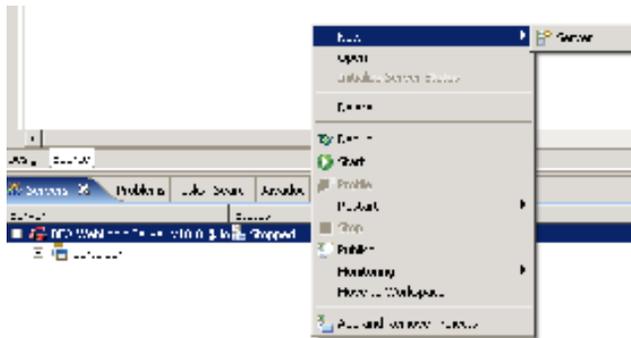
commerce\_6.0\samples\commerce\commerceApp\commerceServicesWS\WEB-INF\web.xml

before the end of the file, but before the </web-app> closure

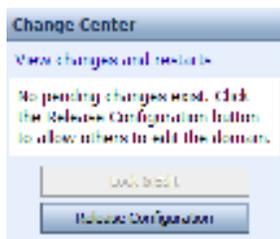
```
<login-config>  
  <auth-method>CLIENT-CERT</auth-method>  
</login-config>
```

### **Configure the Domain Database connection**

1. We will use the existing sample database from the installation.
2. Start the ALCS WLP domain by starting WorkSpace Studio, create a new server (in the bottom, middle panel, click on "Servers", right click in the white space, select "New", then "Server"), select the version of WebLogic Server you are using, then select the "alcs" domain you created.



3. Login to WebLogic Server Console (<http://localhost:7001/console>) and set up a new JDBC Datasource for Derby (the packaged database in the ALCS install). To make changes in the console, you will have to click on "Lock and Edit" in the upper left hand side of the console.



- Click on Services (middle of home page)/JDBC/DataSources -> New
- Name = commercejndi
- JNDI Name = jdbc/epjndi
- Database Type = Derby
- Database Driver = Derby's Driver (Type 4) Versions:Any (this is NOT the default, although they look similar. It is the second choice in the drop down menu)

Name:

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name:

What database type would you like to select?

Database Type:

What database driver would you like to use to create database connections?

Database Driver:

- Click Next, Click Next
- Database Name: commerce\_sample
- Host Name: localhost
- All other fields on the Connection Properties page remain default (no username or password). Click Next.

On the "Test Database Connection" screen:

- Change the Driver Class Name to "org.apache.derby.jdbc.EmbeddedDriver"
- Change the URL to "jdbc:derby:commerce\_sample"
- Delete everything in the Properties box
- Change the Test Table Name to "TCMUSER"

## Getting Started with AquaLogic Commerce Services

**Driver Class Name:**

What is the URL of the database to connect to? The format of the URL is jdbc:derby:databaseName;create=true

**URL:**

What database account user name do you want to use to create database?

**Database User Name:**

What is the database account password to use to create database  
(Note: for secure password management, enter the password in the Password field in the Password Management page)

**Password:**

**Confirm Password:**

What are the properties to pass to the JDBC driver when creating connection?

**Properties:**

What table name or SQL statement would you like to use to test database connection?

**Test Table Name:**

-Click Test Configuration to ensure that the connection is set up correctly

Connection test succeeded.

**Create a New JDBC Data Source**

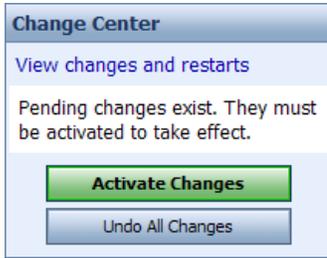
- Click Next, select AdminServer as the target for this datasource

**Create a New JDBC Data Source**

**Select Targets**  
You can select one or more targets to deploy your new JDBC data source. If you don't select a target, the data source will be created but not deployed. You will need to deploy the data source at a later time.

AdminServer

Select the new ALCS datasource by selecting the checkbox next to it.  
Select "Activate Changes" in upper left between each step.



4. The original ALCS6 installer sets all the ALCS application security configuration files (acegi.xml) to expect that they're running on ports 7011 and 7012.
  - o In the WLS console change the Environment (left hand menu)/Servers/AdminServer to use port 7011 instead of 7001
  - o Check the box next to "SSL Listen Port Enabled", then change the SSL listen port to 7012 instead of 7002
  - o Click "Save".

Click "Activate Changes" in the upper left.

Your browser should be directed to port 7011 automatically.

### **Install the ALCS EAR**

1. Click on "Deployments" on the left hand menu, then click on "install" and then navigate to the EAR file in  
 <BEA\_HOME>\commerce\_6.0\samples\commerce\commerceApp  
 (select the radio button next to "commerceApp" to select it)  
 Click Next.
2. Select "Install this deployment as an application". Accept all other defaults on the screen  
 Click Next  
 Select "I will make the deployments accessible from the following location ". Accept all other defaults on the screen.  
 Click Next  
 Click Finish  
 Click Activate Changes



You should now be able to see the commerceApp application listed in the "Deployments" window in either Prepared or Active state. If it's in Prepared state, start it.

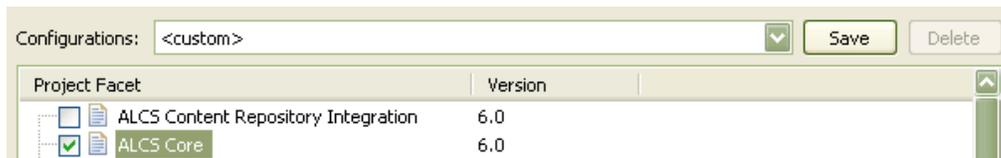
## Install the ALCS Facets plugin into Workspace Studio



### New to WebLogic Portal?

If you aren't yet an expert on setting up Portal EAR and Datasync projects, follow the steps in the Getting Started with WebLogic Portal tutorial at <http://edocs/wlp/docs102/tutorials>.

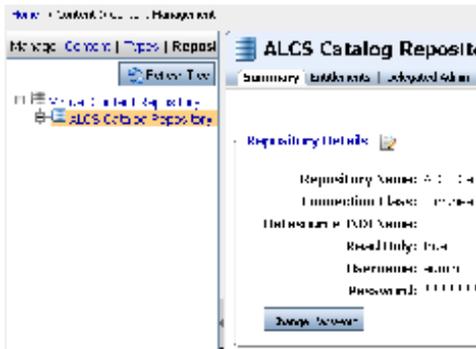
1. Stop the Server before continuing with this section.
2. Copy <BEA\_HOME>/commerce\_6.0/eclipse/plugins/ALCS6plugin\_1.0.0 to <BEA\_HOME>/workSpaceStudio\_1.1/workSpaceStudio/eclipse/plugins/
3. Create a new Portal EAR
4. Create a new Portal WAR, and add it to the EAR
5. Add the ALCS Core, ALCS Web Services Client, and ALCS Portlets facets to the WAR by
  - Right clicking on the web app
  - Select Properties > Project Facets
  - Click Add/Remove Project Facets button
  - Expand the "AquaLogic Commerce Services" list of facets to select these three facets
6. Add the ALCS Core and ALCS Content Repository Integration facets to the EAR (click on "AquaLogic Commerce Services" to select these two facets).



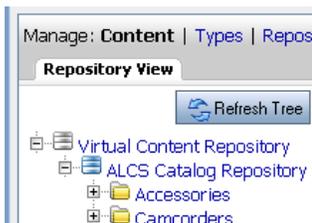
7. Right-click the EAR and select Run As... Run On Server
  - Select the BEA WebLogic Server v10.0 you created earlier
  - Click Next  
Make sure that the Portal EAR and Portal War are selected in the Configured Projects listbox
  - Click Finish
8. Go to the Deployments table, select the EAR, and select the Start button. This will change the state from "ready" to "active".
9. Add the following to your Portal War's WebLogic.xml file to enable SSO in your sample portlets:

```
<wls:session-descriptor>  
  <wls:cookie-name>JSESSIONID</wls:cookie-name>  
  <wls:cookie-path>/</wls:cookie-path>  
</wls:session-descriptor>
```

5. The CommerceWSServer.properties file configures the URL for the webservice server deployment. This is set to http://localhost:7011/webservices/ by default, which we don't need to change.
6. Connect to the Commerce Repository in the Portal Admin Tools
  - Launch the Portal Admin Tool from Workspace Studio
    - Go to the Run menu and select "Open Portal Administration Console".
    - Log in as weblogic/weblogic.
  - Navigate to the Content Management editor in the middle of the landing page.
  - Navigate to the "Repository" tab (the third tab)
  - Select the Repository in the tree (ALCS) and then the Login Options button will appear in the right hand side. From the Login dialog, select the "Set my credentials and log in" radio button. Fill in the username and password (admin/111111)".



- After successfully connecting to the Commerce Repository, navigate to the "Content" tab (the left-hand tab) you will see the catalog categories and products populated from the sample AquaLogic Commerce Services store, Snap It Up (cameras, video recorders, etc) The connection of the catalog to the WLP Virtual Content Repository is now in place.



### Accessing the portlets

- To see the sample Shopping Cart portlet, navigate to <http://localhost:7011/<NAME OF YOUR WAR PROJECT>/com/bea/alcs/portlet/ShoppingCart.portlet>.  
If you login to the storefront and add something to your cart, then navigate to the portlet, it will show you what's in your cart. You may need to log out and log back in order to see the items show up in your cart.
- To see the sample order history portlet navigate to

<http://localhost:7011/<NAME OF YOUR WAR PROJECT>/com/bea/alcs/portlet/OrderHistory.portlet>

If you login to the storefront and create an order, then navigate to the portlet, it will show you what's in your order history.



**Warning**

Customer sessions will not be available through the SSO plugin immediately after the account has been created. A customer session will be available to the Portlets and other applications in the same domain after a customer explicitly logs into the AquaLogic Commerce Services storefront with an existing account.

## Adding AquaLogic Commerce Services to a Domain

It is quite easy to add the AquaLogic Commerce Services applications to a new or existing WebLogic Server domain. You may prefer to do this rather than make changes to the sample domain created by the installer.

### *Creating a new domain using the AquaLogic Commerce Services template*

You can use the WebLogic Configuration to create a new domain based on the AquaLogic Commerce Services domain template included in the installation.

Run the Configuration Wizard and select the **Create a new WebLogic Domain** option. On the Select Domain Source page, select the Base this domain on an existing template

Browse for the template which you can find at  
<WL\_HOME>/common/templates/applications/commerce\_services.jar.

You can then follow the rest of the configuration wizard as usual.

Once you have created your new domain based on this template you can run the configuration wizard again to extend that domain to include other BEA products if you have them (e.g. WebLogic Portal, Workshop for WebLogic, etc).

### *Deploying to an existing domain using WLST*

If you chose to install the AquaLogic Commerce Services sample application on your own domain then you can easily deploy the sample app into your domain using the following wlst command. You may need to change some values to match your server configuration.

```
commerceAppDir='<BEA_HOME>/samples/commerce/commerceApp'  
  
connect('WebLogic','WebLogic','localhost:7011')  
deploy(appName='commerceServicesApp',path=commerceAppDir,stageMode='nostage')
```

You will need to add a data source entry to your domain. This can be done through the WebLogic console or using wlst. An example script is shown below containing the settings for connecting to the sample derby database bundled with AquaLogic Commerce Services.

### createJNDI.py

```

dsName="commercejndi"
jndiName="jdbc/epjndi"

connect('WebLogic','WebLogic','localhost:7011')

edit()
startEdit()

jdbcSR = create(dsName,"JDBCSystemResource")
theJDBCResource = jdbcSR.getJDBCResource()
theJDBCResource.setName(dsName)

connectionPoolParams = theJDBCResource.getJDBCConnectionPoolParams()
connectionPoolParams.setConnectionReserveTimeoutSeconds(25)
connectionPoolParams.setMaxCapacity(100)
connectionPoolParams.setTestTableName("TCMUSER")

dsParams = theJDBCResource.getJDBCDataSourceParams()
dsParams.addJNDIName(jndiName)

driverParams = theJDBCResource.getJDBCDriverParams()
driverParams.setUrl("jdbc:derby:commerce_sample")
driverParams.setDriverName("org.apache.derby.jdbc.EmbeddedDriver")

driverParams.setPassword("")
driverProperties = driverParams.getProperties()

proper = driverProperties.createProperty("user")
proper.setValue("")

save()
activate()

```

If you used the above script to create your data source then you will need to edit its configuration in your WebLogic console in order to target it to the desired server (ex. AdminServer).

## ***Further reading***

More information on deploying AquaLogic Commerce Services can be found in the *AquaLogic Commerce Services Deployment Guide*.

## ***Developing with WorkSpace Studio***

- Read the BEA WorkSpace Studio documentation for more information on developing applications using WorkSpace Studio.
- Read the *AquaLogic Commerce Services Developer Guide* for additional information on enhancing your development environment with plug-ins to assist with making enhancements to the AquaLogic Commerce Services applications.
- The *AquaLogic Commerce Services Developer Guide* is also rich with information on making changes to the Storefront application, including tutorials. Give it a read and you will be making advanced modifications in no time!

## **Additional WebLogic Portal Integration Information**

### ***Implementing Unified User Profile (UUP) with WebLogic Portal***

AquaLogic Commerce Services includes a plug-in for WebLogic Portal that allows you to access AquaLogic Commerce Services Customer profile attributes using WebLogic Portal's Unified User Profile (UUP) functionality. For more information about UUP, please refer to the *AquaLogic Commerce Services Deployment Guide*.

### ***AquaLogic Commerce Services user profile property set***

The AquaLogic Commerce Services installation includes a property set file, `<INSTALL_DIR>/commerce/portal-plugins/uup/CommerceServices.usr`.

Assuming you have already set up a WebLogic Portal EAR Project and Datasync Project in Workshop for WebLogic, copy the `CommerceServices.usr` file to your Datasync project's userprofile folder.

### ***AquaLogic Commerce Services UUP shared library***

To use UUP in your portal project, the UUP shared library needs to be included. This can be done as follows

1. Enter WorkSpace Studio Project Explorer view (Window > Show View > Project Explorer) and expand Enterprise Applications.

2. Expand the node relating to your Portal EAR
3. Expand the WebLogic Deployment Descriptor node
4. Right-click WebLogic J2EE Libraries and choose Add
5. In the Add WebLogic J2EE Library reference window, click Browse
6. Look for alcs-uup-lib in the J2EE Library list. If you don't see it on the list add it as follows
  - Click the Add button
  - In the Add WebLogic J2EE Library window, click Browse
  - Browse to <BEA\_HOME>/portal-plugins/uup
  - Select alcs-uup-lib-6.0.2.ear and click Open.
  - Click the OK button.
7. Select alcs-uup-lib from the list and click OK.
8. Click OK in the remaining open windows.

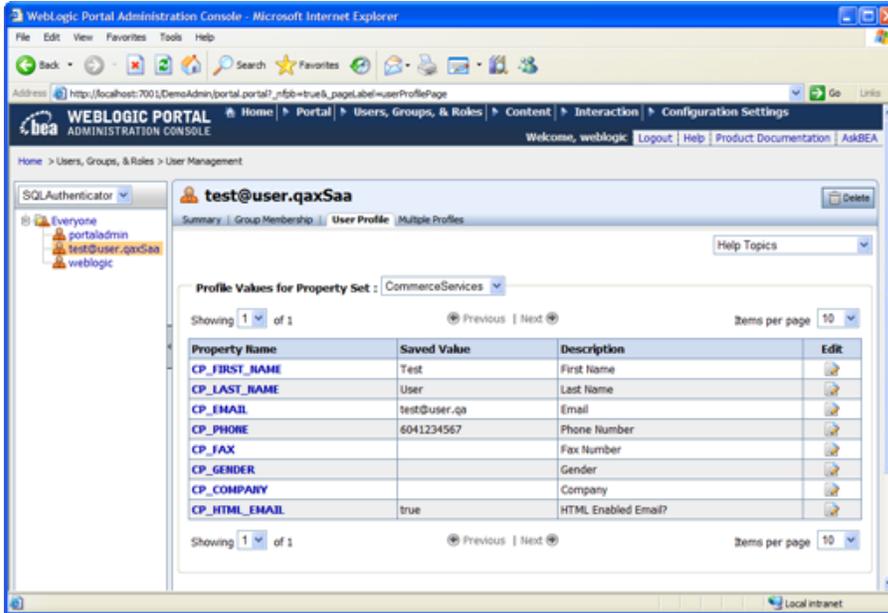
### ***UUP in WebLogic Portal Administration Console***

Once you deploy your WebLogic Portal project, you should be able to access the AquaLogic Commerce Services Customer profile data from the WebLogic Portal Administration Console.

1. Log in to the console.
2. Select User Management from the Users, Groups, & Roles menu to display the user management page.
3. From the list of users on the left, click on a user you created in Storefront.
4. Click on the User Profile tab over on the right
5. Ensure Profile Values for Property Set: is set to CommerceServices

You should now be able to see the Customer Profile properties similar to the following:

## Getting Started with AquaLogic Commerce Services



You can click on the Edit button to the right of any property to change the value of that property.



### A note about caching

The UUP property values are cached, so a change made to a profile property in Storefront make take a few minutes to appear in the User Profile view in the administration console. However, changes you make to properties within the administration console should be visible immediately.