

ellucian™

Banner Web Services Installation Guide

Release 8.1.3
January 2012



Banner®, Colleague®, PowerCAMPUS®, Luminis® and Datatel® are trademarks of Ellucian or its affiliates and are registered in the U.S. and other countries. Ellucian, Advance, DegreeWorks, fsaATLAS, Course Signals, SmartCall, Recruiter, MOX, ILP, and WCMS are trademarks of Ellucian or its affiliates. Other names may be trademarks of their respective owners.

©2012 Ellucian. All rights reserved. The unauthorized possession, use, reproduction, distribution, display or disclosure of this material or the information contained herein is prohibited.

Contains confidential and proprietary information of Ellucian and its subsidiaries. Use of these materials is limited to Ellucian licensees, and is subject to the terms and conditions of one or more written license agreements between Ellucian and the licensee in question.

In preparing and providing this publication, Ellucian is not rendering legal, accounting, or other similar professional services. Ellucian makes no claims that an institution's use of this publication or the software for which it is provided will guarantee compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Prepared by: Ellucian
4375 Fair Lakes Court
Fairfax, Virginia 22033
United States of America

Revision History

Publication Date	Summary
January 2012	New version that supports Banner Web Services 8.1.3 software.

Banner Web Services 8.1.3 Installation Guide

Contents

Chapter 1	Introduction	7
	What is a Web service?	7
	How do Banner Web services work?	7
	Example uses of Banner Web services	8
	Integration with campus card systems	8
	Integration with housing systems	8
	Supporting components	9
	Banner Translation Service	9
	Banner Web Services Adapters	9
	Banner Cardholder Event Publisher	10
	Installation overview	10
	Available documentation	11
Chapter 2	Install Banner Web Service Adapters	13
	Requirements	13
	Oracle application server and Java	13
	Oracle database	13
	Banner Translation Service	14
	Banner products	14
	Installation on Oracle Application Server 10.1.3.4/5	14
	Step 1 Install the adapter	15
	Step 2 Define the data source	18
	Step 3 Configure the security role and user	23
	Step 4 Enable schema validation (optional)	28

Step 5 Configure logging	29
Step 6 Verify the deployment	30
Installation on Oracle WebLogic Server 11g	31
Step 1 Configure the Oracle WebLogic Server	32
Step 2 Configure logging (optional)	34
Step 3 Define the data source	36
Step 4 Install the adapter	41
Step 5 Configure the security group and user	47
Step 6 Enable schema validation (optional)	53
Step 7 Verify the deployment	54
WSDL definitions	55
WSDLs for campus card Web services	55
WSDLs for housing Web services	55
Chapter 3 Verify the Configuration	57
Verification steps	57
Step 1 Download and install soapUI	57
Step 2 Open the testing workspace	58
Step 3 Import the soapUI project	59
Test results	60
No errors	60
Data source configuration errors	61
Banner Translation Service configuration errors	62
Chapter 4 Import Translations	65
Delivered translation files	65
Files for campus card systems	65
Files for housing systems	65
Import steps	66
Step 1 Extract Banner-specific translation values	66
Step 2 Import translation values	67



Chapter 5	Customize Web Service Responses	69
	Scripts that check configuration settings	69
	GORRSQL rules	70
	GTVSDAX settings	71
Chapter 6	Test Banner Web Services	75
	Test method	75
	Web services for campus card systems	75
	GetPersonIdentity	76
	GetEligibleCardholder	77
	Web services for housing systems	79
	GetPersonIdentity	79
	GetHousingApplicantEligibility	80
	GetHousingApplicantProfile	82
	GetAcademicPeriods	83
	AddEntityAddress	84
	ExpireEntityAddress	85
	AddStudentDeposit	87
	ReleaseStudentDeposit	88
	AddStudentAccountTransaction	90
Chapter 7	Install Banner Cardholder Event Publisher	93
	Requirements	93
	External campus card system	93
	Oracle application server and Java	93
	Oracle database	94
	Banner Translation Service	94
	Banner dependency	94
	Installation on Oracle Application Server 10.1.3.4/5	95
	Step 1 Verify the capture process rules	95

Step 2 Create, configure, and start the Oracle Streams processes.	95
Step 3 Define the data source for Oracle Advanced Queuing.	96
Step 4 Define the data source for the bulk load process.	97
Step 5 Define the data source for the Oracle Streams administrator.	102
Step 6 Install the Publisher	106
Step 7 Verify the instance configuration	110
Step 8 Configure the security role and user	110
Installation on Oracle WebLogic Server 11g	115
Step 1 Verify the capture process rules	115
Step 2 Create, configure, and start the Oracle Streams processes.	116
Step 3 Configure the Oracle WebLogic Server	116
Step 4 Define the data source for Oracle Advanced Queuing.	119
Step 5 Define the data source for the bulk load process.	119
Step 6 Define the data source for the Oracle Streams administrator.	125
Step 7 Install the Publisher	127
Step 8 Configure the security group and user.	133
Configuration.	139

Chapter 8 Test Banner Cardholder Event Publisher.....141

Setup and use of soapUI.	141
Step 1 Download and install soapUI	142
Step 2 Open the testing workspace.	142
Step 3 Import the soapUI project	143
Step 4 Start the MockService	144
Step 5 Send a test message.	147
Step 6 Add accessible URL for the MockService	150
Step 7 Reconfigure the Banner Cardholder Event Publisher	151
Step 8 Test the Banner Cardholder Event Publisher	152
Test cases.	152



1 Introduction

This chapter introduces Web services and describes how Banner® uses Web services.

What is a Web service?

A Web service exposes an application's processing logic to support a service-oriented architecture and to facilitate integration with external systems. A Web service allows an external system or business process to invoke the application's logic without having to understand the application's internal structure.

Web services are based on open, Internet-based standards. This makes them relevant to application integration within an organization and with external organizations. Standards such as XML, SOAP, WSDL, and UDDI provide cross-platform compatibility that does not depend on a single programming language or network transport.

How do Banner Web services work?

Java-based adapters expose Banner functions as Web services. This exposure makes the Banner functions available to external systems using the SOAP protocol over HTTP/HTTPS. External systems interact with the Web services, which in turn are supported by Banner APIs. This layered approach provides an insulating buffer between external systems and Banner. External systems do not interact with Banner directly, but rather exchange XML messages with the exposed Web services.

The Banner Web Services Adapters support the synchronous, request/reply message exchange pattern as follows:

1. The external system requests a service of Banner by sending an XML message to the Web service endpoint that is exposed by the adapter. The message contains the information required for Banner to service the request.
2. The Banner Web Services Adapter invokes the appropriate Banner API.
3. The Banner API performs the necessary Banner processing logic.

4. One of the following occurs:
 - 4.1. If the action is completed successfully, the API provides a response message, which the adapter forwards to the external system.
 - 4.2. If the action is not completed successfully, the adapter sends an error message (called a SOAP fault) to the external system.

Each Web service is independent. A business process, however, can invoke several Web services during the process.

The Banner Cardholder Event Publisher also uses Web services, in a different way, to integrate Banner with external systems. Rather than receive requests from external systems, the Banner Cardholder Event Publisher pushes data from Banner to external systems when specified data elements change. The Publisher provides near real-time publication of data for which Banner is authoritative.

Example uses of Banner Web services

Banner Web services connect external systems with Banner. These connections enforce vital business processes and allow data to be centrally maintained. Data from one system can be used as input to the other. Data updates in one system can initiate data update in the other.

Banner Web services support integration with campus card systems and housing systems. The available services, however, are not limited to these systems and can be used for other business needs.

Integration with campus card systems

Campus cards are electronic or magnetic cards that are issued to students, staff, faculty, and other constituents. Institutions use campus card systems to identify individuals, allow physical access to institution facilities, provide purchasing capabilities at campus point-of-purchase venues, and provide access to other institutional offerings such as libraries.

Web services can be used to provide real-time interaction between third-party campus card systems and Banner. Banner is the authoritative source for constituent information. Campus card systems are the authoritative source for card-access related data.

Integration with housing systems

Institutions use housing systems to accept applications for housing, assign applicants to residence halls and related services, and manage campus residence facilities.



Web services can be used to provide real-time interaction between third-party housing systems and Banner. Banner is the authoritative source for constituent information. Housing systems are the authoritative source for housing and resident-related data.

Supporting components

The following components work together to provide Web services-based integration with Banner:

- Banner Translation Service
- Banner Web Services Adapters
- Banner Cardholder Event Publisher

Banner Translation Service

Data in Banner is often constrained by lists of valid values. These valid values are stored in support tables and can be configured by your institution. As a result, the data might not be appropriate or usable by external systems. Banner Translation Service converts institution-specific data values in Banner to standard values that external systems can recognize and use.

Refer to the *Banner Translation Service Installation and Administration Guide* for more details.

Banner Web Services Adapters

The Banner Web Services Adapters expose Banner functions as Web services. An adapter can be configured to expose any number of defined Web services. Two sample configurations are provided as J2EE compatible enterprise archive files. One configuration exposes Web services that external campus card systems need to integrate with Banner. Another configuration exposes Web services that external housing systems need to integrate with Banner. These adapters are referred to as the Banner Web Services Adapter for Campus Card Systems and the Banner Web Services Adapter for Housing Systems, respectively.

The adapters refer to an XML document for configuration information. This in-memory singleton class is built from a preconfigured XML file (`process-config.xml`).

Configuration information includes the following elements:

- Data source that identifies the Banner database for performing database transactions
- Banner Translation Service lookup, regular expression, and delimiters that perform static and dynamic value translations

- Mappings for message types identified by the incoming root XML element to a PL/SQL packaged procedure to process the request
- List of XSL transformations used to convert UDC schema document instances to Banner schema document instances, and vice versa

Banner Cardholder Event Publisher

The Banner Cardholder Event Publisher publishes data from Banner to external systems when cardholder data changes in Banner tables. The following processing occurs:

1. Oracle Streams captures the table changes and publishes a corresponding Banner Identity event to the Campus Card Event Topic.
2. The Banner Cardholder Event Publisher reads events posted to this topic, retrieves Banner cardholder data, transforms the retrieved data to the proper format, and publishes SyncEligibleCardholder messages to a campus card system's exposed Web service endpoint that supports the SyncEligibleCardholder interface and SOAP binding.

The Banner Cardholder Event Publisher is delivered as a J2EE compatible enterprise archive file and works with the Banner Web Services Adapter for Campus Card Systems.

Installation overview

Your institution's specific integration requirements influence which components you install. Use the following steps to guide your installation.

1. Review the *Banner Web Services Handbook* for an overview of available Banner Web services and associated configuration requirements.
2. Decide what components need to be installed, based on your institution's needs.
3. Install the Banner Translation Service. Refer to the *Banner Translation Service Installation and Administration Guide* for details.
4. If needed, install and configure the appropriate Banner Web Services Adapters (see [Chapter 2, "Install Banner Web Service Adapters"](#)).
5. If Banner Web Services Adapters are installed, verify the configuration (see [Chapter 3, "Verify the Configuration"](#)).
6. Import translations for each deployed Banner Web Services Adapter into the Banner Translation Service (see [Chapter 4, "Import Translations"](#)).

7. Customize the exposed Web services for each deployed Banner Web Services Adapter (see [Chapter 5, “Customize Web Service Responses”](#) for an overview and the *Banner Web Services Handbook* for more details).
8. (Recommended) Test the installed Web services (see [Chapter 6, “Test Banner Web Services”](#)).
9. If needed, install the Banner Cardholder Event Publisher (see [Chapter 7, “Install Banner Cardholder Event Publisher”](#)).
10. If the Banner Cardholder Event Publisher is installed, test the deployment (see [Chapter 8, “Test Banner Cardholder Event Publisher”](#)).

Available documentation

This installation guide provides detailed information on installing Banner Web Service Adapters and the Banner Cardholder Event Publisher.

The following documents provide more information on Banner Web services:

- The *Banner Web Services Handbook* describes the messages, message mapping to Banner, intended usage, setup requirements, and translations for the Banner Web services that support integration with campus card systems and housing systems.
- The *Banner Translation Service Installation and Administration Guide* provides information on installing and administering the Banner Translation Service, a prerequisite component for exposing the Banner Web services.



2 Install Banner Web Service Adapters



The Banner® Web Services Adapters expose Banner functions as Web services. Two sample configurations are provided as J2EE compatible enterprise archive files. One configuration exposes Web services that external campus card systems need to integrate with Banner. Another configuration exposes Web services that external housing systems need to integrate with Banner.

This chapter gives instructions for installing the adapters on Oracle Application Server 10.1.3.4/5 and Oracle WebLogic Server 11g. This chapter also lists the URLs that expose the WSDL (Web Services Description Language) files that define the Web services exposed by the Banner Web Services Adapters.

Requirements

The Banner Web Services Adapters require the following components.

Oracle application server and Java

The Banner Web Services Adapters are certified on Oracle Application Server (OAS) 10.1.3.4/5 and Oracle WebLogic Server 11g with Java 1.6.

OAS 10.1.4/5.x is delivered with Java 1.5. The following Oracle document provides instructions for changing to Java 1.6. If you contract with Ellucian for Oracle support, you can access the FAQ on the Customer Support Center. Otherwise, you can use your Oracle support account to access the document.

Document Title:	How to change the Java version used to run a specific OC4J instance
Ellucian FAQ:	1-AXZ803
Oracle Doc ID:	351476.1

Oracle database

The required Oracle database depends on the application server that you are using:

Application Server	Required Database
Oracle Application Server 10.1.3.4/5	Oracle Database 10gR2 or 11g
Oracle WebLogic Server 11g	Oracle Database 11g

Banner Translation Service

You *must* install the Banner Translation Service *before* you deploy the Banner Web Services Adapters. Refer to the *Banner Translation Service Installation and Administration Guide* for details.

Banner products

The Banner Web services are an interface layer to Banner APIs. The following Banner products support the Banner Web Services Adapters and must be installed:

Banner Web Services Adapter for Campus Card Systems

Product	Minimum Version - 7.x	Minimum Version - 8.x
Banner General	7.3	8.0 plus patch p1-46c8mj_gen80100

Banner Web Services Adapter for Housing Systems

Product	Minimum Version - 7.x	Minimum Version - 8.x
Banner General	7.3	8.0 plus patch p1-46c8mj_gen80100
Banner Student	7.3	8.0
Banner Accounts Receivable	7.2	8.0

Installation on Oracle Application Server 10.1.3.4/5

The adapters are packaged as J2EE compatible enterprise archive files. Each file must be deployed and configured separately:

- `CampusCardIntegration_v8.1.3.ear` exposes Web services that external campus card systems need to integrate with Banner.

- HousingIntegration_v8.1.3.ear exposes Web services that external housing systems need to integrate with Banner.

Use the following steps to install each adapter on OAS 10.1.3.4/5:

- [Step 1, “Install the adapter”](#)
- [Step 2, “Define the data source”](#)
- [Step 3, “Configure the security role and user”](#)
- [Step 4, “Enable schema validation \(optional\)”](#)
- [Step 5, “Configure logging”](#)
- [Step 6, “Verify the deployment”](#)

A new OC4J instance was created when the Banner Translation Service was installed. Banner Web Service Adapters should be deployed in this instance. Dedicate a separate OC4J instance for the Banner Translation Service and Banner Web Services Adapters so they can be independently managed.

Step 1 Install the adapter

Before beginning this step, you must understand the concepts published by Oracle regarding the deployment of ear files.

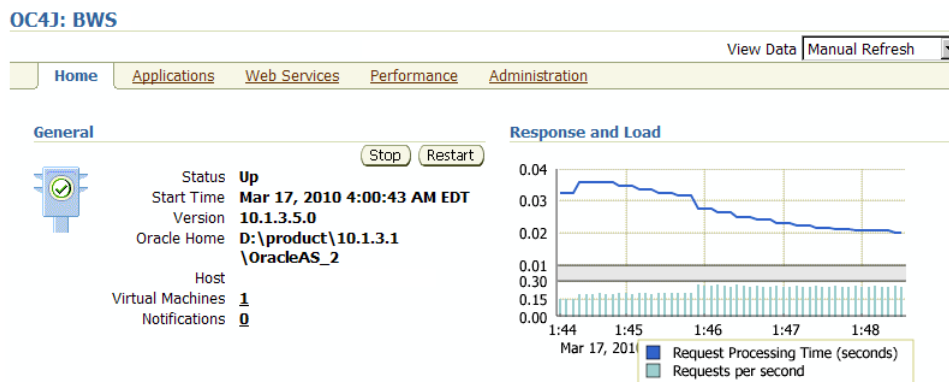
Use the following steps to install the adapter to the Oracle Application Server.

1. Connect to the Oracle Enterprise Manager:

`http://<host>:<port>/em`

The console is displayed.

2. Click the name of the OC4J instance where the Banner Translation Service is deployed. The Home page for the selected instance is displayed.



3. Select the **Applications** tab. A list of deployed applications is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration

This page shows the J2EE applications and application components (EJB Modules, WAR Modules, Resource Adapter Modules) deployed to this OC4J instance.

View Applications

Start Stop Restart Undeploy Redeploy Deploy

Select All Select None Expand All Collapse All

Select	Name	Status	Start Time	Active Requests	Request Processing Time (seconds)	Active EJB Methods	Application Defined MBeans
<input type="checkbox"/>	▼ All Applications						
<input type="checkbox"/>	ascontrol	↓					
<input type="checkbox"/>	▼ default	↑	Apr 6, 2010 5:18:11 AM EDT	0	0.00	0	
<input type="checkbox"/>	Translation Service	↑	Apr 6, 2010 5:18:39 AM EDT	0	0.00	0	
<input type="checkbox"/>	► Middleware Services						

4. Click **Deploy**. The Deploy: Select Archive page is displayed.

Deploy: Select Archive

Cancel Step 1 of 3 Next

Archive

The following types of archives can be deployed: J2EE application (EAR files), Web Modules (WAR files), EJB Modules (EJB JAR files) and Resource Adapter Modules (RAR files).

Archive is present on local host. Upload the archive to the server where Application Server Control is running.

Archive Location Browse...

Archive is already present on the server where Application Server Control is running.

Location on Server
The location on server must be the absolute path or the relative path from j2ee/home

Deployment Plan

The deployment plan is an XML file that contains the deployment settings for an application. If you do not have a deployment plan, one will be created automatically during the deployment process. Later in the deployment process, you can optionally edit the deployment plan and save it for a future deployment of this application.

Automatically create a new deployment plan.
The deployment plan settings will be based on OC4J defaults and information contained in the archive

Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running.

Plan Location Browse...

Deployment plan is already present on server where Application Server Control is running.

Location on Server
The location on server must be the absolute path or the relative path from j2ee/home

Cancel Step 1 of 3 Next

5. Select the file to be uploaded:

5.1. In the Archive section, select **Archive is present on local host. Upload the archive to the server where Application Server Control is running.**

5.1. In the **Archive Location** field, click **Browse** and navigate to the appropriate ear file:

CampusCardIntegration_v8.1.3.ear

or
HousingIntegration_v8.1.3.ear

- 5.2. Select the file and click **Open**.
6. Select the deployment plan for the application:
 - 6.1. In the Deployment Plan section, select **Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running**.
 - 6.2. In the **Plan Location** field, click **Browse** and navigate to the appropriate deployment plan:

CampusCardIntegration_v8.1.3_plan.dat
or
HousingIntegration_v8.1.3_plan.dat
 - 6.3. Select the file and click **Open**.
7. Click **Next** on the Deploy: Select Archive page. The files are uploaded and the Deploy: Application Attributes page is displayed.

Deploy: Application Attributes

Cancel Back Step 2 of 3 Next

Archive Type **J2EE Application (EAR file)**
Archive Location **\\banner_web_services_8.1.3
\\campus_card_adapter\ear\CampusCardIntegration_v8.1.3.ear**
Deployment Plan **CampusCardIntegration_v8.1.3_plan.dat**

* Application Name
Parent Application default
Bind Web Module to Site default-web-site
Context Root

Web Module	Context Root
CampusCardIntegration_web.war	/campuscard

Cancel Back Step 2 of 3 Next

8. Enter a name for the application (for example, *Campuscard* or *Housing*) in the **Application Name** field.

Warning

If the adapter is being deployed in multiple OC4J instances on the same server, the application name must be unique for each deployment. ■

9. If the adapter is being deployed in one instance only, accept the default context root and go to step 10.

-or-

If the adapter is being deployed in multiple instances on the same server, add descriptive text to the default context root in the **Context Root** field (for example, /campuscard/test or /campuscard/prod). Adding descriptive text to the default

string, rather than changing the entire default string, is preferable. Use this new context root in any subsequent steps that refer to the default URL.

Warning

If the adapter is being deployed in multiple OC4J instances on the same server, the context root must be unique for each deployment. ■

10. Click **Next**. The Deploy: Deployment Settings page is displayed.

Deploy: Deployment Settings

Step 3 of 3

Archive Type **J2EE Application (EAR file)**
 Archive Location **\banner_web_services_8.1.3**
 \campus_card_adapter\ear\CampusCardIntegration_v8.1.3.ear
 Deployment Plan **CampusCardIntegration_v8.1.3_plan.dat**

Application Name **Campuscard**
 Parent Application **default**
 Web Module to Site **default-web-site**
 Context Root **/campuscard**

Deployment Tasks

The table below provides a set of common deployment tasks you might want to perform for this application. Only those tasks that apply to the current application are enabled.

Task Name	Go To Task	Description
Map Environment References		Map any environment references in your application (for example, data sources) to physical entities currently present on the operational environment.
Select Security Provider		A security provider acts as the source for available users and groups when mapping security roles.
Map Security Roles		Map any security roles exposed by your application to existing users and groups. The list of users and groups is obtained from the security provider you selected for this application.
Configure EJBs		Configure the Enterprise JavaBeans in your application.
Configure Clustering		Configure clustering of your application.
Configure Class Loading		Manipulate the classpath of your application.

Advanced Deployment Plan Editing

Click Edit Deployment Plan to set more advanced deployment options.

Save Deployment Plan

After you make changes, you can save the deployment plan to your local disk. You can then use the saved deployment plan to redeploy this application later.

Step 3 of 3

11. Click **Deploy** to accept the values and install the adapter. A deployment confirmation page is displayed.

12. Click **Return** to continue. The **Applications** tab is displayed with the deployed adapter.

Step 2 Define the data source

A data source provides the connection properties to the Banner database. By default, the adapter needs a data source with lookup name `jdbc/bannerws`.

There are two ways to define a data source:

- **At the OC4J instance level** - This method promotes resource sharing, allowing multiple applications in the instance to use the same connection pool to connect to the database.

- **At the application level** - This method permits each application in the instance to access the database via an application-specific connection pool.

If you previously installed a Banner Web Services Adapter in the OC4J instance and defined the data source at the OC4J instance level, then you can skip this step. Otherwise, use the following steps to define the data source.

1. Select the **Administration** tab. A list of tasks is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration		
Expand All Collapse All		
Task Name	Go to Task	Description
▼ Administration Tasks		
▼ Properties		
EJB Compiler Settings		Configure the EJB Compiler.
J2EE Websites		Manage the J2EE websites in this OC4J instance.
JSP Properties		Set JSP container properties.
Logger Configuration		Set log levels for all Loggers.
Thread Pool Configuration		Configure the thread pools of this OC4J instance.
Shared Libraries		Manage the shared libraries of this OC4J instance.
Server Properties		Configure server properties for this OC4J instance.
▼ Services		
JDBC Resources		Create/delete/view data sources and connection pools.
▼ Enterprise Messaging Service		
JMS Destinations		Create/delete/edit JMS destinations.
JMS Connection Factories		Configure JMS connection factories.
In-Memory and File Based Persistence		Configure settings for in-memory and file based persistence.
Database Persistence		Configure settings for database persistence.
OracleAS JMS Router		Configure the JMS Router.
JNDI Browser		Browse the JNDI bindings of this OC4J instance.
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.
▼ Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.
▼ JMX		
System MBean Browser		Browse the system MBeans exposed by this OC4J instance.
Notification Subscriptions		View/change subscriptions for notifications for all MBeans.
Notifications Received		View received notifications.

2. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.

JDBC Resources

Application

Data Sources

Attributes						
Name	Application	JNDI Location	Connection Pool	Managed by OC4J	Test Connection	Delete
"transsvc"	default	jdbc/transsvc	"Transsvc"	✓		

Connection Pools

Name	Application	Connection Factory Class	Monitor Performance	Test Connection	Refresh Connection Pool	Delete
"Transsvc"	default	oracle.jdbc.pool.OracleDataSource				

- Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.

Create Connection Pool - Application

Application

Select the application to which this new connection pool is to be added.

Application

Connection Pool Type

New Connection Pool

New Connection Pool from Existing Connection Pool

Create a new connection pool that is configured like an existing connection pool.

Existing Connection Pool

- Select the application and connection pool type for the new pool:

Application

If you want to define the connection pool at the OC4J level, select *default*. All applications in the instance will use this connection pool.

If you want to define the connection pool at the application level, select the adapter application.

New Connection Pool

Select the button.

- Click **Continue**. The Create Connection Pool page is displayed.

Create Connection Pool

Home | **Attributes** | **Proxy Interfaces**

* Name

* Connection Factory Class
Class must be available to the application's class loader.

URL

You can either specify a URL directly or have it generated from connection information. When you test a connection, the connection factory class and credentials specified on this page will be used to perform the test.

JDBC URL

Generate URL from Connection Information

Driver Type

DB Host Name

DB Listener Port

DB Identifier Type

SID/Service Name

TNS Alias

Credentials

TIP For OracleDataSources, credentials must be entered if not already specified in the URL.

Username

Use Cleartext Password
Password

Use Indirect Password
example: Scott, customers/Scott

6. Enter the following information to set up the connection pool for the `integmgr` schema:

Name	<i>bannerWS_pool</i> (This is an example. Enter the name of your choice.)
Connection Factory Class	<i>oracle.jdbc.pool.OracleDataSource</i>
JDBC URL	<i>jdbc:oracle:thin:@host:port:SID</i> where <i>host</i> = database host <i>port</i> = database listener port (usually 1521) <i>SID</i> = database instance
Username	<i>integmgr</i>
Use Cleartext Password	Select Use Cleartext Password and enter a password for the <code>integmgr</code> schema.

7. Click **Test Connection**. The Test Connection page is displayed.

Test Connection

Enter a SQL statement to use to test the connection. Cancel Test

* SQL Statement

Cancel Test

8. Click **Test** to test the connection pool for the `integmgr` schema. The Create Connection Pool page is redisplayed with a success or failure message.
 - 8.1. If the test succeeds, continue with the next step.
 - 8.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
9. Click **Finish**.
10. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

Create Data Source - Application & Type

Cancel Continue

Application
Select the application to which this new data source is to be added.

Application

Data Source Type

Managed Data Source
A managed data source is one where OC4J provides critical system infrastructure such as global transaction management, connection pooling, statement caching and error handling.

Native Data Source
A native data source is one that implements the `java.sql.DataSource` interface and does not make use of OC4J's connection pooling or statement caching capabilities. A native data source can only participate in local transactions.

New Data Source from Existing Data Source
Create a new data source that is configured like an existing data source.

Existing Data Source

Cancel Continue

11. Select the application and data source type for the data source:

Application

If you want to define the data source at the OC4J level, select *default*. All applications in the instance will use this data source.

If you want to define the data source at the application level, select the adapter application.

Managed Data Source

Select the button.

12. Click **Continue**. The Create Data Source - Managed Data Source page is displayed.

Create Data Source - Managed Data Source

Application **default**

Cancel Back Finish

* Name

* JNDI Location

Transaction Level Global & Local Transactions

Connection Pool bansecr

* Login Timeout (seconds) 0

Maximum time to wait while attempting to connect to a database.

13. Enter the following information to set up the bannerWS data source:

Name	<i>bannerWS</i>
JNDI Location	<i>jdbc/bannerws</i>
Connection Pool	<i>bannerWS_pool</i>

14. Click **Finish**.

Step 3 Configure the security role and user

Before beginning this step, refer to the security configuration for your version of the Oracle Application Server.

Use the following steps to add the `bannerws` role and an administrative user to the adapter. application. This role and user protect the defined endpoint.

1. Select the **Administration** tab. A list of tasks is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration		
Expand All Collapse All		
Task Name	Go to Task	Description
▼ Administration Tasks		
▼ Properties		
EJB Compiler Settings		Configure the EJB Compiler.
J2EE Websites		Manage the J2EE websites in this OC4J instance.
JSP Properties		Set JSP container properties.
Logger Configuration		Set log levels for all Loggers.
Thread Pool Configuration		Configure the thread pools of this OC4J instance.
Shared Libraries		Manage the shared libraries of this OC4J instance.
Server Properties		Configure server properties for this OC4J instance.
▼ Services		
JDBC Resources		Create/delete/view data sources and connection pools.
▼ Enterprise Messaging Service		
JMS Destinations		Create/delete/edit JMS destinations.
JMS Connection Factories		Configure JMS connection factories.
In-Memory and File Based Persistence		Configure settings for in-memory and file based persistence.
Database Persistence		Configure settings for database persistence.
OracleAS JMS Router		Configure the JMS Router.
JNDI Browser		Browse the JNDI bindings of this OC4J instance.
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.
▼ Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.
▼ JMX		
System MBean Browser		Browse the system MBeans exposed by this OC4J instance.
Notification Subscriptions		View/change subscriptions for notifications for all MBeans.
Notifications Received		View received notifications.

2. Select **Security Providers** in the Security section. The Security Providers page is displayed.

Security Providers

Instance Level Security

You can configure the security attributes (realms, users & roles) for all applications deployed to this OC4J instance by clicking on the button below.

[Instance Level Security](#)

Application Server Control Security



You can configure the security provider, users & roles for the Application Server Control management application by clicking on the button below or by using the global Setup link.

[Application Server Control Security](#)

Application Level Security

The table lists applications currently deployed to this OC4J instance and the security provider in use by each application. You can edit the properties of the security provider specified for a given application by clicking on the Edit icon.

[Expand All](#) | [Collapse All](#)

Application Name	Security Provider	Edit
default		
PCT	File-Based Security Provider	
BSR	File-Based Security Provider	
TranslationService	File-Based Security Provider	
AdvancementServices	File-Based Security Provider	
javasso	File-Based Security Provider	
Campuscard	File-Based Security Provider	
Housing	File-Based Security Provider	
datatags	File-Based Security Provider	
CardholderEventPublisher	File-Based Security Provider	

3. In the Application Level Security section, click the **Edit** button for the adapter application. The Security Provider page is displayed.
4. Select the **Realms** tab.

Security Provider


Security Provider Type **File-Based Security Provider** [Change Security Provider](#)

Security Provider Attributes: File-Based Security Provider

[General](#) [Realms](#)

Search
Name [Go](#)

Results
[Create](#)

Realm Name ▲	Roles	Users	Delete
jazn.com	9	6	

5. Click the link under the **Roles** column. The Roles page is displayed.

Roles

Security Provider Type **File-Based Security Provider**
Realm Name **jazn.com**

Search

Name

Results

Role Name [△]	Users	Delete
ascontrol_admin	1	
ascontrol_appadmin	0	
ascontrol_monitor	1	

- Click **Create**. The Add Role page is displayed.

Add Role

Realm Name **jazn.com**

* Name

Grant RMI Login Permission

Grant Administration Permission

Assign Roles

A role may inherit from other roles. Select the roles you would like this role to inherit.

Available Roles		Selected Roles
ascontrol_admin ascontrol_appadmin ascontrol_monitor	 Move Move All Remove Remove All	

- Enter *bannerws* in the **Name** field.
- Click **OK**. The Roles page is redisplayed with the new role.
- Return to the Security Provider page.

Security Provider

Security Provider Type **File-Based Security Provider**

Security Provider Attributes: File-Based Security Provider

General	Realms

Search

Name

Results

Realm Name [△]	Roles	Users	Delete
jazn.com	9	6	

10. Click the link under the **Users** column. The Users page is displayed.

Users

Security Provider Type **File-Based Security Provider**
Realm Name **jzn.com**

Search

Name

Results

User Name	Assigned Roles	Delete
anonymous		
JtaAdmin	oc4j-administrators*	
oc4jadmin	oc4j-administrators*, ascontrol_admin*	
rmiuser	ascontrol_monitor*	

11. Click **Create**. The Add User page is displayed.

Add User

Realm Name **jzn.com**

* Name

* Password

* Confirm Password

Assign Roles

Available Roles

- ascontrol_admin
- ascontrol_appadmin
- ascontrol_monitor
- bannerws

Selected Roles

12. Enter the following information to create a user:

Name *Admin*
(This is an example. Enter the name of your choice.)

Password Password for the user being created

Confirm Password Confirmation of the password

13. In the Assign Roles section, select the *bannerws* role in the **Available Roles** list and move it to the **Selected Roles** list.























14. Click **OK**. The Users page is redisplayed with the new user.

Step 4 Enable schema validation (optional)

Validating XML request and response messages for each Web service invocation degrades system performance. For this reason, schema validation is turned off by default. To enable schema validation, you must set system property `BANNERWS_SCHEMA_VALIDATION` with a value of `true` for the OC4J instance where the adapter is installed. Use the following steps to enable schema validation.

1. Select the **Administration** tab. A list of tasks is displayed.

OC4J: BWS

Home	Applications	Web Services	Performance	Administration
Expand All Collapse All				
Task Name	Go to Task	Description		
▼ Administration Tasks				
▼ Properties				
EJB Compiler Settings		Configure the EJB Compiler.		
J2EE Websites		Manage the J2EE websites in this OC4J instance.		
JSP Properties		Set JSP container properties.		
Logger Configuration		Set log levels for all Loggers.		
Thread Pool Configuration		Configure the thread pools of this OC4J instance.		
Shared Libraries		Manage the shared libraries of this OC4J instance.		
Server Properties		Configure server properties for this OC4J instance.		
▼ Services				
JDBC Resources		Create/delete/view data sources and connection pools.		
▼ Enterprise Messaging Service				
JMS Destinations		Create/delete/edit JMS destinations.		
JMS Connection Factories		Configure JMS connection factories.		
In-Memory and File Based Persistence		Configure settings for in-memory and file based persistence.		
Database Persistence		Configure settings for database persistence.		
OracleAS JMS Router		Configure the JMS Router.		
JNDI Browser		Browse the JNDI bindings of this OC4J instance.		
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.		
▼ Security				
Security Providers		Configure security providers, create/delete/view users and roles.		
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.		
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.		
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.		
▼ JMX				
System MBean Browser		Browse the system MBeans exposed by this OC4J instance.		
Notification Subscriptions		View/change subscriptions for notifications for all MBeans.		
Notifications Received		View received notifications.		

2. Select **Server Properties** in the Properties section. The Server Properties page is displayed.

Server Properties

Revert Apply

Multiple VM Configuration

Number of VM Processes

Ports

 **TIP** Be sure that the port ranges specified below are large enough to accommodate the total number of processes

RMI Port RMIS Port

JMS Port

Web Sites

Name	Port	Protocol
default-web-site	<input type="text" value="12501-12600"/>	<input type="text" value="ajp"/>

Command Line Options

Start-parameters: Java Options

Server VM

Enable J2SE 5.0 Platform MBeans




Only applicable to
Java HotSpot VM

Verbose

Verbose:gc

Maximum heap size

Initial heap size

Options	Delete
<input type="text" value="-Djava.security.policy=\$ORACLE_HOME/j2ee/BWS_16_Certification/config/java2.policy"/>	
<input type="text" value="-Djava.awt.headless=true"/>	
<input type="text" value="-Dhttp.webdir.enable=false"/>	
<input type="button" value="Add Another Row"/>	

- In the Command Line Options section, click **Add Another Row**.
- Add the following system property:

```
-DBANNERWS_SCHEMA_VALIDATION=true
```

- Click **Apply**. A confirmation message is displayed.
- Click **Yes** to restart the server.

Step 5 Configure logging

The Banner Web Services Adapters use Apache's log4j to log the activities performed by the application at runtime. Log4j uses a properties file to establish specific runtime options. The following options should be reviewed and modified as appropriate:

- Location of the log files.** The default location is `<IAS_HOME>/j2ee/home/log`. This location should be changed to the OC4J instance where the Banner Web Services Adapter is installed.
- Logging level.** The default level is `DEBUG`, resulting in large amount of information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide more limited logging after initial operations, you should modify the logging level to `INFO`.

Use the following steps to modify the logging options as appropriate.

Banner Web Services Adapter for Campus Card Systems

1. Navigate to `<IAS_HOME>/j2ee/<OC4J instance>/applications/CampusCardIntegration_v8.1.3/CampusCardIntegration_web/WEB-INF/classes`.
2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
<code>log4j.appender.out.File</code>	<code>log/campuscard.log</code>	<code>../<OC4J instance>/log/campuscard.log</code>
<code>log4j.category.com.sungardsct</code>	DEBUG	INFO

3. Restart the OC4J instance for the changes to take effect.

Banner Web Services Adapter for Housing Systems

1. Navigate to `<IAS_HOME>/j2ee/<OC4J instance>/applications/HousingIntegration_v8.1.3/HousingIntegration_web/WEB-INF/classes`.
2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
<code>log4j.appender.out.File</code>	<code>log/housing.log</code>	<code>../<OC4J instance>/log/housing.log</code>
<code>log4j.category.com.sungardsct</code>	INFO	DEBUG

3. Restart the OC4J instance for the changes to take effect.

Step 6 Verify the deployment

Use the following steps to verify that the adapter is successfully deployed.

1. Use a Web browser to access the URL mapped in [Step 1, “Install the adapter”](#):

```
<http or https>://<host>:<port>/<context root>/
```

This URL is used to access an information page, not the endpoint. The context root is *campuscard* or *housing*, unless the context root was changed when the adapter was installed.

2. Log in with the name and password configured in [Step 3, “Configure the security role and user”](#). An information page for the adapter is displayed. This page shows the version of the adapter and provides a link to the Web service’s WSDL.
3. (Optional) If you need to determine the URL that is being used to listen for messages, click the link on the information page to display the associated WSDL. The `<soap:address location>` attribute under the `<service>` tag at the bottom of the WSDL identifies the URL that you should always use to invoke the Web service.

Installation on Oracle WebLogic Server 11g

The adapters are packaged as J2EE compatible enterprise archive files. Each file must be deployed and configured separately:

- `CampusCardIntegration_v8.1.3.ear` exposes Web services that external campus card systems need to integrate with Banner.
- `HousingIntegration_v8.1.3.ear` exposes Web services that external housing systems need to integrate with Banner.

Use the following steps to install each adapter on Oracle WebLogic Server 11g:

- [Step 1, “Configure the Oracle WebLogic Server”](#)
- [Step 2, “Configure logging \(optional\)”](#)
- [Step 3, “Define the data source”](#)
- [Step 4, “Install the adapter”](#)
- [Step 5, “Configure the security group and user”](#)
- [Step 6, “Enable schema validation \(optional\)”](#)

Banner Web Service Adapters can be installed on an existing Oracle WebLogic Domain. However, not on a Classic Domain that comes with `WLS_FORMS` and `WLS_REPORTS` servers. Banner Web Services Adapters must be installed on the same Managed Server as the Banner Translation Service. The Managed Server to which these applications are deployed should be dedicated to the Banner Web Services components so they can be managed independently from other applications.

Step 1 Configure the Oracle WebLogic Server

The Oracle WebLogic Server must be configured to use the *Advanced* security model instead of the default *DD only* option. Use the following steps to configure the server.

Note

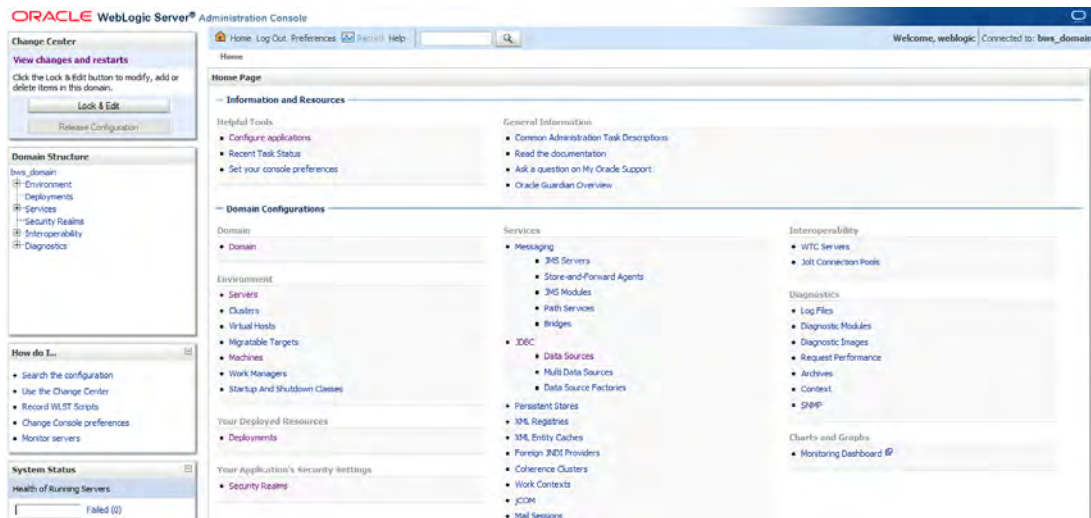
The Oracle WebLogic Server needs to be configured only once. If the server was previously configured, you can skip this step. ■

Use the following steps to configure the server.

1. Connect to the Oracle WebLogic Server Administration Console:

`http://<host>:<port>/console`

The Home Page is displayed.



2. Click **Lock & Edit** from the Change Center.

3. In the Domain Structure pane, click **Security Realms**.



The Summary of Security Realms page is displayed.

Summary of Security Realms

A security realm is a container for the mechanisms—including users, groups, security roles, security policies, and security providers—that are used to protect WebLogic resources. You can have multiple security realms in a WebLogic Server domain, but only one can be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebLogic Server domain. Click the name of the realm to explore and configure that realm.

Customize this table

Realms(Filtered - More Columns Exist)

<input type="checkbox"/>	Name ↕	Default Realm
<input type="checkbox"/>	myrealm	true

- Click **myrealm**. The Settings for myrealm page is displayed.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

General RDBMS Security Store User Lockout Performance

Save

Use this page to configure the general behavior of this security realm.

Note:
If you are implementing security using JACC (Java Authorization Contract for Containers as defined in JSR 115), you must use the DD Only security model. Other WebLogic Server models are not available and the security functions for Web applications and EJBs in the Administration Console are disabled.

Name: myrealm The name of this security realm. [More Info...](#)

Security Model Default: Advanced Specifies the default security model for Web applications or EJBs that are secured by this security realm. You can override this default during deployment. [More Info...](#)

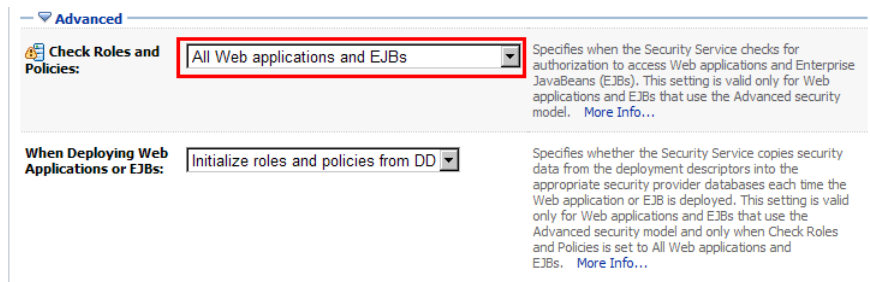
Combined Role Mapping Enabled Determines how the role mappings in the Enterprise Application, Web application, and EJB containers interact. This setting is valid only for Web applications and EJBs that use the Advanced security model and that initialize roles from deployment descriptors. [More Info...](#)

Use Authorization Providers to Protect JMX Access Configures the WebLogic Server MBean servers to use the security realm's Authorization providers to determine whether a JMX client has permission to access an MBean attribute or invoke an MBean operation. [More Info...](#)

▶ Advanced

Save

- Select *Advanced* in the **Security Model Default** drop-down list.
- Click the **Advanced** link to display the advanced options.



7. Select *All Web Applications and EJBs* in the **Check Roles and Policies** drop-down list.
8. Click **Save**.
9. Restart the server for the changes to take effect.

Step 2 Configure logging (optional)

The Banner Web Services Adapters use Apache's log4j to log the activities performed by the application at runtime. The log file is located at the following location:

```
Oracle\Middleware\user_projects\domains\\log
```

where `<domain_name>` is the name of the domain where the Banner Web Services Adapters will be installed. This location cannot be changed.

A property in the `log4j.properties` file determines the logging level. The default logging level is *DEBUG*, resulting in large amount of information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide more limited logging after initial operations, you should modify the logging level to *INFO*.

Banner Web Services Adapter for Campus Card Systems

1. Copy `CampusCardIntegration_v8.1.3.ear` to a temporary location. This location is referred to as `<EAR_HOME>`.

2. Navigate to `<EAR_HOME>` and execute the following command.

```
jar xvf CampusCardIntegration_v8.1.3.ear
```

The extract contains a Web archive named `CampusCardIntegration_web.war`.

3. Create a folder under `<EAR_HOME>` and name it `war_home`.

4. Navigate to `war_home` and execute the following command.

```
jar xvf <EAR_HOME>/CampusCardIntegration_web.war
```

5. Open `war_home\WEB-INF\classes\log4j.properties`

6. Edit the `log4j.category.com.sungardsct` property as follows:

Original value: *DEBUG*
New value: *INFO*

7. Save the change.
8. From `war_home` execute the following command to rebuild the Web archive file.

```
jar cvf <EAR_HOME>/CampusCardIntegration_web.war META-INF/*  
WEB-INF/* ui/* index.jsp
```

9. From `<EAR_HOME>` execute the following command to rebuild the enterprise archive file.

```
jar cvf CampusCardIntegration_v8.1.3.ear *.war META-INF/*  
legal/* APP-INF/*
```

The rebuilt `CampusCardIntegration_v8.1.3.ear` is used for installation.

Banner Web Services Adapter for Housing Systems

1. Copy `HousingIntegration_v8.1.3.ear` to a temporary location. This location is referred to as `<EAR_HOME>`.

2. Navigate to `<EAR_HOME>` and execute the following command.

```
jar xvf HousingIntegration_v8.1.3.ear
```

The extract contains a Web archive named `HousingIntegration_web.war`.

3. Create a folder under `<EAR_HOME>` and name it `war_home`.

4. Navigate to `war_home` and execute the following command.

```
jar xvf <EAR_HOME>/HousingIntegration_web.war
```

5. Open `war_home\WEB-INF\classes\log4j.properties`.

6. Edit the `log4j.category.com.sungardsct` property as follows:

Original value: *DEBUG*
New value: *INFO*

7. Save the change.
8. From `war_home` execute the following command to rebuild the Web archive file.

```
jar cvf <EAR_HOME>/HousingIntegration_web.war META-INF/* WEB-  
INF/* ui/* index.jsp
```

- From <EAR_HOME> execute the following command to rebuild the enterprise archive file.

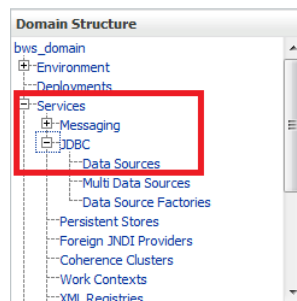
```
jar cvf HousingIntegration_v8.1.3.ear *.war META-INF/* legal/* APP-INF/*
```

The rebuilt `HousingIntegration_v8.1.3.ear` is used for installation.

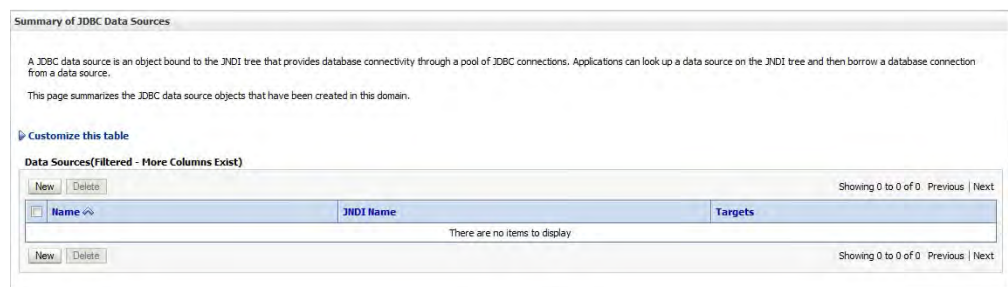
Step 3 Define the data source

A data source provides the connection properties to the Banner database. By default, the adapter needs a data source with lookup name `jdbc/bannerws`. If you previously installed a Banner Web Services Adapter in the instance, then you can skip this step. Otherwise, use the following steps to define the data source.

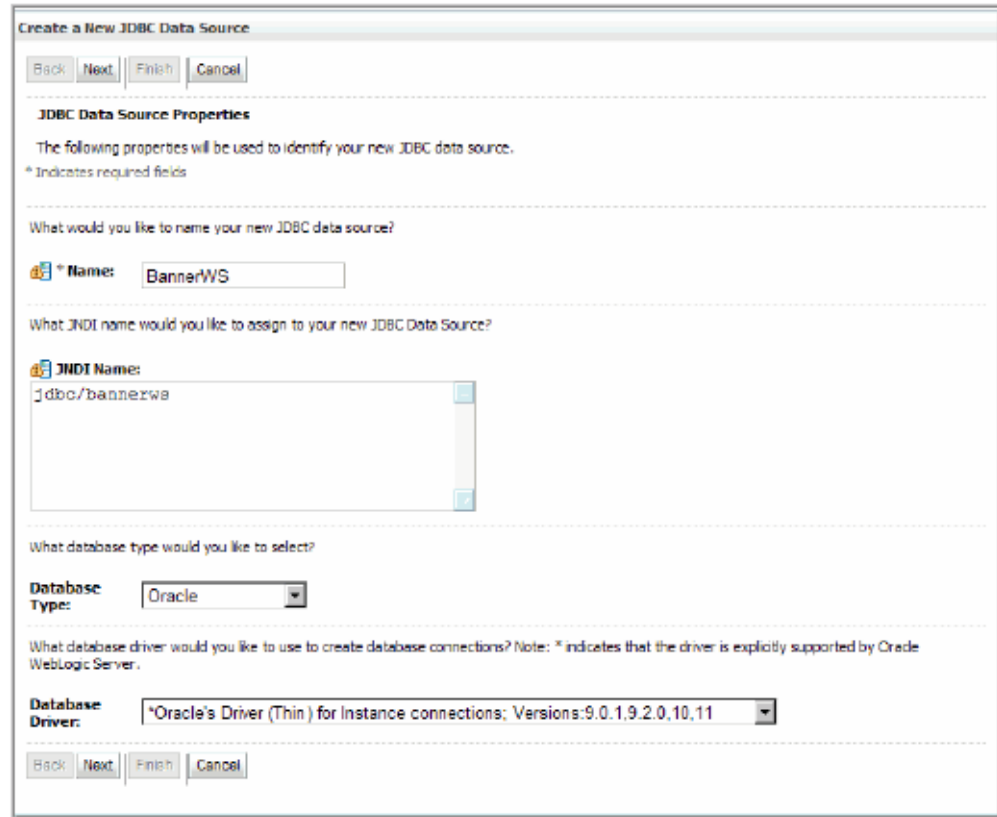
- In the Change Center pane, click **Lock & Edit**.
- In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**.



The Summary of JDBC Data Sources page is displayed.



- Click **New**. The Create a New JDBC Data Source page is displayed.



4. Enter the following data source properties:

Name	<i>BannerWS</i>
JNDI Name	<i>jdbc/bannerws</i>
Database Type	<i>Oracle</i>
Database Driver	Appropriate database driver that is used to create database connections: <ul style="list-style-type: none"> • If your database is RAC-based, select <i>Oracle's Driver (Thin) for RAC Service -Instance connections; Versions:10,11</i> • Otherwise, select <i>Oracle's Driver (Thin) for Instance connections; Versions:9.0.1, 9.2.0,10,11</i>

5. Click **Next**. The next page is displayed.

The following page may or may not be displayed. If it is displayed, clear the **Supports Global Transactions** check box and go to step 6. If the following page is not displayed, skip to step 7.

Create a New JDBC Data Source

Back Next Finish Cancel

Transaction Options

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

Supports Global Transactions

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the *Logging Last Resource* (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

Logging Last Resource

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.

Emulate Two-Phase Commit

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.

One-Phase Commit

Back Next Finish Cancel

6. Clear the **Supports Global Transactions** check box.
7. Click **Next**. The next page is displayed.

Create a New JDBC Data Source

Back Next Finish Cancel

Connection Properties

Define Connection Properties.

What is the name of the database you would like to connect to?

Database Name:

What is the name or IP address of the database server?

Host Name:

What is the port on the database server used to connect to the database?

Port:

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

Database User Name:

What is the database account password to use to create database connections?

Password:

Confirm Password:

Back Next Finish Cancel

8. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	<i>integmgr</i>
Password	Password for the <i>integmgr</i> user
Confirm Password	Confirmation of the password

- Click **Next**. The next page is displayed with the properties that you entered.

The screenshot shows the 'Create a New JDBC Data Source' wizard at the 'Test Database Connection' step. The window title is 'Create a New JDBC Data Source'. At the top, there are navigation buttons: 'Test Configuration', 'Back', 'Next', 'Finish', and 'Cancel'. The main content area is titled 'Test Database Connection' and contains the following text and fields:

- Text: 'Test the database availability and the connection properties you provided.'
- Text: 'What is the full package name of JDBC driver class used to create database connections in the connection pool?' (Note that this driver class must be in the classpath of any server to which it is deployed.)
- Field: 'Driver Class Name:' with the value 'oracle.jdbc.OracleDriver'.
- Text: 'What is the URL of the database to connect to? The format of the URL varies by JDBC driver.'
- Field: 'URL:' with the value 'jdbc:oracle:thin:@m08804'.
- Text: 'What database account user name do you want to use to create database connections?'
- Field: 'Database User Name:' with the value 'integmgr'.
- Text: 'What is the database account password to use to create database connections?' (Note: for secure password management, enter the password in the Password field instead of the Properties field below)
- Field: 'Password:' with a masked password (dots).
- Field: 'Confirm Password:' with a masked password (dots).
- Text: 'What are the properties to pass to the JDBC driver when creating database connections?'
- Field: 'Properties:' with the value 'user=integmgr'.
- Text: 'What table name or SQL statement would you like to use to test database connections?'
- Field: 'Test Table Name:' with the value 'SQL SELECT 1 FROM DUAL'.

- Verify the property values.

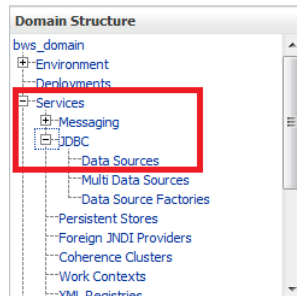
11. Click **Test Configuration**. The page is redisplayed with a success or failure message.
 - 11.1. If the test succeeds, continue with the next step.
 - 11.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
12. Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.

Name	JNDI Name	Targets
banners	jdbc:banners	

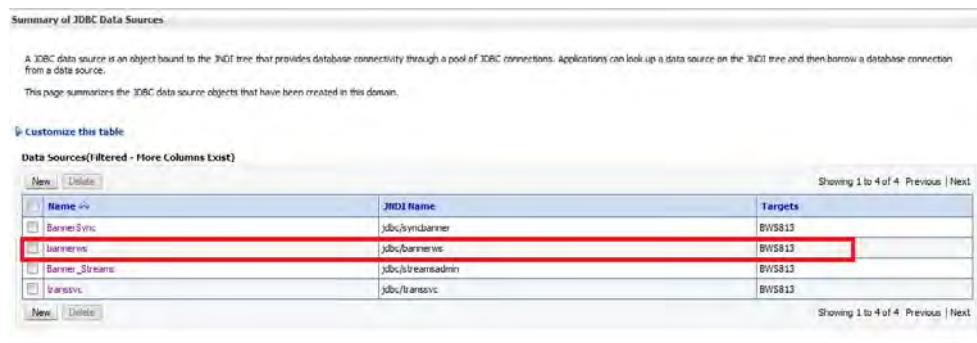
13. In the Change Center pane, click **Activate Changes**.
14. On the Summary of JDBC Data Sources page, click the name of the new data source. The Settings for BannerWS page is displayed.
15. Select the **Targets** tab.

Servers
<input type="checkbox"/> AdminServer
<input type="checkbox"/> BEISManagedserver
<input type="checkbox"/> BWS813

16. In the Change Center pane, click **Lock & Edit**.
17. On the Settings for BannerWS page, select the server where the data source should be deployed.
18. Click **Save**.
19. In the Change Center pane, click **Activate Changes**.
20. In the Domain Structure page, expand and click **Services -> JDBC -> Data Sources**.



The Summary of JDBC Data Sources page is displayed.



Verify that the new data source is associated with the server.

Step 4 Install the adapter

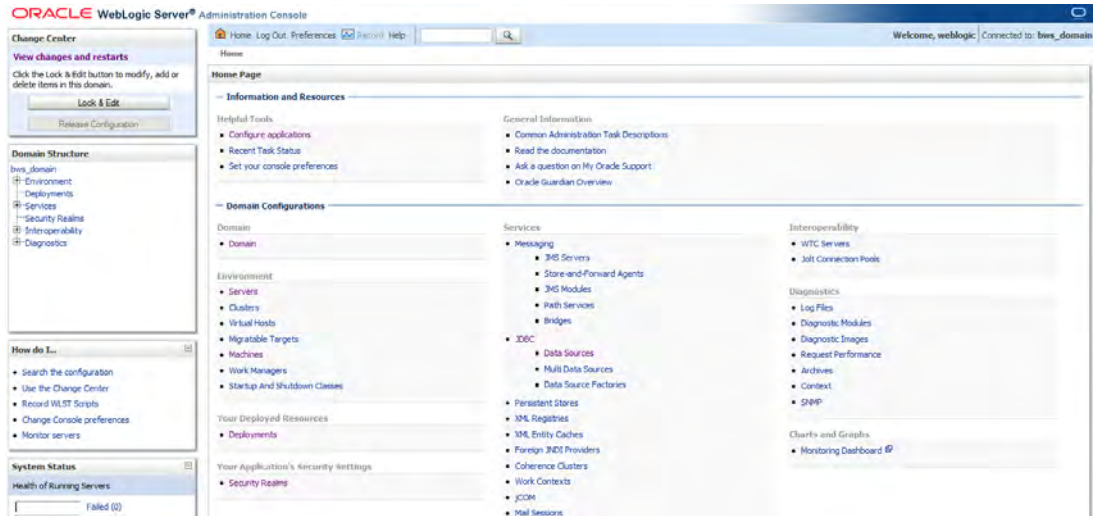
Before beginning this step, you must understand the concepts published by Oracle regarding the deployment of ear files.

Use the following steps to install the adapter to the Oracle WebLogic Server.

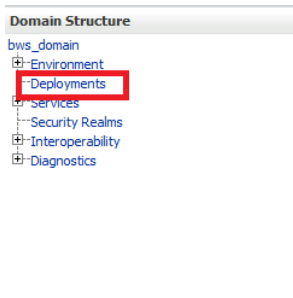
1. Connect to the Oracle WebLogic Server Administration Console:

`http://<host>:<port>/console`

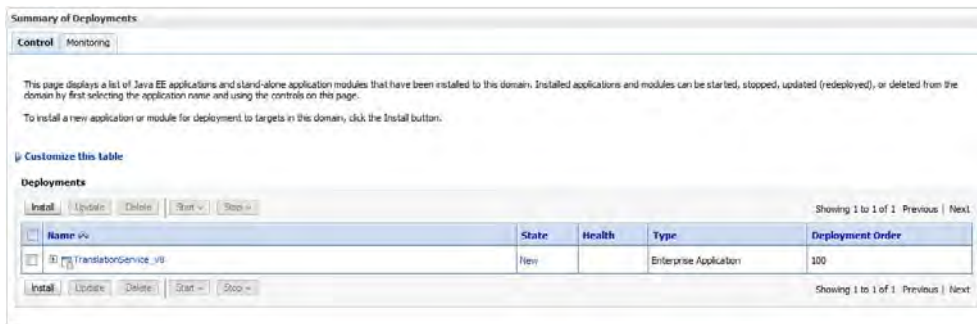
The Home Page is displayed.



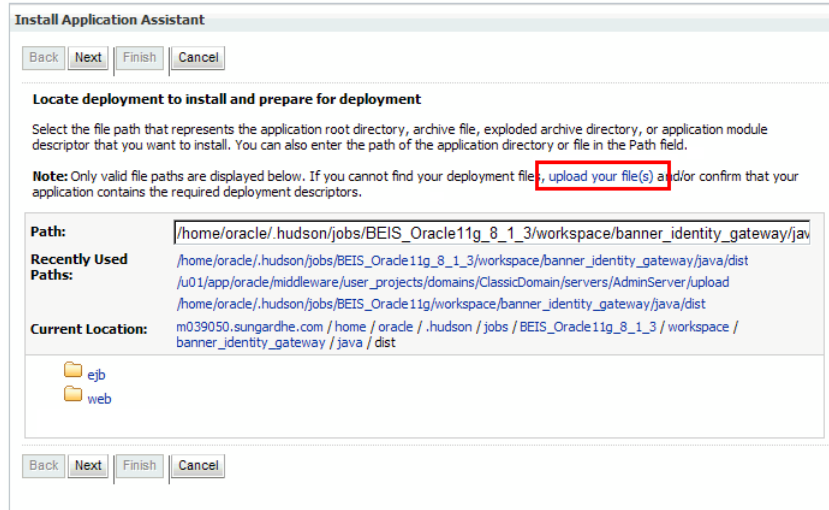
2. In the Change Center pane, click **Lock & Edit**.
3. In the Domain Structure pane, click **Deployments**.



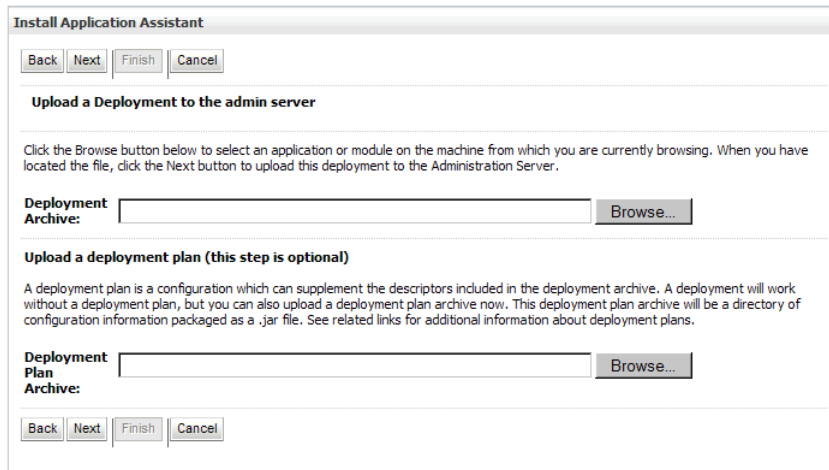
The Summary of Deployments page is displayed.



4. In the Summary of Deployments page, click **Install**. The Install Application Assistant page is displayed.



5. Click **upload your file(s)**. The next installation page is displayed.

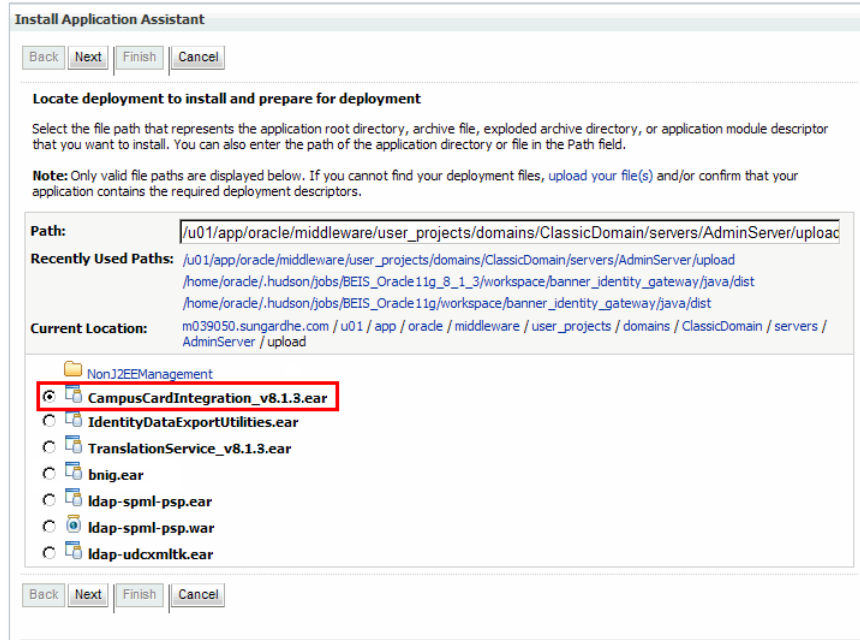


6. Select the file to be uploaded:
 - 6.1. In the **Deployment Archive** field, click **Browse** and navigate to the appropriate ear file:

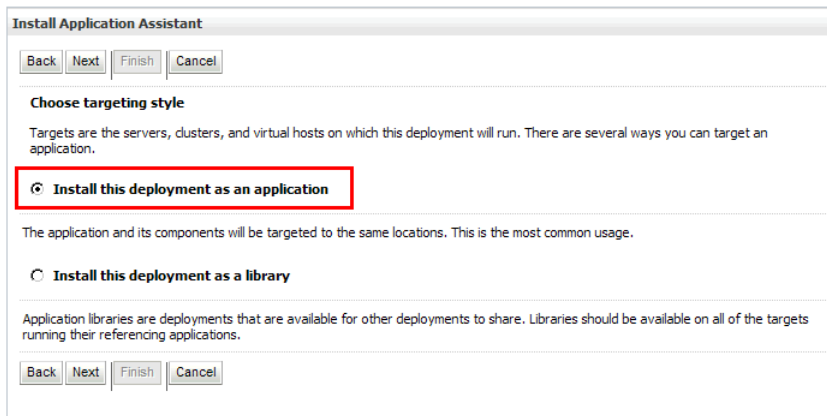

```
CampusCardIntegration_v8.1.3.ear
```

or

```
HousingIntegration_v8.1.3.ear
```
 - 6.1. Select the file and click **Open**.
7. Click **Next**. The next installation page is displayed.



8. Select the adapter ear file from the list.
9. Click **Next**. The next installation page is displayed.



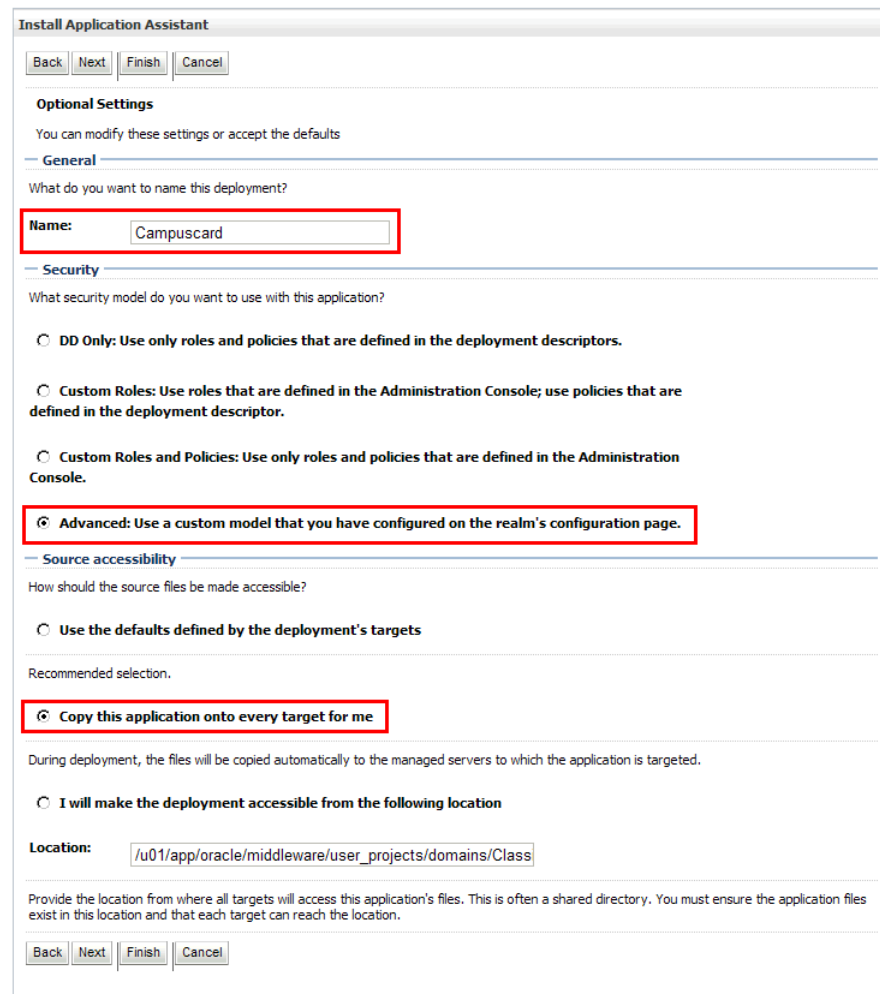
10. Select **Install this deployment as an application**.
11. Click **Next**. The next installation page is displayed.
 - 11.1. In some domains, the following page is displayed. Select the server where the adapter should be deployed and go to step 13.

The adapter must be installed in the instance where the Banner Translation Service is installed.



11.2. In some domains, the preceding page is skipped. Go directly to step 13.

12. Click **Next**. The next installation page is displayed.



13. Enter a name for the application (for example, *Campuscard*) in the **Name** field.

14. Select **Advanced: Use a custom model that you have configured on the realm's configuration page**.

15. Select **Copy this application onto every target for me**.

16. Click **Next**. The next installation page is displayed.

Install Application Assistant

Back Next Finish Cancel

Review your choices and click Finish

Click Finish to complete the deployment. This may take a few moments to complete.

Additional configuration

In order to work successfully, this application may require additional configuration. Do you want to review this application's configuration after completing this assistant?

Yes, take me to the deployment's configuration screen.

No, I will review the configuration later.

Summary

Deployment: /u01/app/oracle/middleware/user_projects/domains/ClassicDomain/servers/AdminServer/upload/CampusCardIntegration_v8.1.2.ear

Name: Campuscard

Staging mode: Copy this application to every target for me

Security Model: Advanced: Use a custom model that you have configured on the realm's configuration page.

Target Summary

Components	Targets
CampusCardIntegration_v8.1.3.ear	AdminServer

Back Next Finish Cancel

17. Select **No, I will review the configuration later.**

18. Click **Finish** to start the deployment. When deployment is completed, the Summary of Deployments page is redisplayed with the newly deployed adapter.

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Customize this table

Deployments

Install Update Delete Start Stop Previous Next

Name	State	Health	Type	Deployment Order
adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
bniq	New		Enterprise Application	100
Campuscard	distribute Initializing		Enterprise Application	100
DMS Application (11.1.1.1.0)	Active	OK	Web Application	5

19. In the Change Center pane, click **Activate Changes**.

20. Start the newly deployed application as follows:

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install | Update | Delete | **Start** | Stop | Previous | Next

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input type="checkbox"/>	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	bnig	New		Enterprise Application	100
<input checked="" type="checkbox"/>	Campuscard	distribute Initializing		Enterprise Application	100
<input type="checkbox"/>	DMS Application (11.1.1.1.0)	Active	OK	Web Application	5

20.1. Select the newly deployed adapter.

20.1. Click **Start -> Servicing all requests**. The Start Application Assistant page is displayed.

Start Application Assistant

Yes No

Start Deployments

You have selected the following deployments to be started. Click 'Yes' to continue, or 'No' to cancel.

- Campuscard

Yes No

20.2. Click **Yes**.

Step 5 Configure the security group and user

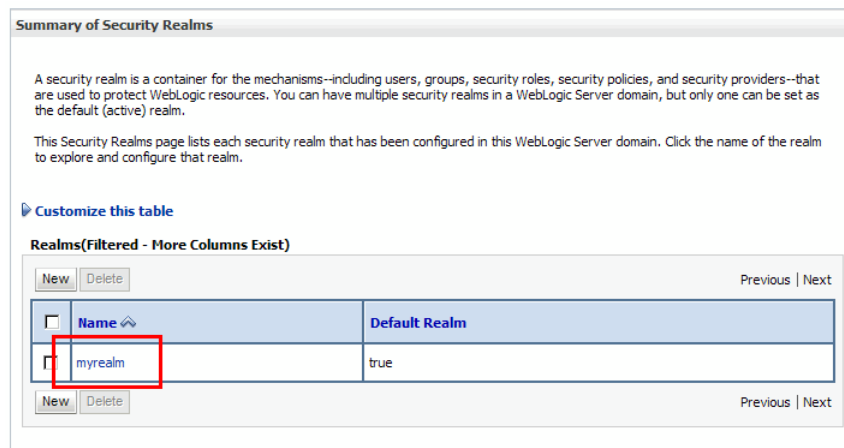
Before beginning this step, refer to the security configuration for your version of the Oracle WebLogic Server.

Use the following steps to add the `bannerwsGroup` group and an administrative user to the adapter. This group and user protect the defined endpoint.

1. In the Domain Structure pane, click **Security Realms**.



The Summary of Security Realms page is displayed.



The screenshot shows the "Summary of Security Realms" page. It includes an introductory paragraph, a "Customize this table" link, and a table titled "Realms(Filtered - More Columns Exist)". The table has two columns: "Name" and "Default Realm". The "myrealm" row is highlighted with a red box.

<input type="checkbox"/>	Name ↕	Default Realm
<input type="checkbox"/>	myrealm	true

2. Click **myrealm**. The Settings for myrealm page is displayed.
3. Select the **Users and Groups** tab.
4. Select the **Groups** sub-tab. A table of existing groups is displayed.

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users **Groups**

This page displays information about each group that has been configured in this security realm.

[Customize this table](#)

Groups

New Delete Previous Next

<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	AdminChannelUsers	AdminChannelUsers can access the admin channel.	DefaultAuthenticator
<input type="checkbox"/>	Administrators	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
<input type="checkbox"/>	AppTesters	AppTesters group.	DefaultAuthenticator
<input type="checkbox"/>	CrossDomainConnectors	CrossDomainConnectors can make inter-domain calls from foreign domains.	DefaultAuthenticator
<input type="checkbox"/>	DemoGroup	Demo group created for demo purpose	DefaultAuthenticator
<input type="checkbox"/>	Deployers	Deployers can view all resource attributes and deploy applications.	DefaultAuthenticator
<input type="checkbox"/>	idpadmin	Enterprise Identity Proxy Services Group	DefaultAuthenticator
<input type="checkbox"/>	Monitors	Monitors can view and modify all resource attributes and perform operations not restricted by roles.	DefaultAuthenticator

New Delete Previous Next

- Click **New**. The Create a New Group page is displayed.

Create a New Group

OK Cancel

Group Properties

The following properties will be used to identify your new Group.

* Indicates required fields

What would you like to name your new Group?

* **Name:**

How would you like to describe the new Group?

Description:

Please choose a provider for the group.

Provider:

OK Cancel

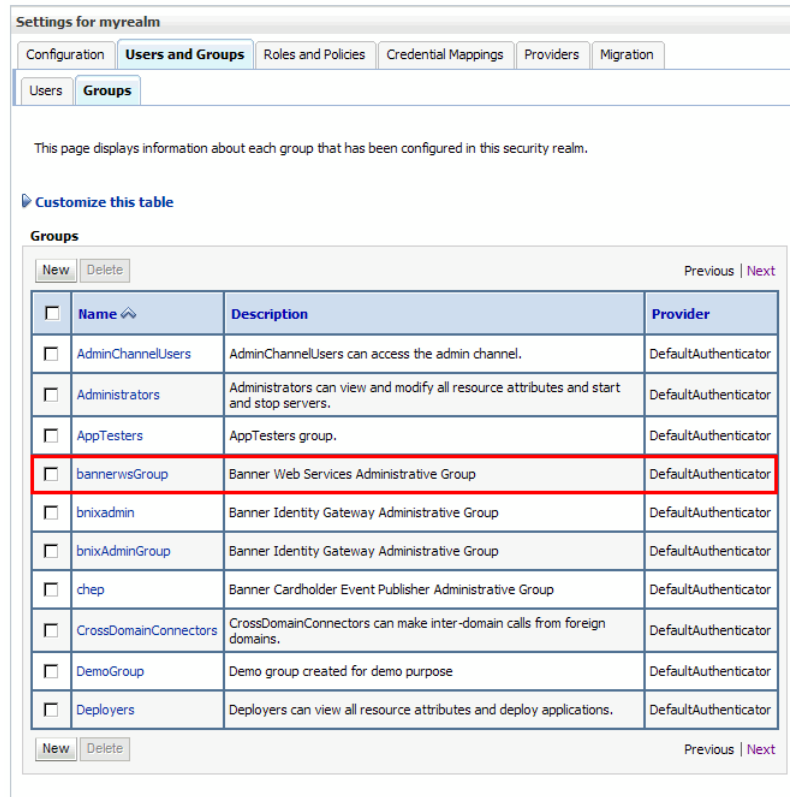
- Enter the following information to create a group:

Name *bannerwsGroup*

Description *Banner Web Services Administrative Group*

Provider *DefaultAuthenticator*

- Click **OK**. The table of groups is redisplayed with the new group.



- Select the **Users** sub-tab. A table of existing users is displayed.

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users Groups

This page displays information about each user that has been configured in this security realm.

[Customize this table](#)

Users

New Delete Previous | Next

<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	bnix	Banner Identity Gateway Administrator	DefaultAuthenticator
<input type="checkbox"/>	DemoUser	Demo user created for demo purpose	DefaultAuthenticator
<input type="checkbox"/>	idproxy	Enterprise Identity Proxy Services User	DefaultAuthenticator
<input type="checkbox"/>	OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
<input type="checkbox"/>	transsvc		DefaultAuthenticator
<input type="checkbox"/>	weblogic		DefaultAuthenticator

New Delete Previous | Next

9. Click **New**. The Create a New User page is displayed.

Create a New User

OK Cancel

User Properties

The following properties will be used to identify your new User.

* Indicates required fields

What would you like to name your new User?

* **Name:**

How would you like to describe the new User?

Description:

Please choose a provider for the user.

Provider:

The password is associated with the login name for the new User.

Password:

Confirm Password:

OK Cancel

10. Enter the following information to create a user:

Name *admin*
(This is an example. Enter the name of your choice.)

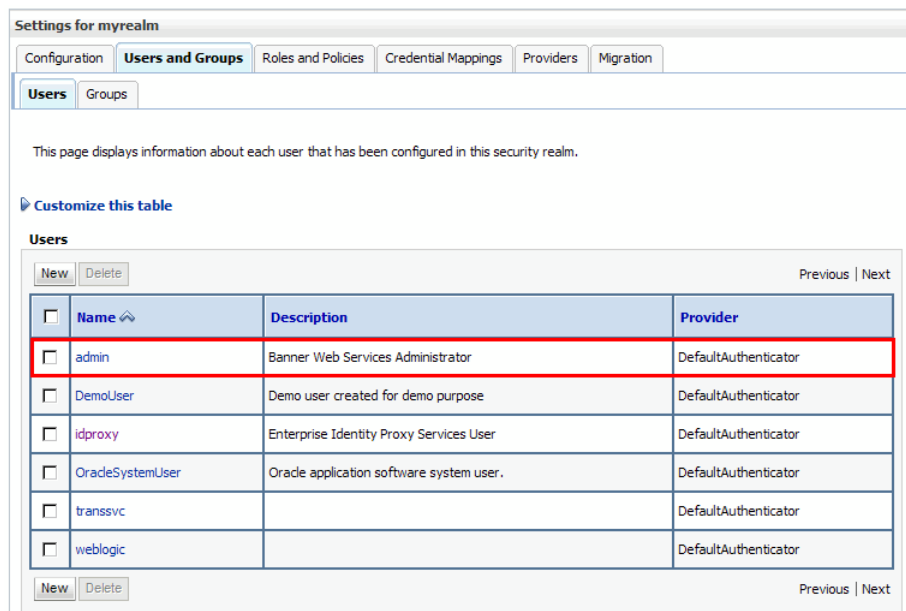
Description *Banner Web Services Administrator*

Provider *DefaultAuthenticator*

Password Password for the user being created

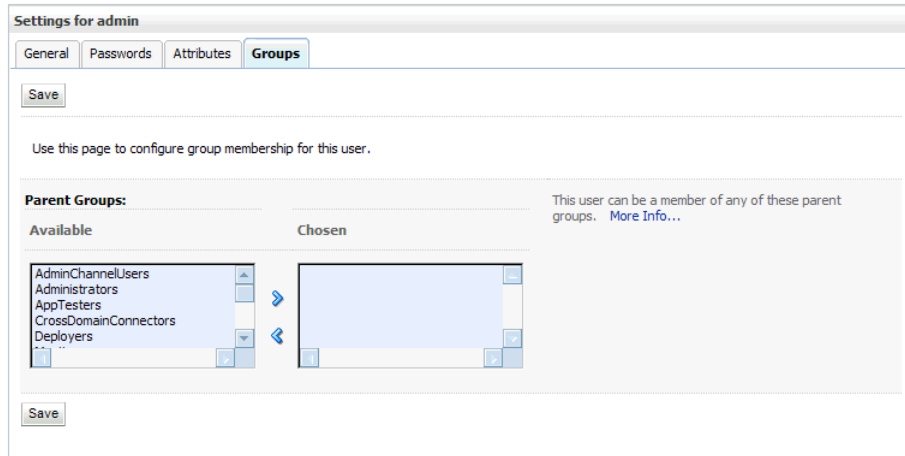
Confirm Password Confirmation of the password

11. Click **OK**. The table of users is redisplayed with the new user.

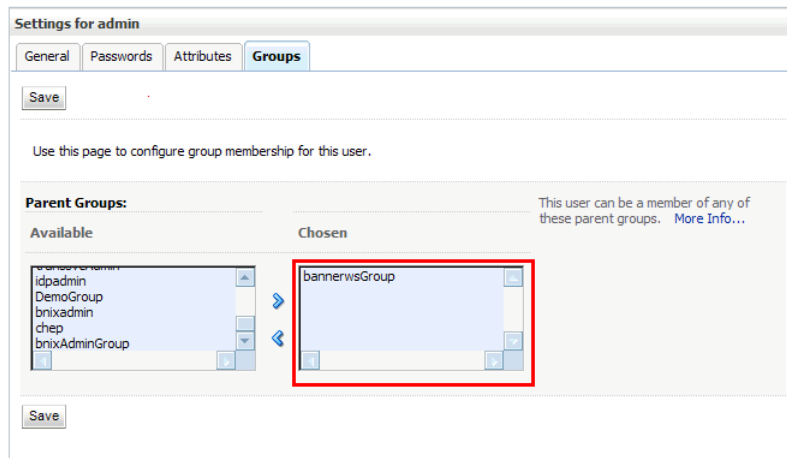


12. Click the name of the user you just created. The Settings page for the user is displayed.

13. Select the **Groups** tab.



14. In the Parent Groups section, select *bannerwsGroup* in the **Available** list and move it to the **Chosen** list.



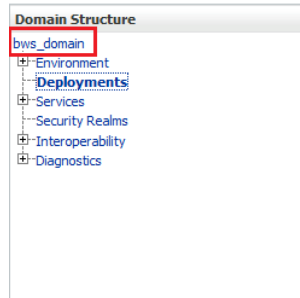
15. Click **Save**.

Step 6 Enable schema validation (optional)

Validating XML request and response messages for each Web service invocation degrades system performance. For this reason, schema validation is turned off by default. To enable schema validation, you must set system property `BANNERWS_SCHEMA_VALIDATION` with

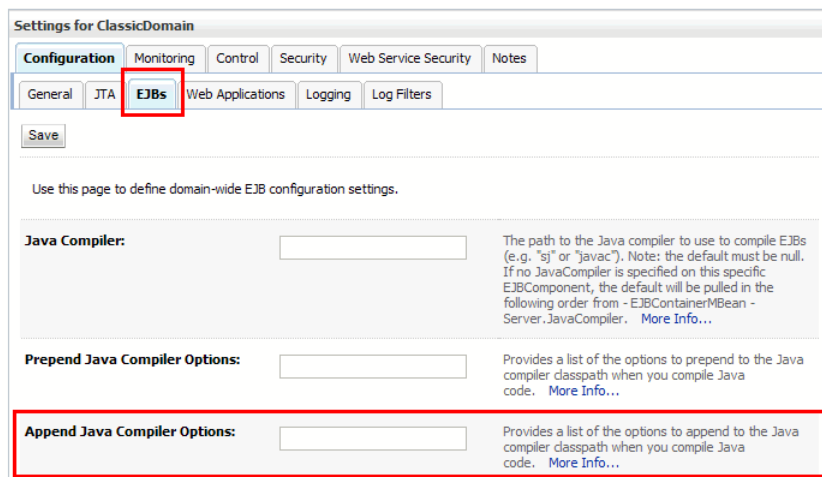
a value of *true* for the instance where the adapter is installed. Use the following steps to enable schema validation.

1. In the Domain Structure pane, click the name of the domain.



The Settings page is displayed.

2. Select the **EJBs** tab.



3. Add the following value in the **Append Java Compiler Options** field:

```
-DBANNERWS_SCHEMA_VALIDATION=true
```

4. Click **Save**.
5. Restart the server for the changes to take effect.

Step 7 Verify the deployment

Use the following steps to verify that the adapter is successfully deployed.

1. Use a Web browser to access the URL mapped in following URL:

`<http or https>://<host>:<port>/<context root>/`

This URL is used to access an information page, not the endpoint. The context root is *campuscard* or *housing*.

2. Log in with the name and password configured in [Step 5, “Configure the security group and user”](#). An information page for the adapter is displayed. This page shows the version of the adapter and provides a link to the Web service’s WSDL.
3. (Optional) If you need to determine the URL that is being used to listen for messages, click the link on the information page to display the associated WSDL. The `<soap:address location>` attribute under the `<service>` tag at the bottom of the WSDL identifies the URL that you should always use to invoke the Web service.

WSDL definitions

The following URLs expose the WSDL (Web Services Description Language) files that define the Web services exposed by the Banner Web Services Adapters. The protocol, host, port (if used), and URL string reflect the location of the associated adapter.

WSDLs for campus card Web services

`http://<host>:<port>/campuscard/eligibleCardholderService.wsdl`
`http://<host>:<port>/campuscard/personIdentityService.wsdl`

WSDLs for housing Web services

`http://<host>:<port>/housing/academicPeriodService.wsdl`
`http://<host>:<port>/housing/entityAddressService.wsdl`
`http://<host>:<port>/housing/housingApplicantService.wsdl`
`http://<host>:<port>/housing/personIdentityService.wsdl`
`http://<host>:<port>/housing/studentAccountService.wsdl`
`http://<host>:<port>/housing/studentDepositService.wsdl`



3 Verify the Configuration



The open source soapUI tool, available from eviware (www.soapui.org), can be used to test exposed Web services. A soapUI project is delivered with each Banner® Web Services Adapter to perform the following functions:

- Verify that the Banner Web Services Adapter is deployed and configured correctly.
- Verify that Banner Web services have access to the data sources and the Banner Translation Service.
- Detail problems with an incorrectly configured data source (for example, a bad user ID or password for connecting to the Banner database).

These functions are achieved by sending a test SOAP request to a corresponding application that is deployed with the Banner Web Services Adapter, but has a different endpoint URL.

This chapter gives instructions for using soapUI to verify your Banner Web services configuration.

Verification steps



Use the following steps to verify the Banner Web services configuration:

- [Step 1, “Download and install soapUI”](#)
- [Step 2, “Open the testing workspace”](#)
- [Step 3, “Import the soapUI project”](#)

Step 1 Download and install soapUI

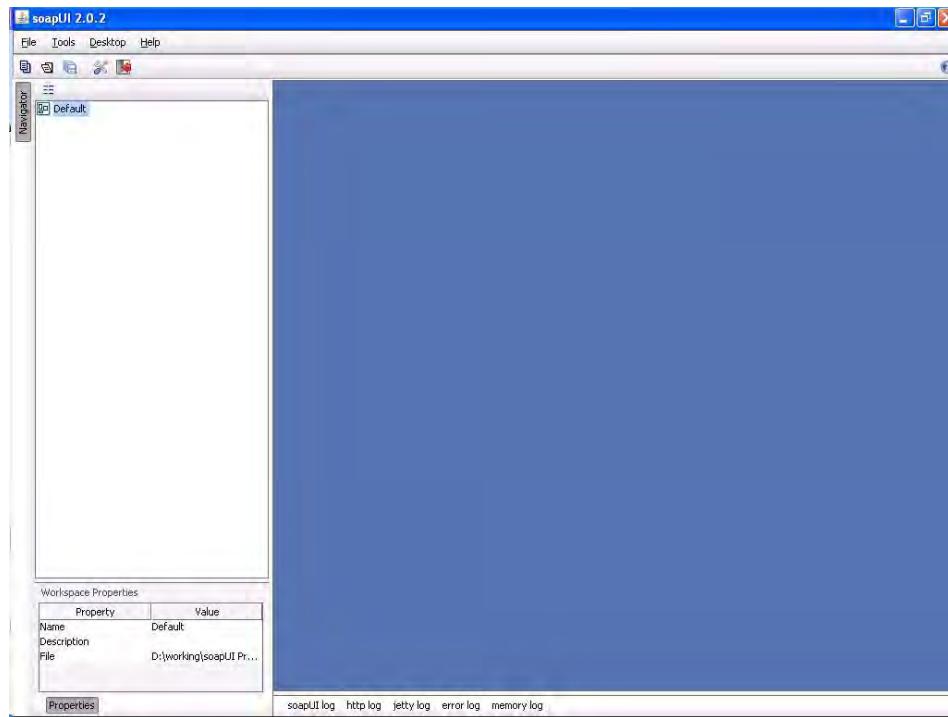
Download soapUI from eviware (www.soapui.org) and install it.



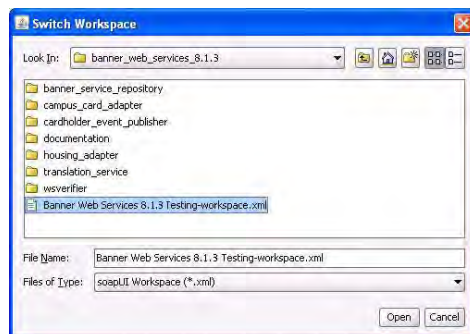
Step 2 Open the testing workspace

In soapUI, workspaces contain projects. Projects contain Web services definitions. The Banner Web services testing workspace is located in the root directory of the unzipped Banner Web services download. Use the following steps to open the testing workspace.

1. Open soapUI. A default workspace is displayed in the Navigator.



2. Select Switch Workspace from the File menu. The Switch Workspace window is displayed.
3. Navigate to Banner Web Services 8.1.3 Testing-workspace.xml in the root directory of the unzipped Banner Web services download:



4. Click **Open**. The testing workspace is displayed.

Step 3 Import the soapUI project

Use the following steps to import the soapUI project.

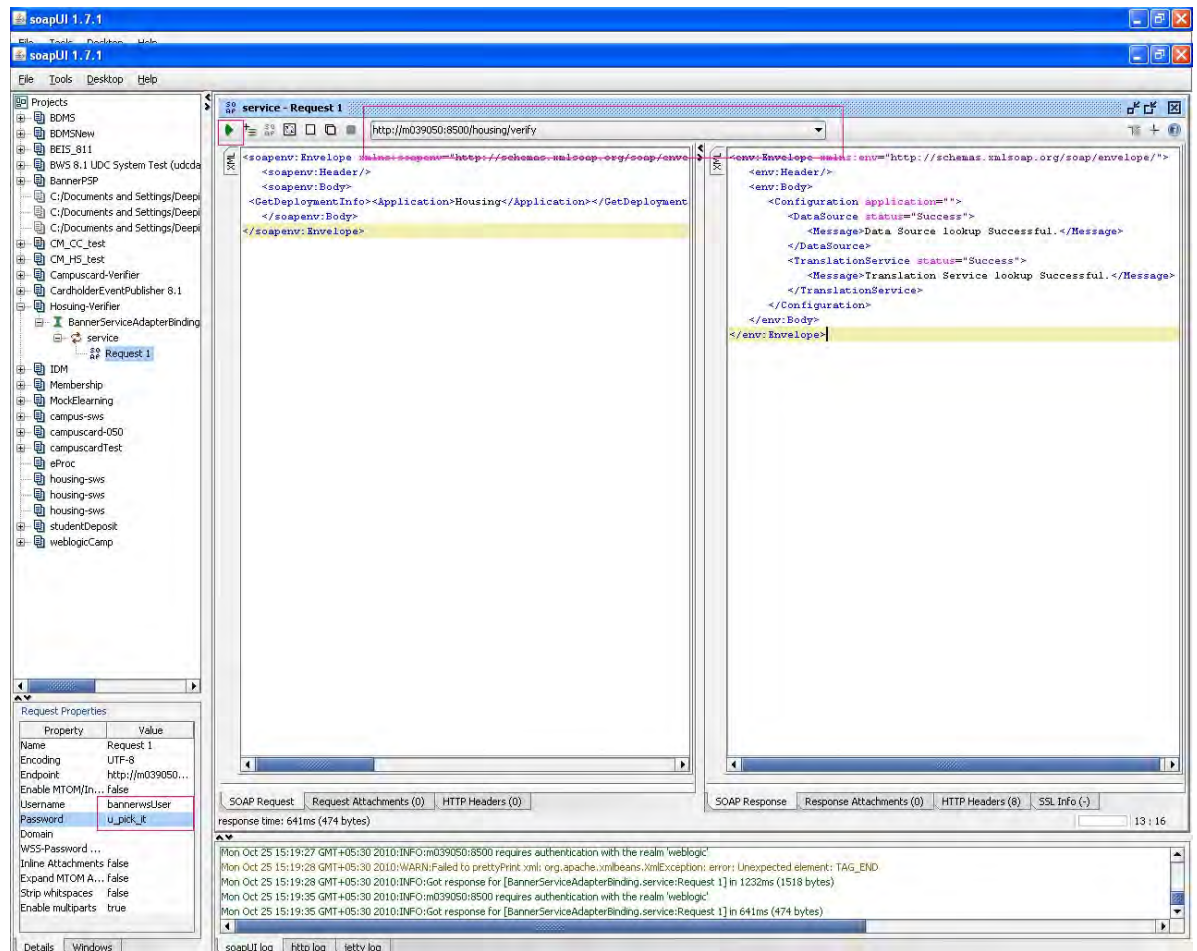
1. Select Import Project from the File menu. The Select soapui project file window is displayed.
2. Navigate to the soapUI project in the corresponding adapter subdirectory of the unzipped Banner Web services download:

`Campuscard-Verifier-soapui-project.xml`

-or-

`Housing-Verifier-soapui-project.xml`

3. Click **Open**. The workspace for the project is displayed with a single node, **BannerServiceAdapterBinding**.
4. Expand the **BannerServiceAdapter Binding** node and double-click the request tag. A service Request 1 window is displayed.
5. Edit the endpoint URL information and the username/password used to access the URL.
6. Click **Run**. The right pane displays the test results.



Test results

The following sections describe the results that you can get when the configuration is tested with the soapUI tool.

No errors

If the Banner Web Services Adapter for Housing Systems is installed and configured correctly, you get the following response message:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <Configuration application="Housing">
```

```

    <DataSource status="Success">
      <Message>Data Source lookup Successful.</Message>
    </DataSource>
    <TranslationService status="Success">
      <Message>Translation Service lookup Successful.
      </Message>
    </TranslationService>
  </Configuration>
</env:Body>
</env:Envelope>

```

If the Banner Web Services Adapter for Campus Card Systems is installed and configured correctly, you get the following response message:

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/
envelope/">
  <env:Header/>
  <env:Body>
    <Configuration application="Campuscard">
      <DataSource status="Success">
        <Message>Data Source lookup Successful.</Message>
      </DataSource>
      <TranslationService status="Success">
        <Message>Translation Service lookup Successful.
        </Message>
      </TranslationService>
    </Configuration>
  </env:Body>
</env:Envelope>

```

Data source configuration errors

If the data source has errors (JNDI lookup is not found or is misconfigured), then the `DataSource status` attribute of the XML response embedded in the response shows "Failure". The following are examples:

```

<Configuration application="Housing">
  <DataSource status="Failure">
    <Message>Error getting Connection from data
    source:DataSource lookup failed. jdbc/bannerws not
    found
    </Message>
  </DataSource>

```

```

    <TranslationService status="Success">
      <Message>Translation Service lookup Successful.</Message>
    </TranslationService>
  </Configuration>

<Configuration application="CampusCard">
  <DataSource status="Failure">
    <Message>Error getting Connection from data
      source:DataSource lookup failed. jdbc/bannerws not
      found
    </Message>
  </DataSource>
  <TranslationService status="Success">
    <Message>Translation Service lookup Successful.</Message>
  </TranslationService>
</Configuration>

```

Banner Translation Service configuration errors

If the Banner Translation Service is not found, an error is displayed in the response and the `TranslationService status` attribute shows "Failure". The following are examples:

```

<Configuration application="Housing">
  <DataSource status="Success">
    <Message>Data Source lookup Successful.</Message>
  </DataSource>
  <TranslationService status="Failure">
    <Message>Translation Service lookup failed. Error
      instantiating web-app JNDI-context: No location specified
      and no suitable instance of the type
      com.sct.translation.ejb.local.
      TranslationServiceLocal' found for the ejb-local-ref
      TranslationServiceBean
    </Message>
  </TranslationService>
</Configuration>

<Configuration application="CampusCard">
  <DataSource status="Success">
    <Message>Data Source lookup Successful.</Message>
  </DataSource>
  <TranslationService status="Failure">

```

```
<Message>Translation Service lookup failed. Error
instantiating web-app JNDI-context: No location specified
and no suitable instance of the
typecom.sct.translation.ejb.local.
TranslationServiceLocal' found for the ejb-local-ref
TranslationServiceBean
</Message>
</TranslationService>
</Configuration>
```

To correct this error, undeploy and redeploy the adapter. The Banner Translation Service must be deployed before the adapter is deployed. If installed properly, the Banner Translation Service lookup should not fail.



4 Import Translations



For Banner® Web services to function, the Banner Translation Service must be seeded with translations that map Banner codes to enterprise values. This is achieved by importing the translations using a specific XML format.

Delivered translation files



Two types of translation files are delivered with Banner Web services:

- .xml files contain enterprise value translations for Banner codes. These translations are unlikely to change and can be imported directly into the Banner Translation Service. These files should *not* be edited before importing.
- .sql files are used to extract Banner support table values for use in translation. These files should be run from SQL*Plus against your Banner instance. Each script spools an output .xml file that contains institution-specific translation values that can be imported into the Banner Translation Service.

Files for campus card systems

The following translation files are located in the `campus_card_adapter/scripts` directory:

```
rco_cardholder_gender.xml
rco_country_code.sql
```

The .sql file is used to create the following .xml file:

SQL File Name	Enterprise Value	Banner Value	Output File Name
rco_country_code.sql	STVNATN_NATION	STVNATN_CODE	rco_country_code.xml

Files for housing systems

The following translation files are located in the `housing_adapter/scripts` directory:

```
rco_category.xml
rco_gender.xml
dar_deposit_amount_currency.sql
```

```

dar_transaction_amount_currency.sql
dpa_country_code.sql
rar_card_type.sql
rco_country_code.sql

```

The .sql files are used to create the following .xml files:

SQL File Name	Enterprise Value	Banner Value	Output File Name
dar_deposit_amount_currency.sql	GTVCURR_TITLE	GTVCURR_CURR_CODE	dar_deposit_amount_currency_code.xml
dar_transaction_amount_currency.sql	GTVCURR_TITLE	GTVCURR_CURR_CODE	dar_transaction_amount_currency.xml
dpa_country_code.sql	STVNATN_NATION	STVNATN_CODE	dpa_country_code.xml
rar_card_type.sql	GTVCCRD_DESC	GTVCCRD_CODE	rar_card_type.xml
rco_country_code.sql	STVNATN_NATION	STVNATN_CODE	rco_country_code.xml

Import steps

Use the following steps to import enterprise value translations into the Banner Translation Service:

- [Step 1, “Extract Banner-specific translation values”](#).
- [Step 2, “Import translation values”](#).

Step 1 Extract Banner-specific translation values

Run the .sql files from SQL*Plus against your Banner instance. The scripts should be run by an Oracle user with select permissions on the tables where the data values are selected. Each script spools an output .xml file that contains the translation values needed for your Banner instance.

Step 2 Import translation values

Use the following steps to import translation values from each .xml file into the Banner Translation Service. You must import values from the .xml files delivered with Banner Web services as well as values from the .xml files created by the delivered .sql scripts.

1. Access the following URL:

```
http://<host>:<port>/transsvc
```

where <host> is the server name and <port> is the http port number of the server where the Banner Translation Service is installed.

2. Click **LOGIN**.
3. Enter the appropriate username and password.
4. Click **OK**.
5. Click **List** under the Enterprise Field menu.
6. Click **Import** in the Enterprise Fields section. The Load Enterprise Field/s page is displayed.
7. Click **Browse**.
8. Navigate to the file to be imported and select it.
9. Click **Open**. The Load Enterprise Field/s page is redisplayed.
10. Click **Load**. The Import Completed page is displayed.
11. Click **Continue**. An updated list of enterprise field names is displayed.
12. Repeat steps 6 through 11 to import each translation file.



5 Customize Web Service Responses

Underlying APIs contain the business logic for the Banner® Web services and provide configuration rules and options. These rules and options are controlled by entries on the following forms:

- Business Rules Form (GORRSQL)
- Crosswalk Validation Form (GTVSDAX)

Entries on these forms are loaded during the Banner installation and contain data that may or may not be valid for your institution. Entries on GTVSDAX, in particular, must be scrutinized because they are delivered with a generic entry of *UPDATE_ME* instead of values that are valid for your institution.

This chapter provides instructions for validating the GORRSQL and GTVSDAX settings and summarizes the settings that are used by Banner Web services. Refer to the *Banner Web Services Handbook* for more details on the settings.

Scripts that check configuration settings

The following SQL scripts in the Banner Web services download provide a quick glimpse of the configuration parameters required by the Banner Web Services Adapters:

Adapter	Script Name	Script Location
Campus Card Systems	campuscard_check.sql	\campus_card_adapter\ scripts
Housing Systems	housing_check.sql	\housing_adapter\scripts

The scripts validate the GORRSQL and GTVSDAX settings. The scripts also verify the existence of the required API packages and object types.

Execute the scripts via an SQL*Plus session to the database. The following figure displays a successful result for the Web services associated with campus card system integration.

```

Untitled - Notepad
File Edit Format View Help
+-----+
|Check #1 |Check process API packages. |
+-----+
RESULT: Passed
|
+-----+
|Check #2 |Check object types required by API packages. |
+-----+
RESULT: Passed
|
+-----+
|Check #3 |Check gtvsdax settings. |
+-----+
RESULT: Warnings
Setting not found: INTCOMP/USERSOURCE
+-----+
|Check #4 |Visually inspect gtvsdax settings. |
+-----+

GTVSDEX_INTERNAL_CODE_GROUP          GTVSDEX_INTERNAL_CODE GTVSDEX_EXTERNAL_CODE
-----
ADDRESS                               CC_LOCAL              MA
ADDRESS                               CC_LOCAL              TE
ADDRESS                               CC_FERM               FI
ADDRESS                               CC_FERM               PR
ADDRESS                               CC_RESIDE             RH
ADDRESS                               CC_RESIDE             SC
ASSIGNMENTSTATUS                     ACTIVEHEAL            AC
ASSIGNMENTSTATUS                     ACTIVEPHON            AC
DATASOURCE                           CC_RESADDR            P
DATASOURCE                           CC_RESPHON            P
EMAIL                                 CC_EMAIL              HOME
EMAIL                                 CC_EMAIL              FERS
EMAIL                                 CC_EMAIL              SCHL
TELEPHONE                             CC_LOCAL              MA
TELEPHONE                             CC_LOCAL              TE
TELEPHONE                             CC_FERM               PA
TELEPHONE                             CC_FERM               PR
TELEPHONE                             CC_RESIDE             RH
TELEPHONE                             CC_RESIDE             SC
TELEPHONE                             CC_WORK               BU
+-----+
|Check #5 |Check gorrsq rules. |
+-----+
RESULT: Passed

```

The script displays warnings if settings are missing or are not updated, as shown in the following figure.

```

Untitled - Notepad
File Edit Format View Help
+-----+
|Check #3 |Check gtvsdax settings. |
+-----+
RESULT: Warnings
Setting should not be "UPDATE ME": DEPOSITALLTERM/HOUSINGINT
Setting should not be "UPDATE ME": DEPOSITSHOWTERM/HOUSINGINT
Setting not found: INTCOMP/USERSOURCE
.

```

GORRSQL rules

The following rules are created on the Business Rules Form (GORRSQL). Refer to the *Banner Web Services Handbook* for more details.

Banner Web Service	GORRSQL Process Code	Description
GetEligibleCardholder and SyncEligibleCardholder	CARDHOLDER_ROLES	Defines the criteria for assigning institution-defined cardholder roles to extracted <code>EligibleCardholder</code> XML objects. Campus card systems can use roles to assign card privileges to specific cardholders.
GetHousingApplicant Eligibility	HOUSING_ELIGIBILITY	Defines the criteria for assigning institution-defined housing applicant roles to extracted <code>HousingApplicant</code> XML objects. Housing systems can use roles to determine if an applicant is eligible for specific types of housing (for example, athletic, honors).

GTVSDAX settings

The following settings are entered on the Crosswalk Validation Form (GTVSDAX). Refer to the *Banner Web Services Handbook* for more details.

Banner Web Service	GTVSDAX Internal Code	GTVSDAX Internal Group	Description
AddEntityAddress	INTEG	ADDRTYPE	Address type codes used to create new addresses in Banner
	INTEG	ADDRSRCE	Address source codes used to create new addresses in Banner
GetEligibleCardholder and SyncEligibleCardholder	CC_EMAIL	EMAIL	E-mail address type codes used to select cardholder e-mail addresses
	CC_PERM	ADDRESS	Address type codes used to select cardholder permanent mailing addresses

Banner Web Service	GTVSDAX Internal Code	GTVSDAX Internal Group	Description
	CC_PERM	TELEPHONE	Telephone type codes used to select cardholder permanent telephone numbers
	CC_LOCAL	ADDRESS	Address type codes used to select cardholder local mailing addresses
	CC_LOCAL	TELEPHONE	Telephone type codes used to select cardholder local telephone numbers
	CC_RESADDR	DATASOURCE	Source (SPRADDR or SLRRASG) used to get campus residence locations
	CC_RESIDE	ADDRESS	Address type codes used to select cardholder campus residence addresses (if SPRADDR is used to get campus residence locations)
	ACTIVEROOM	ASSIGNMENTSTATUS	Room assignment status codes used to select active room assignments (if SLRRASG is used to get campus residence locations)
	CC_RESPHON	DATASOURCE	Source (SPRTELE or SLRPASG) used to get campus residence location telephone numbers
	CC_RESIDE	TELEPHONE	Telephone type codes used to select cardholder campus residence telephone numbers (if SPRTELE is used to get campus residence location telephone numbers)
	ACTIVEPHON	ASSIGNMENTSTATUS	Phone assignment status codes used to select active telephone assignments (if SLRPASG is used to get campus residence location telephone numbers)

Banner Web Service	GTVSDAX Internal Code	GTVSDAX Internal Group	Description
	CC_WORK	TELEPHONE	Telephone type codes used to select cardholder work telephone numbers
	ACTIVEMEAL	ASSIGNMENTSTATUS	Meal plan assignment status codes used to identify active meal plan assignments for cardholders
GetHousingApplicant Eligibility	HOUSINGINT	DEPOSITALLTERM	Deposit type codes used to select deposit records for summarization, if the term is not included in the summary
	HOUSINGINT	DEPOSITSHOWTERM	Deposit type codes used to select deposit records for summarization, if the term is included in the summary
	HOUSINGINT	DEPOSITSPECIFICTERM	Deposit type codes used to select deposit records for summarization, if a specific term is specified
	HOUSINGINT	FEEALLTERM	Detail codes used to select housing-related fee records for summarization, if the term is not included in the summary
	HOUSINGINT	FEESHOWTERM	Detail codes used to select housing-related fee records for summarization, if the term is included in the summary
	HOUSINGINT	FEESPECIFICTERM	Detail codes used to select housing-related fee records for summarization, if a specific term is specified
GetHousingApplicant Profile	HOUSINGINT	ADDRESS	Address type codes used to select applicant mailing addresses

Banner Web Service	GTVSDAX Internal Code	GTVSDAX Internal Group	Description
	HOUSINGINT	TELEPHONE	Telephone type codes used to select applicant telephone numbers
	HOUSINGINT	EMAIL	E-mail address type codes used to select applicant e-mail addresses
	HOUSINGINT	SPORT	Activity codes used to select applicant athletic participation information
	HOUSINGINT	ACTIVITY	Activity codes used to select applicant extracurricular activity participation information
GetPersonIdentity	INTEG	CM_SOURCE_CODE	Set of Common Matching rules used to locate persons in Banner if user INTEGMGR (user ID that connects integration software with the Banner database) does not have a default Common Matching source code defined on the Common Matching User Setup Form (GORCMUS)

6 Test Banner Web Services

Once the Banner® Web Services Adapters are installed and the appropriate WSDL files are modified, you should test the exposed Banner Web services. This step is optional but highly recommended.

The degree of testing depends on the data and tools that your institution uses. For this reason, this section provides testing guidelines rather than specific steps.

Test method

Ellucian highly recommends the use of an interactive testing tool that provides visibility into Web service request and response messages. Such a tool helps functional and technical users understand the options available for specific Banner Web services and their effect on the content of response messages. Users can create schema-compliant XML messages for their specific environment, quickly inspect the results, modify Web service settings in Banner (if necessary), and execute the same request to see the resulting differences.

Several commercial and open source Web service testing tools provide these capabilities. One open source tool is eviware soapUI (www.soapui.org). This tool uses a WSDL document and associated XML schema as input to generate compliant request message stubs to which data can be added for calling the associated service. Response messages are displayed in an adjacent window for easy visibility. The tool also allows individual service requests to be tied together to form larger test suites.

Manually creating Web service requests and visually inspecting responses provide practical insight into the data returned by the Web services under specific conditions. In addition, issues with the underlying Banner configuration for each Web service are noticed more readily.

Whatever tool is used, be sure to reference the *Banner Web Services Handbook* to understand the messages, valid input data, and configuration options for each Banner Web service that is deployed in your environment.

Web services for campus card systems

The Banner Web Services Adapter for Campus Card Systems supports two Banner Web services:

- GetPersonIdentity
- GetEligibleCardholder

The following tests can help you understand these Web services. These tests are examples only. They do not reflect actual data in your database.

GetPersonIdentity

This Banner Web service finds the unique identifier of a person in Banner for use in subsequent Web service calls. It allows for various combinations of person data to limit the result set.

SPRIDEN ID

Test GetPersonIdentity using a known SPRIDEN ID.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetPersonIdentity>
      <urn:UnidentifiedPerson>
        <urn1:PersonIdentifier>
          <urn2:IdValue name="BannerUID">977111370
        </urn2:IdValue>
        </urn1:PersonIdentifier>
      </urn:UnidentifiedPerson>
    </urn:GetPersonIdentity>
  </soapenv:Body>
</soapenv:Envelope>
```

Last name

Test GetPersonIdentity using a known last name.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
```

```

<soapenv:Header/>
<soapenv:Body>
  <urn:GetPersonIdentity>
    <urn:UnidentifiedPerson>
      <urn1:PersonName>
        <urn2:FamilyName>Mitchell</urn2:FamilyName>
      </urn1:PersonName>
    </urn:UnidentifiedPerson>
  </urn:GetPersonIdentity>
</soapenv:Body>
</soapenv:Envelope>

```

First name and qualifying data

Test `GetPersonIdentity` with a first name and qualifying data, such as a phone number.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
  xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
  xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
  xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetPersonIdentity>
      <urn:UnidentifiedPerson>
        <urn1:PersonName>
          <urn2:GivenName>James</urn2:GivenName>
          <urn2:FamilyName>Mitchell</urn2:FamilyName>
        </urn1:PersonName>
        <urn1:ContactPhone>
          <urn2:SubscriberNumber>5551212
          </urn2:SubscriberNumber>
        </urn1:ContactPhone>
      </urn:UnidentifiedPerson>
    </urn:GetPersonIdentity>
  </soapenv:Body>
</soapenv:Envelope>

```

GetEligibleCardholder

This Banner Web service returns all known “eligible cardholder” data for a person based on a given Banner identifier. For a definition of the `EligibleCardholder` structure and the customization options, refer to the *Banner Web Services Handbook*.

The response messages in the following examples might vary, but the request message is similar, if not the same:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetEligibleCardholder>
      <urn:CardholderIdentifier>
        <urn1:IdValue name="BannerUID">977111370
        </urn1:IdValue>
      </urn:CardholderIdentifier>
    </urn:GetEligibleCardholder>
  </soapenv:Body>
</soapenv:Envelope>
```

SPRIDEN ID

Test GetEligibleCardholder for a basic person using a known SPRIDEN ID.

Delivered roles

Test delivered roles:

1. Ensure that delivered cardholder roles are enabled on the Business Rules Form (GORRSQL).
2. Test GetEligibleCardholder for a person who should have one of the roles.
3. Ensure that the ShowEligibleCardholder message includes the role.

New cardholder role

Test a new cardholder role:

1. Create a new cardholder role on GORRSQL for criteria that can be easily applied to a person (for example, a specific e-mail address domain or a specific address).
2. Test GetEligibleCardholder for a person that matches the cardholder role criteria.
3. Ensure that the ShowEligibleCardholder message includes the role.

Changed GTVSDAX settings

Test changed GTVSDAX settings:

1. Change settings on the Crosswalk Validation Form (GTVSDAX) for cardholder contact details, residence location, work location, or meal plan assignment status.
2. Retest GetEligibleCardholder for a known person.
3. Ensure that the ShowEligibleCardholder messages match expected results.

Web services for housing systems

The Banner Web Services Adapter for Housing Systems supports the following Banner Web services:

- GetPersonIdentity
- GetHousingApplicantEligibility
- GetHousingApplicantProfile
- GetAcademicPeriods
- AddEntityAddress
- ExpireEntityAddress
- AddStudentDeposit
- ReleaseStudentDeposit
- AddStudentAccountTransaction

The following tests can help you understand the Web services. These tests are examples only. They do not reflect actual data in your database.

GetPersonIdentity

This Banner Web service finds the unique identifier of a person in Banner for use in subsequent Web service calls. It allows for various combinations of person data to limit the result set.

GetPersonIdentity is exposed by both the Banner Web Services Adapter for Campus Card Systems and the Banner Web Services Adapter for Housing Systems. Therefore, the tests are identical. See [“GetPersonIdentity” on page 76](#) for testing guidelines.

GetHousingApplicantEligibility

This Banner Web service returns “eligibility” data for a person that might be applying for housing at the institution. For a definition of the `GetHousingApplicantEligibility` structure and the customization options, refer to the *Banner Web Services Handbook*.

The response messages in the following examples might vary, but the request message is similar, if not the same:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:residentiallife:1.0"
" xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetHousingApplicantEligibility>
      <urn:SearchCriteria>
        <urn1:HousingApplicantIdentifier>
          <urn2:IdValue name="BannerUID">A00012649
          </urn2:IdValue>
        </urn1:HousingApplicantIdentifier>
        <urn1:AcademicPeriodIdentifier type="Term"
          id="200711"/>
        </urn:SearchCriteria>
      </urn:GetHousingApplicantEligibility>
    </soapenv:Body>
  </soapenv:Envelope>
```

SPRIDEN ID and term code

Test `GetHousingApplicantEligibility` for a student using a known SPRIDEN ID and term code.

Delivered roles

Test delivered roles:

1. Ensure that delivered housing applicant roles are enabled on the Business Rules Form (GORRSQL).
2. Test `GetHousingApplicantEligibility` for a person who should have one of the roles.
3. Ensure that the `ShowHousingApplicantEligibility` message includes the role.

Deposit information

Test deposit information:

1. Review the section on Deposit Information in chapter 8 of the *Banner Web Services Handbook*.
2. Update settings on the Crosswalk Validation Form (GTVSDAX) for the service with valid deposit type codes from the Deposit Type Code Validation Form (TTVDTYP) for each deposit summarization.
3. Create deposits for a student or find a student with deposits.
4. Retest GetHousingApplicantEligibility using the student's SPRIDEN ID.
5. Ensure that the ShowHousingApplicantEligibility message includes the appropriate deposit summarization.

Fee information

Test fee information:

1. Review the section on Fee Information in chapter 8 of the *Banner Web Services Handbook*.
2. Update settings on GTVSDAX for the service with valid detail codes from the Student Account Detail Form (TSADETL) for each fee summarization.
3. Create fees for a student or find a student with fees.
4. Retest GetHousingApplicantEligibility using the student's SPRIDEN ID.
5. Ensure that the ShowHousingApplicantEligibility message includes the appropriate fee summarization.

Holds

Test holds:

1. Create holds for a student.
2. Retest GetHousingApplicantEligibility using the student's SPRIDEN ID.
3. Ensure that the ShowHousingApplicantEligibility message includes the holds.

GetHousingApplicantProfile

This Banner Web service returns detailed information about a housing applicant. For a definition of the `HousingApplicantProfile` structure and the customization options, refer to the *Banner Web Services Handbook*.

The response messages in the following examples might vary, but the request message is similar, if not the same:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:residentiallife:1.0"
" xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetHousingApplicantProfile>
      <urn:SearchCriteria>
        <urn1:HousingApplicantIdentifier>
          <urn2:IdValue name="BannerUID">977111370
          </urn2:IdValue>
        </urn1:HousingApplicantIdentifier>
        <urn1:AcademicPeriodIdentifier type="Term"
          id="200711"/>
      </urn:SearchCriteria>
    </urn:GetHousingApplicantProfile>
  </soapenv:Body>
</soapenv:Envelope>
```

SPRIDEN ID and term code

Test `GetHousingApplicantProfile` for a student using a known SPRIDEN ID and term code.

Expanded profile

Add information to the student to expand the profile and retest. Information that can be added and verified includes privacy information, biographic data, demographic data, military service, driver's license information, mailing addresses, medical records, athletic participation, and extracurricular activities. Refer to the *Banner Web Services Handbook* for the `HousingApplicantProfile` structure and more information on its content.

Changed GTVSDAX settings

Test changed GTVSDAX settings:

1. Change settings on the Crosswalk Validation Form (GTVSDAX) for housing applicant contact details and participation information.
2. Retest GetHousingApplicantProfile for a known person.
3. Ensure that response messages match expected results.

GetAcademicPeriods

This Banner Web service returns information about a Banner term. This information is helpful for applications that need to create student account charges or search for information based on a term.

All terms

Test GetAcademicPeriods for all terms.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
  xmlns:urn1="urn:sungardhe:enterprise:domain:student:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetAcademicPeriods>
      <urn:SearchCriteria>
        <urn1:TypeOfAcademicPeriod>Term
      </urn1:TypeOfAcademicPeriod>
      </urn:SearchCriteria>
    </urn:GetAcademicPeriods>
  </soapenv:Body>
</soapenv:Envelope>
```

Terms for a specific date

Test GetAcademicPeriods for terms for a specific date.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
  xmlns:urn1="urn:sungardhe:enterprise:domain:student:1.0">
```

```

<soapenv:Header/>
<soapenv:Body>
  <urn:GetAcademicPeriods>
    <urn:SearchCriteria>
      <urn1:TypeOfAcademicPeriod>Term
    </urn1:TypeOfAcademicPeriod>
      <urn1:SearchDate>2005-04-15</urn1:SearchDate>
    </urn:SearchCriteria>
  </urn:GetAcademicPeriods>
</soapenv:Body>
</soapenv:Envelope>

```

AddEntityAddress

This Banner Web service creates a SPRADDR record in Banner. The EntityAddress structure follows an industry standard specification; not all elements are supported by Banner. Refer to the *Banner Web Services Handbook* for a list of supported elements.

Known person

Test creation of an address for a known person in Banner.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
  xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
  xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:AddEntityAddress>
      <urn:AddressSource>SOAPUI</urn:AddressSource>
      <urn:EntityAddress type="RESIADDRESS">
        <urn1:AddressIdentifier>
          <urn2:IdValue name="BannerUID">A00012649
        </urn2:IdValue>
        </urn1:AddressIdentifier>
        <urn1:PostalCode>68502</urn1:PostalCode>
        <urn1:Region>NE</urn1:Region>
        <urn1:Municipality>Lincoln</urn1:Municipality>
        <urn1:DeliveryAddress>
          <urn2:AddressLine>Johnson Tower
        </urn2:AddressLine>
          <urn2:AddressLine>Suite 432</urn2:AddressLine>
        </urn1:DeliveryAddress>
      </urn:EntityAddress>
    </urn:AddEntityAddress>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        </urn1:DeliveryAddress>
    </urn:EntityAddress>
</urn:AddEntityAddress>
</soapenv:Body>
</soapenv:Envelope>

```

Temporary address

Test creation of a temporary address with effective and expiration dates.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:AddEntityAddress>
            <urn:AddressSource>SOAPUI</urn:AddressSource>
            <urn:EntityAddress type="RESIADDRESS"
            validFrom="2007-12-01"
            validTo="2008-05-31">
                <urn1:AddressIdentifier>
                    <urn2:IdValue name="BannerUID">A00012720
                </urn2:IdValue>
                </urn1:AddressIdentifier>
                <urn1:PostalCode>29208</urn1:PostalCode>
                <urn1:Region>SC</urn1:Region>
                <urn1:Municipality>Columbia</urn1:Municipality>
                <urn1:DeliveryAddress>
                    <urn2:AddressLine>Bates Tower</urn2:AddressLine>
                    <urn2:AddressLine>Suite 432</urn2:AddressLine>
                </urn1:DeliveryAddress>
            </urn:EntityAddress>
        </urn:AddEntityAddress>
    </soapenv:Body>
</soapenv:Envelope>

```

ExpireEntityAddress

This Banner Web service expires a SPRADDR record in Banner by updating the address end date. The Web service requires the unique ID of the address, which is returned by the AddEntityAddress Web service.

Known person

Test expiration of an address for a known person in Banner.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:ExpireEntityAddress>
      <urn:EntityAddressExpirationRequest>
        <urn1:EntityAddressIdentifier>
          <urn1:AddressIdentifier>
            <!--1 or more repetitions:-->
            <urn2:IdValue name="BannerUID">A00012649
            </urn2:IdValue>
            <urn2:IdValue name="AddressType">SC
            </urn2:IdValue>
            <urn2:IdValue name="SequenceNumber">1
            </urn2:IdValue>
          </urn1:AddressIdentifier>
        </urn1:EntityAddressIdentifier>
        <urn1:ExpirationDate>2007-12-13
        </urn1:ExpirationDate>
      </urn:EntityAddressExpirationRequest>
    </urn:ExpireEntityAddress>
  </soapenv:Body>
</soapenv:Envelope>
```

Start date and expiration date are the same

Test expiration of an address with the same start date and expiration date.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:party:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:ExpireEntityAddress>
      <urn:EntityAddressExpirationRequest>
```

```

        <urn1:EntityAddressIdentifier>
            <urn1:AddressIdentifier>
                <!--1 or more repetitions:-->
                <urn2:IdValue name="BannerUID">A00012720
                </urn2:IdValue>
                <urn2:IdValue name="AddressType">SC
                </urn2:IdValue>
                <urn2:IdValue name="SequenceNumber">2
                </urn2:IdValue>
            </urn1:AddressIdentifier>
        </urn1:EntityAddressIdentifier>
        <urn1:ExpirationDate>2007-12-01
        </urn1:ExpirationDate>
    </urn:EntityAddressExpirationRequest>
</urn:ExpireEntityAddress>
</soapenv:Body>
</soapenv:Envelope>

```

AddStudentDeposit

This Banner Web service creates deposits in Banner for a given student. Deposits are reflected in the GetHousingApplicantEligibility Web service. Therefore, GetHousingApplicantEligibility can be used to validate the creation of deposits via AddStudentDeposit. For details on the service's options, refer to the *Banner Web Services Handbook*.

New deposit

Test the creation of a deposit for a student.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:ar:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0"
xmlns:urn3="urn:sungardhe:enterprise:resources:ar:1.0">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:AddStudentDeposit>
            <urn:StudentDeposit>
                <urn1:AccountIdentifier>
                    <urn2:IdValue name="BannerUID">IT121405G
                    </urn2:IdValue>
                </urn1:AccountIdentifier>
            </urn:StudentDeposit>
        </urn:AddStudentDeposit>
    </soapenv:Body>
</soapenv:Envelope>

```

```

        <urn1:AcademicPeriodIdentifier type="Term"
        id="200510"/>
        <urn1:DepositTransactionType id="GDD1"/>
        <urn1:TransactionDescription>Housing Reservation
        Deposit
        </urn1:TransactionDescription>
        <urn1:DepositPaymentTransactionType id="GDP1"/>
        <urn1:DepositAmount>500.00</urn1:DepositAmount>
        <urn1:EffectiveDate>2007-04-16</urn1:EffectiveDate>
        <urn1:ReleaseInstructions>
            <urn3:AutoRelease>Yes</urn3:AutoRelease>
            <urn3:DepositReleaseDate>2007-08-29
            </urn3:DepositReleaseDate>
            <urn3:DepositBalanceMinimumAmount>250.00
            </urn3:DepositBalanceMinimumAmount>
            <urn3:DepositExpirationDate>2007-08-29
            </urn3:DepositExpirationDate>
        </urn1:ReleaseInstructions>
        <urn1:CashierSession>
            <urn3:CashierId>WSTEST</urn3:CashierId>
            <urn3:SessionNumber>0</urn3:SessionNumber>
        </urn1:CashierSession>
        <urn1:DepositTransactionSource>
            <urn3:SourceSystem>soapUI</urn3:SourceSystem>
        </urn1:DepositTransactionSource>
        <urn1:DocumentNumber>MMB001</urn1:DocumentNumber>
        <urn1:OverrideHoldChecking>Yes
        </urn1:OverrideHoldChecking>
    </urn:StudentDeposit>
</urn:AddStudentDeposit>
</soapenv:Body>
</soapenv:Envelope>

```

Additional deposits

Create additional deposits to test various release options.

ReleaseStudentDeposit

This Banner Web service is the counterpart to AddStudentDeposit. It allows external systems to request Banner to release funds held on deposit on a student's account. These funds might be released to cover damages charged to a student's account, or they might be remitted to the student at the end of a specified period. This service can be tested in

conjunction with AddStudentDeposit and GetHousingApplicantEligibility. For details on the service's options, refer to the *Banner Web Services Handbook*.

Release of deposits

Request release of deposits of a specific type within a specific term.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:ar:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0"
xmlns:urn3="urn:sungardhe:enterprise:resources:ar:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:ReleaseStudentDeposit>
      <urn:StudentDepositReleaseRequest>
        <urn1:AccountIdentifier>
          <urn2:IdValue name="BannerUID">A00012649
          </urn2:IdValue>
        </urn1:AccountIdentifier>
        <urn1:CashierSession>
          <urn3:CashierId>SYSTEST24</urn3:CashierId>
          <urn3:SessionNumber>0</urn3:SessionNumber>
        </urn1:CashierSession>
        <urn3:ForceRelease>No</urn3:ForceRelease>
        <urn3:DepositReleaseTransactionType>%
        </urn3:DepositReleaseTransactionType>
        <urn3:AcademicPeriodIdentifier type="Term"
        id="200711"/>
        <urn3:DepositType>HOU</urn3:DepositType>
      </urn:StudentDepositReleaseRequest>
    </urn:ReleaseStudentDeposit>
  </soapenv:Body>
</soapenv:Envelope>
```

Forced release

Force release of all deposits.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:ar:1.0"
```

```

xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0"
xmlns:urn3="urn:sungardhe:enterprise:resources:ar:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:ReleaseStudentDeposit>
      <urn:StudentDepositReleaseRequest>
        <urn1:AccountIdentifier>
          <urn2:IdValue name="BannerUID">IT121405G
          </urn2:IdValue>
        </urn1:AccountIdentifier>
        <urn1:CashierSession>
          <urn3:CashierId>WSTEST</urn3:CashierId>
          <urn3:SessionNumber>0</urn3:SessionNumber>
        </urn1:CashierSession>
        <urn3:ForceRelease>Yes</urn3:ForceRelease>
        <urn3:DepositReleaseTransactionType>%
        </urn3:DepositReleaseTransactionType>
        <urn3:AcademicPeriodIdentifier id="Term"/>
        <urn3:DepositType>AP2</urn3:DepositType>
      </urn:StudentDepositReleaseRequest>
    </urn:ReleaseStudentDeposit>
  </soapenv:Body>
</soapenv:Envelope>

```

AddStudentAccountTransaction

This Banner Web service and its siblings, AddStudentAcctTransSource and AddStudentAcctTransSystem, allow external systems to request that a charge or payment be recorded on a student account in Banner. Refer to the *Banner Web Services Handbook* for an explanation of differences among these services.

New charge

Test the creation of a charge, and verify the creation of the TBRACCD record in Banner.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:ar:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0"
xmlns:urn3="urn:sungardhe:enterprise:resources:ar:1.0">
  <soapenv:Header/>
  <soapenv:Body>

```

```

<urn:AddStudentAccountTransaction>
  <urn:StudentAccountTransaction>
    <urn1:AccountIdentifier>
      <urn2:IdValue name="BannerUID">IT121405G
      </urn2:IdValue>
    </urn1:AccountIdentifier>
    <urn1:AcademicPeriodIdentifier type="Term"
    id="200510"/>
    <urn1:TransactionType id="TECH"/>
    <urn1:TransactionSource>
      <urn3:SourceSystem>MMB001</urn3:SourceSystem>
    </urn1:TransactionSource>
    <urn1:TransactionAmount>50.00
    </urn1:TransactionAmount>
    <urn1:TransactionDescription>Technology Fee
    </urn1:TransactionDescription>
    <urn1:EffectiveDate>2007-10-16</urn1:EffectiveDate>
    <urn1:CashierSession>
      <urn3:CashierId>WSTEST</urn3:CashierId>
      <urn3:SessionNumber>0</urn3:SessionNumber>
    </urn1:CashierSession>
  </urn:StudentAccountTransaction>
</urn:AddStudentAccountTransaction>
</soapenv:Body>
</soapenv:Envelope>

```

New payment

Test the creation of a payment and verify the creation of the TBRACCD record in Banner.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:urn="urn:sungardhe:enterprise:ws:messages:1.0"
xmlns:urn1="urn:sungardhe:enterprise:domain:ar:1.0"
xmlns:urn2="urn:sungardhe:enterprise:resources:common:1.0"
xmlns:urn3="urn:sungardhe:enterprise:resources:ar:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:AddStudentAccountTransaction>
      <urn:StudentAccountTransaction>
        <urn1:AccountIdentifier>
          <urn2:IdValue name="BannerUID">IT121405G
          </urn2:IdValue>

```

```
</urn1:AccountIdentifier>
<urn1:AcademicPeriodIdentifier type="Term"
id="200510"/>
<urn1:TransactionType id="GZP1"/>
<urn1:TransactionSource>
  <urn3:SourceSystem>MMB001</urn3:SourceSystem>
</urn1:TransactionSource>
<urn1:TransactionAmount>125.56
</urn1:TransactionAmount>
<urn1:TransactionDescription>Payment by Check
</urn1:TransactionDescription>
<urn1:CashierSession>
  <urn3:CashierId>WSTEST</urn3:CashierId>
  <urn3:SessionNumber>0</urn3:SessionNumber>
</urn1:CashierSession>
<urn1:PaymentDetails>
  <urn3:PaymentID>206</urn3:PaymentID>
</urn1:PaymentDetails>
</urn:StudentAccountTransaction>
</urn:AddStudentAccountTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

7 Install Banner Cardholder Event Publisher



The Banner® Cardholder Event Publisher pushes cardholder data changes to a defined external campus card system. The Publisher works with defined Oracle Streams capture and apply processes and Oracle Advanced Queuing to recognize cardholder data changes in Banner and to publish corresponding SyncEligibleCardholder messages. The Publisher must be installed and configured for the external system to receive messages.

This chapter gives instructions for installing the Banner Cardholder Event Publisher on Oracle Application Server 10.1.3.4/5 and Oracle WebLogic Server 11g.

Requirements



The Banner Cardholder Event Publisher requires the following components.

External campus card system

The external card system that receives cardholder data changes from the Publisher must meet the following criteria:

- Implement the SyncEligibleCardholder Web service interface (see the *Banner Web Services Handbook*)
- Implement SOAP binding
- Expose a compliant endpoint

Oracle application server and Java

The Banner Cardholder Event Publisher is certified on Oracle Application Server (OAS) 10.1.3.4/5 and Oracle WebLogic Server 11g with Java 1.6. All development, testing, and support take place on these application servers. These are the recommended application servers for deploying the Publisher.

Note

Ellucian provides limited support for previous versions of the Publisher that were certified on OAS 10.1.2.x. If you have an issue that cannot be replicated in OAS 10.1.3x or Oracle WebLogic Server 11g, with Java 1.6, you are strongly encouraged to move to a supported application server. ■



OAS 10.1.3.4/5 is delivered with Java 1.5. The following Oracle document provides instructions for changing to Java 1.6. If you contract with Ellucian for Oracle support, you can access the FAQ on the Customer Support Center. Otherwise, you can use your Oracle support account to access the document.

Document Title:	How to change the Java version used to run a specific OC4J instance
Ellucian FAQ:	1-AXZ803
Oracle Doc ID:	351476.1

Oracle database

The required Oracle database depends on the application server that you are using:

Application Server	Required Database
Oracle Application Server 10.1.3.4/5	Oracle Database 10gR2 or 11g
Oracle WebLogic Server 11g	Oracle Database 11g

Banner Translation Service

You *must* install the Banner Translation Service before you deploy the Banner Cardholder Event Publisher. Refer to the *Banner Translation Service Installation and Administration Guide* for details.

Banner dependency

The Banner Cardholder Event Publisher requires the Oracle Streams infrastructure that was introduced in Banner General 7.5.1. If you are installing the Publisher in a 7.x environment, two additional patches are required:

- Patch p1-33tlib_gen70501 adds the rules for the cardholder capture process, the cardholder apply handler API, and queue. The rules, API and queue together publish changes to data that constitute a cardholder XML object.
- Patch p1-4iftnp_gen7050101 provides an API for the mass publication of messages for initial and periodic synchronization of cardholder data.

If you are installing the Publisher in a Banner 8.x environment, the cardholder capture rules should already be loaded in the database as a result of seed data scripts for the GTVSQRUI, GTVSQPRI, GORCTAB, GORCRUL, and GORCCOL tables. Likewise, the cardholder apply handler API should have been applied during the creation of the gp_cardholder package. You must, however, install patch p1-46c8mj_gen80100, which

addresses issues with the publication of international data and the mass publication of messages for initial and periodic synchronization of cardholder data.

Installation on Oracle Application Server 10.1.3.4/5

The Banner Cardholder Event Publisher is packaged as a J2EE compatible enterprise archive file named `CardholderEventPublisher_v8.1.3.ear`. Use the following steps to install the Publisher on OAS 10.1.3.4/5:

- [Step 1, “Verify the capture process rules”](#).
- [Step 2, “Create, configure, and start the Oracle Streams processes”](#).
- [Step 3, “Define the data source for Oracle Advanced Queuing”](#)
- [Step 4, “Define the data source for the bulk load process”](#)
- [Step 5, “Define the data source for the Oracle Streams administrator”](#)
- [Step 6, “Install the Publisher”](#)
- [Step 7, “Verify the instance configuration”](#).
- [Step 8, “Configure the security role and user”](#)

A new OC4J instance was created when the Banner Translation Service was installed. The Publisher must be deployed in this instance with the Banner Translation Service and the Banner Web Service Adapter for Campus Card Systems.

Step 1 Verify the capture process rules

The capture rules for cardholder data should already be loaded in the database. Capture rules are initially provided in Banner General seed data scripts, as described in [“Banner dependency” on page 94](#). Additional capture rules are provided with the Banner Web Services download in the following directory:

```
\cardholder_event_publisher\scripts\capture_rules
```

Use the steps in Appendix B, “Using Oracle Streams,” of the *Banner Web Services Handbook* to verify that the rules are loaded in the database.

Step 2 Create, configure, and start the Oracle Streams processes

Use the steps in Appendix B, “Using Oracle Streams,” of the *Banner Web Services Handbook* to create, configure, and start the Oracle Streams capture and apply processes. These steps accomplish the following:

- Create buffered queues and queue tables to manage events.
- Create supplemental, primary key, and unique key log groups for the configured tables.
- Configure the DML callback handler for the apply process.
- Set the instantiation SCN for the tables in the apply process.

 **Note**

These steps should be performed as the `streamsadmin` Oracle user only. ■

Refer to the *Banner Web Services Handbook* for more information about using Oracle Streams.

Step 3 Define the data source for Oracle Advanced Queuing

A data source provides the connection properties to the Banner database. A data source must be defined for connecting to Oracle Advanced Queuing to consume Banner identity messages.

If the Publisher is deployed in the same instance with the other Banner Web Services Adapters, the Publisher can use the same data source that was previously defined for the adapters. This is the recommended installation.























If the Publisher is deployed in a different instance, use the steps in [“Define the data source” on page 18](#) to define the data source for Oracle Advanced Queuing.

Step 4 Define the data source for the bulk load process

Use the following steps to define the data source for accessing the database schema that has access to invoke the bulk load process.

1. Select the **Administration** tab. A list of tasks is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration		
Expand All Collapse All		
Task Name	Go to Task	Description
Administration Tasks		
Properties		
EJB Compiler Settings		Configure the EJB Compiler.
J2EE Websites		Manage the J2EE websites in this OC4J instance.
JSP Properties		Set JSP container properties.
Logger Configuration		Set log levels for all Loggers.
Thread Pool Configuration		Configure the thread pools of this OC4J instance.
Shared Libraries		Manage the shared libraries of this OC4J instance.
Server Properties		Configure server properties for this OC4J instance.
Services		
JDBC Resources		Create/delete/view data sources and connection pools.
Enterprise Messaging Service		
JMS Destinations		Create/delete/edit JMS destinations.
JMS Connection Factories		Configure JMS connection factories.
In-Memory and File Based Persistence		Configure settings for in-memory and file based persistence.
Database Persistence		Configure settings for database persistence.
OracleAS JMS Router		Configure the JMS Router.
JNDI Browser		Browse the JNDI bindings of this OC4J instance.
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.
Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.
JMX		
System MBean Browser		Browse the system MBeans exposed by this OC4J instance.
Notification Subscriptions		View/change subscriptions for notifications for all MBeans.
Notifications Received		View received notifications.

- Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.

JDBC Resources

Application

Data Sources

Create

Name [△]	Application	Attributes				Delete
		JNDI Location	Connection Pool	Managed by OC4J	Test Connection	
"bannerws"	default	jdbc/bannerws	"BannerWS"	✓		
"OracleDS"	default	jdbc/OracleDS	"Example Connection Pool"	✓		
"transsvc"	default	jdbc/transsvc	"Transsvc"	✓		

Connection Pools

Create

Name [△]	Application	Connection Factory Class	Monitor Performance	Test Connection	Refresh Connection Pool	Delete
"BannerWS"	default	oracle.jdbc.pool.OracleDataSource				
"Example Connection Pool"	default	oracle.jdbc.pool.OracleDataSource				
"Transsvc"	default	oracle.jdbc.pool.OracleDataSource				

- Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.

Create Connection Pool - Application

Cancel Continue

Application

Select the application to which this new connection pool is to be added.

Application

Connection Pool Type

New Connection Pool

New Connection Pool from Existing Connection Pool

Create a new connection pool that is configured like an existing connection pool.

Existing Connection Pool

Cancel Continue

- Select the application and connection pool type for the new connection pool:

Application Select *default*. All applications in the instance will use this connection pool.

New Connection Pool Select the button.

- Click **Continue**. The Create Connection Pool page is displayed.

Create Connection Pool

Cancel Back Finish

Home Attributes Proxy Interfaces

* Name

* Connection Factory Class
Class must be available to the application's class loader.

URL

You can either specify a URL directly or have it generated from connection information. When you test a connection, the connection factory class and credentials specified on this page will be used to perform the test.

JDBC URL Test Connection

Generate URL from Connection Information Test Connection

Driver Type

DB Host Name

DB Listener Port

DB Identifier Type

SID/Service Name

TNS Alias

Credentials

TIP For OracleDataSources, credentials must be entered if not already specified in the URL.

Username

Use Cleartext Password
 Password

Use Indirect Password [?](#)
 Indirect Password
example: Scott, customers/Scott

- Enter the following information to set up the connection pool:

Name	<i>baninst1_pool</i> (This is an example. Enter the name of your choice.)
Connection Factory Class	<i>oracle.jdbc.pool.OracleDataSource</i>
JDBC URL	<i>jdbc:oracle:thin:@host:port:SID</i> where <i>host</i> = database host <i>port</i> = database listener port (usually 1521) <i>SID</i> = database instance
Username	<i>baninst1</i>
Use Cleartext Password	Select Use Cleartext Password and enter a password for the <i>baninst1</i> schema.

- Click **Test Connection**. The Test Connection page is displayed.

Test Connection

Enter a SQL statement to use to test the connection. Cancel Test

* SQL Statement Cancel Test

- Click **Test** to test the connection pool. The Create Connection Pool page is displayed with a success or failure message.
 - If the test succeeds, continue with the next step.
 - If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
- Click **Finish**.
- Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

Create Data Source - Application & Type Cancel Continue

Application
Select the application to which this new data source is to be added.
Application

Data Source Type

Managed Data Source
A managed data source is one where OC4J provides critical system infrastructure such as global transaction management, connection pooling, statement caching and error handling.

Native Data Source
A native data source is one that implements the java.sql.DataSource interface and does not make use of OC4J's connection pooling or statement caching capabilities. A native data source can only participate in local transactions.

New Data Source from Existing Data Source
Create a new data source that is configured like an existing data source.
Existing Data Source

Cancel Continue

- Select the application and data source type for the data source:

Application	Select <i>default</i> . All applications in the instance will use this data source.
Managed Data Source	Select the button.

12. Click **Continue**. The Create Data Source - Managed Data Source page is displayed.

Create Data Source - Managed Data Source

Application **default**

Cancel Back Finish

* Name

* JNDI Location

Transaction Level Global & Local Transactions

Connection Pool bansecr

* Login Timeout (seconds) 0

Maximum time to wait while attempting to connect to a database.

13. Enter the following information to set up the data source:

Name	<i>syncBanner</i>
JNDI Location	<i>jdbc/syncbanner</i>
Connection Pool	<i>baninst1_pool</i>

14. Click **Finish**.

Step 5 Define the data source for the Oracle Streams administrator

Use the following steps to define the data source for connecting to the Oracle database for administering Oracle Streams.

1. Select the **Administration** tab. A list of tasks is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration		
Expand All Collapse All		
Task Name	Go to Task	Description
▼ Administration Tasks		
▼ Properties		
EJB Compiler Settings		Configure the EJB Compiler.
J2EE Websites		Manage the J2EE websites in this OC4J instance.
JSP Properties		Set JSP container properties.
Logger Configuration		Set log levels for all Loggers.
Thread Pool Configuration		Configure the thread pools of this OC4J instance.
Shared Libraries		Manage the shared libraries of this OC4J instance.
Server Properties		Configure server properties for this OC4J instance.
▼ Services		
JDBC Resources		Create/delete/view data sources and connection pools.
▼ Enterprise Messaging Service		
JMS Destinations		Create/delete/edit JMS destinations.
JMS Connection Factories		Configure JMS connection factories.
In-Memory and File Based Persistence		Configure settings for in-memory and file based persistence.
Database Persistence		Configure settings for database persistence.
OracleAS JMS Router		Configure the JMS Router.
JNDI Browser		Browse the JNDI bindings of this OC4J instance.
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.
▼ Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.
▼ JMX		
System MBean Browser		Browse the system MBeans exposed by this OC4J instance.
Notification Subscriptions		View/change subscriptions for notifications for all MBeans.
Notifications Received		View received notifications.

2. Select **JDBC Resources** from the Services section. The JDBC Resources page is displayed.

JDBC Resources

Page Refreshed May 10, 2010 1:07:58 PM EDT

Application

Data Sources

Data Sources						
Create						
Name	Application	Attributes				Delete
		JNDI Location	Connection Pool	Managed by OC4J	Test Connection	
"bannerws"	default	jdbc/bannerws	"BannerWS"	✓		
"OracleDS"	default	jdbc/OracleDS	"Example Connection Pool"	✓		
"syncbanner"	default	jdbc/syncbanner	"baninst1"	✓		
"transsvc"	default	jdbc/transsvc	"Transsvc"	✓		

Connection Pools

Connection Pools						
Create						
Name	Application	Connection Factory Class	Monitor Performance	Test Connection	Refresh Connection Pool	Delete
"baninst1"	default	oracle.jdbc.pool.OracleDataSource				
"BannerWS"	default	oracle.jdbc.pool.OracleDataSource				
"Example Connection Pool"	default	oracle.jdbc.pool.OracleDataSource				
"Transsvc"	default	oracle.jdbc.pool.OracleDataSource				

3. Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.

Create Connection Pool - Application

Application

Select the application to which this new connection pool is to be added.

Application

Connection Pool Type

New Connection Pool

New Connection Pool from Existing Connection Pool

Create a new connection pool that is configured like an existing connection pool.

Existing Connection Pool

4. Select the application and connection pool type for the new connection pool:

Application

Select *default*. All applications in the instance will use this connection pool.

New Connection Pool

Select the button.

- Click **Continue**. The Create Connection Pool page is displayed.

Create Connection Pool

Cancel Back Finish

Home **Attributes** Proxy Interfaces

* Name

* Connection Factory Class
Class must be available to the application's class loader.

URL

You can either specify a URL directly or have it generated from connection information. When you test a connection, the connection factory class and credentials specified on this page will be used to perform the test.

JDBC URL

Generate URL from Connection Information

Driver Type

DB Host Name

DB Listener Port

DB Identifier Type

SID/Service Name

TNS Alias

Credentials

TIP For OracleDataSources, credentials must be entered if not already specified in the URL.

Username

Use Cleartext Password
 Password

Use Indirect Password [?](#)
 Indirect Password
example: Scott, customers/Scott

- Enter the following information to set up the connection pool:

Name	<i>streamsadmin_pool</i> (This is an example. Enter the name of your choice.)
Connection Factory Class	<i>oracle.jdbc.pool.OracleDataSource</i>
JDBC URL	<i>jdbc:oracle:thin:@host:port:SID</i> where <i>host</i> = database host <i>port</i> = database listener port (usually 1521) <i>SID</i> = database instance
Username	<i>streamsadmin</i>
Use Cleartext Password	Select Use Cleartext Password and enter a password for the <i>streamsadmin</i> schema.

7. Click **Test Connection**. The Test Connection page is displayed.

Test Connection

Enter a SQL statement to use to test the connection. Cancel Test

* SQL Statement

Cancel Test

8. Click **Test** to test the connection pool. The Create Connection Pool page is redisplayed with a success or failure message.
 - 8.1. If the test succeeds, continue with the next step.
 - 8.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
9. Click **Finish**.
10. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

Create Data Source - Application & Type

Cancel Continue

Application

Select the application to which this new data source is to be added.

Application

Data Source Type

Managed Data Source
A managed data source is one where OC4J provides critical system infrastructure such as global transaction management, connection pooling, statement caching and error handling.

Native Data Source
A native data source is one that implements the java.sql.DataSource interface and does not make use of OC4J's connection pooling or statement caching capabilities. A native data source can only participate in local transactions.

New Data Source from Existing Data Source
Create a new data source that is configured like an existing data source.

Existing Data Source

Cancel Continue

11. Select the application and data source type for the data source:

Application	Select <i>default</i> . All applications in the instance will use this data source.
Managed Data Source	Select the button.

12. Click **Continue**. The Create Data Source - Managed Data Source page is displayed.

Create Data Source - Managed Data Source

Application **default**

Cancel Back Finish

* Name

* JNDI Location

Transaction Level Global & Local Transactions

Connection Pool bansecr

* Login Timeout (seconds) 0

Maximum time to wait while attempting to connect to a database.

13. Enter the following information to set up the data source:

Name	<i>Banner_streamsadmin</i>
JNDI Location	<i>jdbc/streamsadmin</i>
Connection Pool	<i>streamsadmin_pool</i>

14. Click **Finish**.

Step 6 Install the Publisher

Before beginning this step you must understand the concepts published by Oracle regarding the deployment of ear files.

Use the following steps to install the Banner Cardholder Event Publisher to the Oracle Application Server.

1. Connect to the Oracle Enterprise Manager:

`http://<host>:<port>/em`

The console is displayed.


- Click the name of the OC4J instance where the Banner Translation Service and Banner Web Service Adapter for Campus Card Systems are deployed. The Home page for the selected instance is displayed.

OC4J: BWS

View Data Manual Refresh

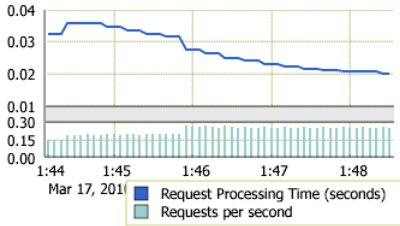
Home Applications Web Services Performance Administration

General Stop Restart



Status **Up**
 Start Time **Mar 17, 2010 4:00:43 AM EDT**
 Version **10.1.3.5.0**
 Oracle Home **D:\product\10.1.3.1\oracleAS_2**
 Host
 Virtual Machines **1**
 Notifications **0**

Response and Load



Request Processing Time (seconds)
Requests per second

- Select the **Applications** tab. A list of deployed applications is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration

This page shows the J2EE applications and application components (EJB Modules, WAR Modules, Resource Adapter Modules) deployed to this OC4J instance.

View Applications

Start Stop Restart Undeploy Redeploy | Deploy

Select All | Select None | Expand All | Collapse All

Select	Name	Status	Start Time	Active Requests	Request Processing Time (seconds)	Active EJB Methods	Application Defined MBeans
<input type="checkbox"/>	▼ All Applications						
<input type="checkbox"/>	ascontrol	↓					
<input type="checkbox"/>	▼ default	↑	Apr 6, 2010 5:18:33 AM EDT	0	0.00	0	
<input type="checkbox"/>	Campuscard	↑	Apr 6, 2010 5:18:44 AM EDT	0	0.00	0	
<input type="checkbox"/>	Housing	↑	Apr 6, 2010 5:18:34 AM EDT	0	0.00	0	
<input type="checkbox"/>	PCI	↑	Apr 6, 2010 5:18:44 AM EDT	0	0.00	0	
<input type="checkbox"/>	TranslationService	↑	Apr 6, 2010 5:18:43 AM EDT	0	0.00	0	
<input type="checkbox"/>	► Middleware Services						

Start Stop Restart Undeploy Redeploy | Deploy

4. Click **Deploy**. The Deploy: Select Archive page is displayed.

Deploy: Select Archive

Cancel Step 1 of 3 Next

Archive

The following types of archives can be deployed: J2EE application (EAR files), Web Modules (WAR files), EJB Modules (EJB JAR files) and Resource Adapter Modules (RAR files).

Archive is present on local host. Upload the archive to the server where Application Server Control is running.

Archive Location Browse...

Archive is already present on the server where Application Server Control is running.

Location on Server

The location on server must be the absolute path or the relative path from j2ee/home

Deployment Plan

The deployment plan is an XML file that contains the deployment settings for an application. If you do not have a deployment plan, one will be created automatically during the deployment process. Later in the deployment process, you can optionally edit the deployment plan and save it for a future deployment of this application.

Automatically create a new deployment plan.

The deployment plan settings will be based on OC4J defaults and information contained in the archive

Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running.

Plan Location Browse...

Deployment plan is already present on server where Application Server Control is running.

Location on Server

The location on server must be the absolute path or the relative path from j2ee/home

Cancel Step 1 of 3 Next

5. Select the file to be uploaded:
 - 5.1. In the Archive section, select **Archive is present on local host. Upload the archive to the server where Application Server Control is running.**
 - 5.2. In the **Archive Location** field, click **Browse** and to the CardholderEventPublisher_v8.1.3.ear file.
 - 5.1. Select the file and click **Open**.
6. Select the deployment plan for the application:
 - 6.1. In the Deployment Plan section, select **Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running.**
 - 6.2. In the **Plan Location** field, click **Browse** and navigate to the CardholderEventPublisher_v8.1.3_plan.dat file.
 - 6.3. Select the file and click **Open**.

- Click **Next** on the Deploy: Select Archive page. The files are uploaded and the Deploy: Application Attributes page is displayed.

Deploy: Application Attributes

Cancel Back Step 2 of 3 Next

Archive Type **J2EE Application (EAR file)**
 Archive Location **\\banner_web_services_8.1.3
 \cardholder_event_publisher\ear\CardholderEventPublisher_v8.1.3.ear**
 Deployment Plan **CardholderEventPublisher_v8.1.3_plan.dat**

* Application Name
 Parent Application default
 Bind Web Module to Site default-web-site
 Context Root

Web Module	Context Root
Banner Cardholder Event Publisher	/cardholderEventPublisher

Cancel Back Step 2 of 3 Next

- Enter a name for the application (for example, *CardholderEventPublisher*) in the **Application Name** field.
- Click **Next**. The Deploy: Deployment Settings page is displayed.

Deploy: Deployment Settings

Cancel Back Step 3 of 3 Deploy

Archive Type **J2EE Application (EAR file)** Application Name **CardholderEventPublisher**
 Archive Location **\\banner_web_services_8.1.3
 \cardholder_event_publisher\ear\CardholderEventPublisher_v8.1.3.ear** Parent Application **default**
 Deployment Plan **CardholderEventPublisher_v8.1.3_plan.dat** Bind Web Module to Site **default-web-site**
 Context Root **/cardholderEventPublisher**

Deployment Tasks
 The table below provides a set of common deployment tasks you might want to perform for this application. Only those tasks that apply to the current application are enabled.

Task Name	Go To Task	Description
Map Environment References		Map any environment references in your application (for example, data sources) to physical entities currently present on the operational environment.
Select Security Provider		A security provider acts as the source for available users and groups when mapping security roles.
Map Security Roles		Map any security roles exposed by your application to existing users and groups. The list of users and groups is obtained from the security provider you selected for this application.
Configure EJBs		Configure the Enterprise JavaBeans in your application.
Configure Clustering		Configure clustering of your application.
Configure Class Loading		Manipulate the classpath of your application.

Advanced Deployment Plan Editing
 Click Edit Deployment Plan to set more advanced deployment options. Edit Deployment Plan

Save Deployment Plan
 After you make changes, you can save the deployment plan to your local disk. You can then use the saved deployment plan to redeploy this application later. Save Deployment Plan

Cancel Back Step 3 of 3 Deploy

- Click **Deploy** to accept the values and install the Publisher. A deployment confirmation page is displayed.
- Click **Return** to continue. The **Applications** tab is displayed with the deployed Publisher.

Step 7 Verify the instance configuration

The instance where the Banner Cardholder Event Publisher is deployed must be properly configured. This step is particularly important if you are deploying the Publisher into an instance where prior versions of the Banner Web Services Adapter were deployed.

Use the following steps to verify and, if necessary, change the established value for the class loader child-parent delegation model parameter.

1. Access the server that hosts the Oracle Application Server. This can be done locally or via a remote desktop or telnet connection.
2. With a text or XML editor, open the `global-web-application.xml` file that resides in the `<IAS_HOME>/j2ee/<OC4J instance>/config` subdirectory.
3. Check to see if there is a `<web-app-class-loader>` XML tag under the `<orion-web-app>` XML tag.
 - 3.1. If the file *does not* contain this tag, exit the editor and the server. The configuration is acceptable.
 - 3.2. If the file *does* contain this tag, continue with step 4.
4. Check the value of the `search-local-classes-first` attribute.
 - 4.1. If the value is `"false,"` exit the editor and the server. The configuration is acceptable.
 - 4.2. If the value is `"true,"` continue with step 5.
5. Set the value to `"false"` The tag should look as follows:

```
<web-app-class-loader search-local-classes-first="false"/>
```

6. Save the file.
7. Exit the editor and the server.
8. Reconnect to the Oracle Enterprise Manager and restart the OC4J container.

Step 8 Configure the security role and user

Before beginning this step, refer to the security configuration for your version of the Oracle Application Server.

Use the following steps to add the `chepadmin` role and an administrative user to the Publisher. This role and user are required for accessing the Banner Cardholder Event Publisher.

1. Select the **Administration** tab. A list of tasks is displayed.

OC4J: BWS

Home Applications Web Services Performance Administration		
Expand All Collapse All		
Task Name	Go to Task	Description
▼ Administration Tasks		
▼ Properties		
EJB Compiler Settings		Configure the EJB Compiler.
J2EE Websites		Manage the J2EE websites in this OC4J instance.
JSP Properties		Set JSP container properties.
Logger Configuration		Set log levels for all Loggers.
Thread Pool Configuration		Configure the thread pools of this OC4J instance.
Shared Libraries		Manage the shared libraries of this OC4J instance.
Server Properties		Configure server properties for this OC4J instance.
▼ Services		
JDBC Resources		Create/delete/view data sources and connection pools.
▼ Enterprise Messaging Service		
JMS Destinations		Create/delete/edit JMS destinations.
JMS Connection Factories		Configure JMS connection factories.
In-Memory and File Based Persistence		Configure settings for in-memory and file based persistence.
Database Persistence		Configure settings for database persistence.
OracleAS JMS Router		Configure the JMS Router.
JNDI Browser		Browse the JNDI bindings of this OC4J instance.
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.
▼ Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.
▼ JMX		
System MBean Browser		Browse the system MBeans exposed by this OC4J instance.
Notification Subscriptions		View/change subscriptions for notifications for all MBeans.
Notifications Received		View received notifications.

2. Select **Security Providers** from the Security section. The Security Providers page is displayed.

Security Providers

Instance Level Security

You can configure the security attributes (realms, users & roles) for all applications deployed to this OC4J instance by clicking on the button below.

[Instance Level Security](#)

Application Server Control Security

You can configure the security provider, users & roles for the Application Server Control management application by clicking on the button below or by using the global Setup link.

[Application Server Control Security](#)

Application Level Security

The table lists applications currently deployed to this OC4J instance and the security provider in use by each application. You can edit the properties of the security provider specified for a given application by clicking on the Edit icon.

[Expand All](#) | [Collapse All](#)

Application Name	Security Provider	Edit
▼ default		
PCT	File-Based Security Provider	
BSR	File-Based Security Provider	
TranslationService	File-Based Security Provider	
AdvancementServices	File-Based Security Provider	
javasso	File-Based Security Provider	
Campuscard	File-Based Security Provider	
Housing	File-Based Security Provider	
datatags	File-Based Security Provider	
CardholderEventPublisher	File-Based Security Provider	

3. In the Application Level Security section, click the **Edit** button for the Publisher application. The Security Provider page is displayed.
4. Select the **Realms** tab.

Security Provider

Security Provider Type **File-Based Security Provider** [Change Security Provider](#)

Security Provider Attributes: File-Based Security Provider

[General](#) [Realms](#)

Search

Name [Go](#)

Results

[Create](#)

Realm Name ▲	Roles	Users	Delete
jazn.com	9	6	

- Click the link under the **Roles** column. The Roles page is displayed.

Roles

Security Provider Type **File-Based Security Provider**
Realm Name **jazn.com**

Search

Name

Results

Role Name [△]	Users	Delete
ascontrol_admin	1	
ascontrol_appadmin	0	
ascontrol_monitor	1	

- Click **Create**. The Add Role page is displayed.

Add Role

Realm Name **jazn.com**

* Name

Grant RMI Login Permission

Grant Administration Permission

Assign Roles

A role may inherit from other roles. Select the roles you would like this role to inherit.

Available Roles		Selected Roles
ascontrol_admin ascontrol_appadmin ascontrol_monitor	 Move Move All Remove Remove All	

- Enter *chepadmin* in the **Name** field.
- Click **OK**. The Roles page is redisplayed with the new role.

9. Return to the Security Provider page.

Security Provider

Security Provider Type **File-Based Security Provider** [Change Security Provider](#)

Security Provider Attributes: File-Based Security Provider

[General](#) [Realms](#)

Search
Name [Go](#)

Results
[Create](#)

Realm Name [△]	Roles	Users	Delete
jazn.com	9	6	

[General](#) [Realms](#)

10. Click the link under the **Users** column. The Users page is displayed.

Users

Security Provider Type **File-Based Security Provider**

Realm Name **jazn.com**

Search

Name [Go](#)

Results

[Create](#)

User Name [△]	Assigned Roles	Delete
anonymous		
JtaAdmin	oc4j-administrators*	
oc4jadmin	oc4j-administrators*, ascontrol_admin*	
rmiuser	ascontrol_monitor*	

11. Click **Create**. The Add User page is displayed.

Add User

[Cancel](#) [OK](#)

Realm Name **jazn.com**

* Name

* Password

* Confirm Password

Assign Roles

Available Roles		Selected Roles
<input type="text" value="ascontrol_admin"/> <input type="text" value="ascontrol_appadmin"/> <input type="text" value="ascontrol_monitor"/> <input type="text" value="chepadmin"/>	Move Move All Remove Remove All	<input type="text"/>

[Cancel](#) [OK](#)

12. Enter the following information to create a user:

Name	<i>Admin</i> (This is an example. Enter the name of your choice.)
Password	Password for the user being created
Confirm Password	Confirmation of the password

13. In the Assign Roles section, select the *chepadmin* role in the **Available Roles** list and move it to the **Selected Roles** list.

14. Click **OK**. The Users page is redisplayed with the new user.

Installation on Oracle WebLogic Server 11g

The Banner Cardholder Event Publisher is packaged as a J2EE compatible enterprise archive file named `CardholderEventPublisher_v8.1.3.ear`. Use the following steps to install the Publisher on Oracle WebLogic Server 11g:

- [Step 1. “Verify the capture process rules”](#).
- [Step 2. “Create, configure, and start the Oracle Streams processes”](#).
- [Step 3. “Configure the Oracle WebLogic Server”](#)
- [Step 4. “Define the data source for Oracle Advanced Queuing”](#)
- [Step 5. “Define the data source for the bulk load process”](#)
- [Step 6. “Define the data source for the Oracle Streams administrator”](#)
- [Step 7. “Install the Publisher”](#)
- [Step 8. “Configure the security group and user”](#)

A new Managed server was created when the Banner Translation Service was installed. The Publisher must be deployed in this same Managed Server with the Banner Translation Service and the Banner Web Service Adapter for Campus Card Systems.

Step 1 Verify the capture process rules

The capture rules for cardholder data should already be loaded in the database. Capture rules are initially provided in Banner General seed data scripts, as described in [“Banner dependency” on page 94](#). Additional capture rules are provided with Banner Web Services download in the following directory:

```
\cardholder_event_publisher\scripts\capture_rules
```

Use the steps in Appendix B, “Using Oracle Streams,” of the *Banner Web Services Handbook* to verify that the rules are loaded in the database.

Step 2 Create, configure, and start the Oracle Streams processes

Use the steps in Appendix B, “Using Oracle Streams,” of the *Banner Web Services Handbook* to create, configure, and start the Oracle Streams capture and apply processes. These steps accomplish the following:

- Create buffered queues and queue tables to manage events.
- Create supplemental, primary key, and unique key log groups for the configured tables.
- Configure the DML callback handler for the apply process.
- Set the instantiation SCN for the tables in the apply process.

Note

These steps should be performed as the `streamsadmin` Oracle user only. ■

Refer to the *Banner Web Services Handbook* for more information about using Oracle Streams.

Step 3 Configure the Oracle WebLogic Server

The Oracle WebLogic Server requires the following configuration changes:

- The server must be configured to use the *Advanced* security model instead of the default *DD only* option.

Note

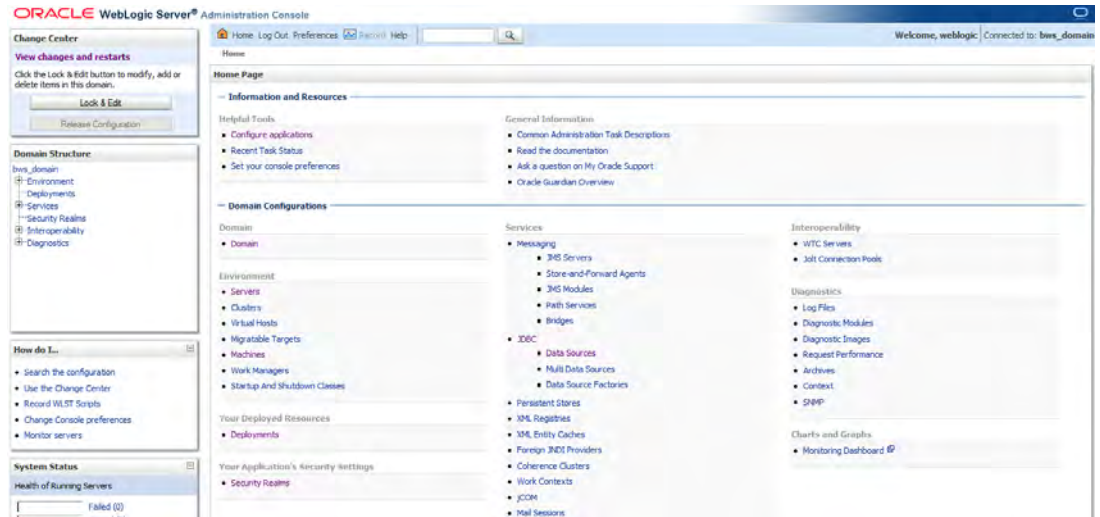
The Oracle WebLogic Server needs to be configured only once. If the server was previously configured, you can skip this step. ■

Use the following steps to configure the server.

1. Connect to the Oracle WebLogic Server Administration Console:

`http://<host>:<port>/console`

The Home Page is displayed.



2. In the Domain Structure pane, click **Security Realms**.



The Summary of Security Realms page is displayed.

Summary of Security Realms

A security realm is a container for the mechanisms—including users, groups, security roles, security policies, and security providers—that are used to protect WebLogic resources. You can have multiple security realms in a WebLogic Server domain, but only one can be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebLogic Server domain. Click the name of the realm to explore and configure that realm.

[Customize this table](#)

Realms(Filtered - More Columns Exist)

<input type="checkbox"/>	Name ↕	Default Realm
<input type="checkbox"/>	myrealm	true

3. Click **myrealm**. The Settings for myrealm page is displayed.

Settings for myrealm

Configuration | Users and Groups | Roles and Policies | Credential Mappings | Providers | Migration

General | RDBMS Security Store | User Lockout | Performance

Use this page to configure the general behavior of this security realm.

Note:
If you are implementing security using JACC (Java Authorization Contract for Containers as defined in JSR 115), you must use the DD Only security model. Other WebLogic Server models are not available and the security functions for Web applications and EJBs in the Administration Console are disabled.

Name: myrealm The name of this security realm. [More Info...](#)

Security Model Default: Advanced Specifies the default security model for Web applications or EJBs that are secured by this security realm. You can override this default during deployment. [More Info...](#)

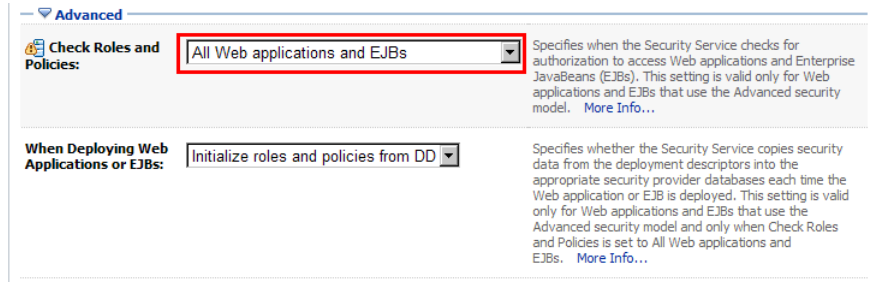
Combined Role Mapping Enabled Determines how the role mappings in the Enterprise Application, Web application, and EJB containers interact. This setting is valid only for Web applications and EJBs that use the Advanced security model and that initialize roles from deployment descriptors. [More Info...](#)

Use Authorization Providers to Protect JMX Access Configures the WebLogic Server MBean servers to use the security realm's Authorization providers to determine whether a JMX client has permission to access an MBean attribute or invoke an MBean operation. [More Info...](#)

[Advanced](#)

4. Select *Advanced* in the **Security Model Default** drop-down list.

5. Click the **Advanced** link to display the advanced options.



6. Select *All Web Applications and EJBs* in the **Check Roles and Policies** drop-down list.
7. Click **Save**.
8. Restart the server for the changes to take effect.

Step 4 Define the data source for Oracle Advanced Queuing

A data source provides the connection properties to the Banner database. A data source must be defined for connecting to Oracle Advanced Queuing to consume Banner identity messages.

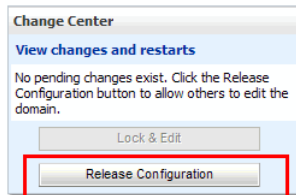
If the Publisher will be deployed in the same instance with the other Banner Web Services Adapters, the Publisher can use the same data source that was previously defined for the adapters. This is the recommended installation.

If the Publisher will be deployed in a different instance, use the steps in [“Define the data source” on page 36](#) to define the data source for Oracle Advanced Queuing.

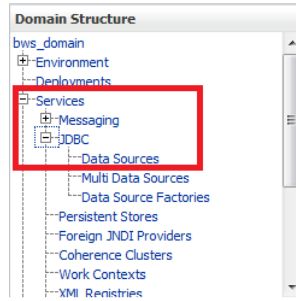
Step 5 Define the data source for the bulk load process

Use the following steps to define the data source for accessing the database schema that has access to invoke the bulk load process.

1. In the Change Center pane, click **Release Configuration**.



- In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**.



The Summary of JDBC Data Sources page is displayed.

- In the Change Center pane, click **Lock & Edit**.
- On the Summary of JDBC Data Sources page, click **New**. The Create a New JDBC Data Source page is displayed.

The image shows a web form titled 'Create a New JDBC Data Source'. It includes navigation buttons (Back, Next, Finish, Cancel) and a section for 'JDBC Data Source Properties'. The form contains the following fields:

- Name:** A text input field containing 'syncBanner'.
- JNDI Name:** A large empty text area.
- Database Type:** A dropdown menu with 'Oracle' selected.
- Database Driver:** A dropdown menu with '*Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11' selected.

- Enter the following data source properties:

Name	<i>syncBanner</i>
JNDI Name	<i>jdbc/syncbanner</i>
Database Type	<i>Oracle</i>
Database Driver	Appropriate database driver that is used to create database connections <i>Select Oracle's Driver (Thin) for Instance connections; Versions:9.0.1, 9.2.0,10,11</i>

6. Click **Next**. The next page is displayed.

The screenshot shows the 'Create a New JDBC Data Source' dialog box, specifically the 'Transaction Options' page. At the top, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'. Below this, the text reads: 'You have selected non-XA JDBC driver to create database connection in your new data source.' The main question is 'Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.' There are three radio button options: 'Supports Global Transactions' (which is highlighted with a red rectangle and is currently unselected), 'Logging Last Resource' (selected), and 'Emulate Two-Phase Commit' (unselected). Below these are two more radio button options: 'One-Phase Commit' (selected). At the bottom, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

7. Clear the **Supports Global Transactions** check box.

8. Click **Next**. The next page is displayed.

The screenshot shows the 'Create a New JDBC Data Source' dialog box, specifically the 'Connection Properties' page. At the top, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'. Below this, the text reads: 'Define Connection Properties.' The main question is 'What is the name of the database you would like to connect to?' The 'Database Name' field contains 'smpl'. The next question is 'What is the name or IP address of the database server?' The 'Host Name' field contains 'm088042'. The next question is 'What is the port on the database server used to connect to the database?' The 'Port' field contains '1521'. The next question is 'What database account user name do you want to use to create database connections?' The 'Database User Name' field contains 'baninst1'. The next question is 'What is the database account password to use to create database connections?' The 'Password' and 'Confirm Password' fields are filled with dots. At the bottom, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

9. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	<i>baninst1</i>
Password	Password for the <code>baninst1</code> user
Confirm Password	Confirmation of the password

10. Click **Next**. The next page is displayed with the properties that you entered.

The screenshot shows the 'Create a New JDBC Data Source' wizard, specifically the 'Test Database Connection' step. The window title is 'Create a New JDBC Data Source'. At the top, there are navigation buttons: 'Test Configuration', 'Back', 'Next', 'Finish', and 'Cancel'. The main content area is titled 'Test Database Connection' and contains the following sections:

- Test Database Connection:** A heading followed by the text 'Test the database availability and the connection properties you provided.'
- Question:** 'What is the full package name of JDBC driver class used to create database connections in the connection pool?' (Note that this driver class must be in the classpath of any server to which it is deployed.)
- Driver Class Name:** A text input field containing 'oracle.jdbc.OracleDriver'.
- Question:** 'What is the URL of the database to connect to? The format of the URL varies by JDBC driver.'
- URL:** A text input field containing 'jdbc:oracle:thin:@m08804'.
- Question:** 'What database account user name do you want to use to create database connections?'
- Database User Name:** A text input field containing 'baninst1'.
- Question:** 'What is the database account password to use to create database connections?' (Note: for secure password management, enter the password in the Password field instead of the Properties field below)
- Password:** A password input field with 10 dots.
- Confirm Password:** A password input field with 10 dots.
- Question:** 'What are the properties to pass to the JDBC driver when creating database connections?'
- Properties:** A text area containing 'user=baninst1'.
- Question:** 'What table name or SQL statement would you like to use to test database connections?'
- Test Table Name:** A text area containing 'SQL SELECT 1 FROM DUAL'.

At the bottom of the window, there are navigation buttons: 'Test Configuration', 'Back', 'Next', 'Finish', and 'Cancel'.

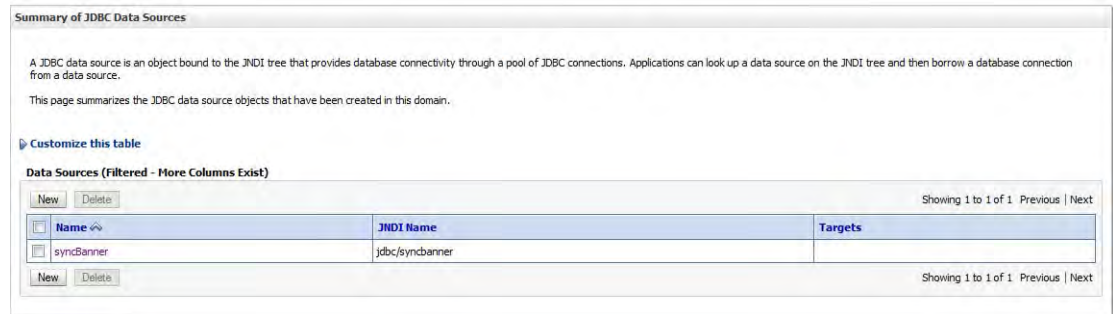
11. Verify the property values.

12. Click **Test Configuration**. The page is redisplayed with a success or failure message.

12.1. If the test succeeds, continue with the next step.

12.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

- Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.

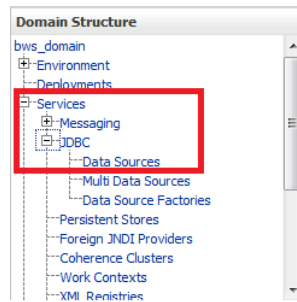


- In the Change Center pane, click **Activate Changes**.
- On the Summary of JDBC Data Sources page, click the name of the new data source. The Settings for syncBanner page is displayed.
- Select the Targets tab.



- In the Change Center pane, click **Lock & Edit**.
- On the Settings for syncBanner page, select the server where the data source should be deployed.
- Click **Save**.
- In the Change Center pane, click **Activate Changes**.

21. In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**.



The Summary of JDBC Data Sources page is displayed.

22. Verify that the new data source is associated with the server.

Step 6 Define the data source for the Oracle Streams administrator

Use the following steps to define the data source for connecting to the Oracle database for administering Oracle Streams.

1. In the Change Center pane, click **Lock & Edit**.
2. Ensure that the Summary of JDBC Data Sources page is displayed. (If it is not displayed, expand and click **Services -> JDBC -> Data Sources** in the Domain Structure pane.)
3. Click **New** on the Summary of JDBC Data Sources page. The Create a New JDBC Data Source page is displayed.
4. Enter the following data source properties:

Name	<i>Banner_streamsadmin</i>
JNDI Name	<i>jdbc/streamsadmin</i>
Database Type	<i>Oracle</i>
Database Driver	Appropriate database driver that is used to create database connections

5. Click **Next**. The next page is displayed.
6. Clear the **Supports Global Transactions** check box.
7. Click **Next**. The next page is displayed.

8. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	<i>streamsadmin</i>
Password	Password for the <code>streamsadmin</code> user
Confirm Password	Confirmation of the password

9. Click **Next**. The next page is displayed with the properties that you entered.
10. Verify the property values.
11. Click **Test Configuration**. The page is redisplayed with a success or failure message.
 - 11.1. If the test succeeds, continue with the next step.
 - 11.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
12. Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.
13. In the Change Center pane, click **Activate Changes**.
14. On the Summary of JDBC Data Sources page, click the name of the new data source. The Settings for `streamsadmin` page is displayed.
15. Select the Targets tab.
16. In the Change Center pane, click **Lock & Edit**.
17. On the Settings for `streamsadmin` page, select the server where the data source should be deployed.
18. Click **Save**.
19. In the Change Center pane, click **Activate Changes**.

20. In the Domain Structure pane, click **Services -> JDBC -> Data Sources**. The Summary of JDBC Data Sources page is displayed.

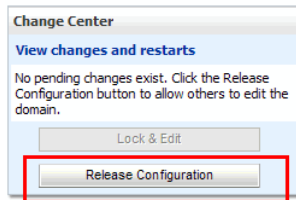
21. Verify that the new data source is associated with the server.

Step 7 Install the Publisher

Before beginning this step you must understand the concepts published by Oracle regarding the deployment of ear files.

Use the following steps to install the Banner Cardholder Event Publisher to the Oracle WebLogic Server.

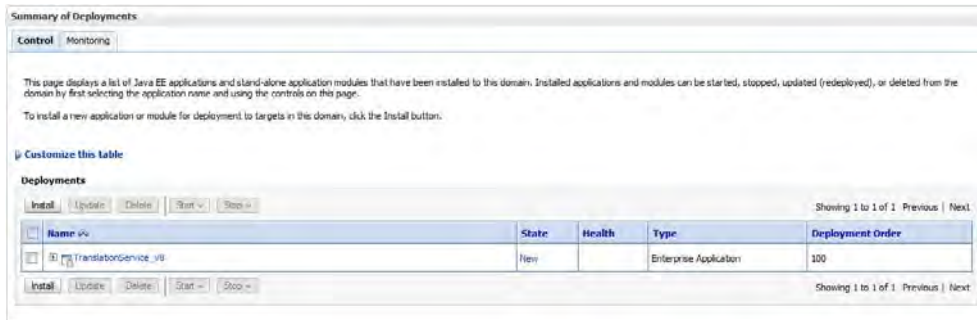
1. In the Change Center pane, click **Release Configuration**.



2. In the Domain Structure pane, click **Deployments**.



The Summary of Deployments page is displayed.



3. In the Change Center pane, click **Lock & Edit**.

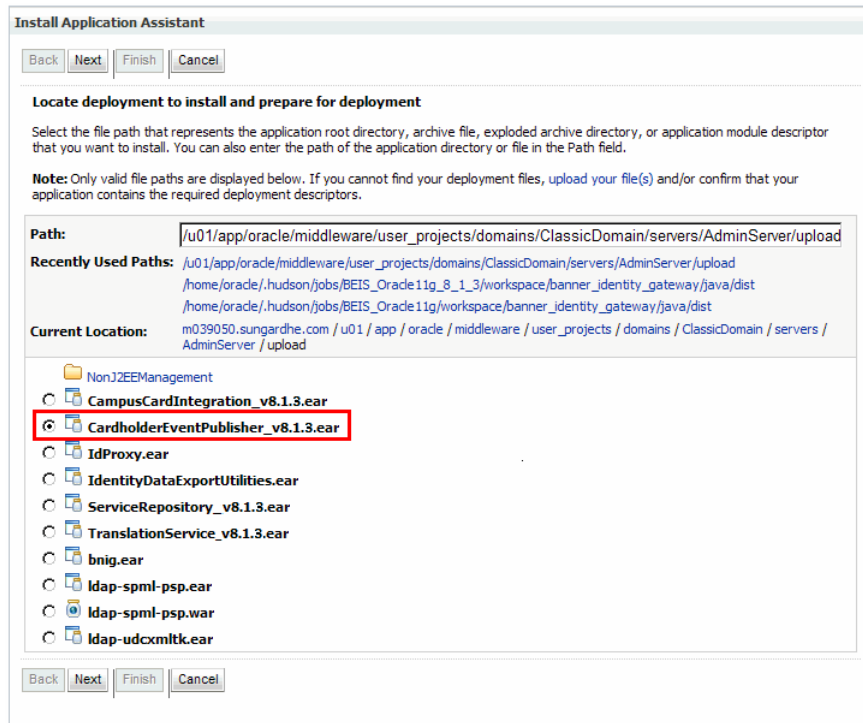
4. In the Summary of Deployments page, click **Install**. The Install Application Assistant page is displayed.

The screenshot shows the 'Install Application Assistant' dialog box. At the top, there are navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'. The main heading is 'Locate deployment to install and prepare for deployment'. Below this, there is a text box with instructions: 'Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. You can also enter the path of the application directory or file in the Path field.' A note follows: 'Note: Only valid file paths are displayed below. If you cannot find your deployment file, upload your file(s) and/or confirm that your application contains the required deployment descriptors.' The 'Path:' field contains the text: '/home/oracle/.hudson/jobs/BEIS_Oracle11g_8_1_3/workspace/banner_identity_gateway/java'. Below the path field, there are sections for 'Recently Used Paths:' and 'Current Location:'. The 'Recently Used Paths:' section lists several paths, including '/home/oracle/.hudson/jobs/BEIS_Oracle11g_8_1_3/workspace/banner_identity_gateway/java/dist' and '/u01/app/oracle/middleware/user_projects/domains/ClassicDomain/servers/AdminServer/upload'. The 'Current Location:' section shows a tree view with folders 'ejb' and 'web'. At the bottom, there are navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

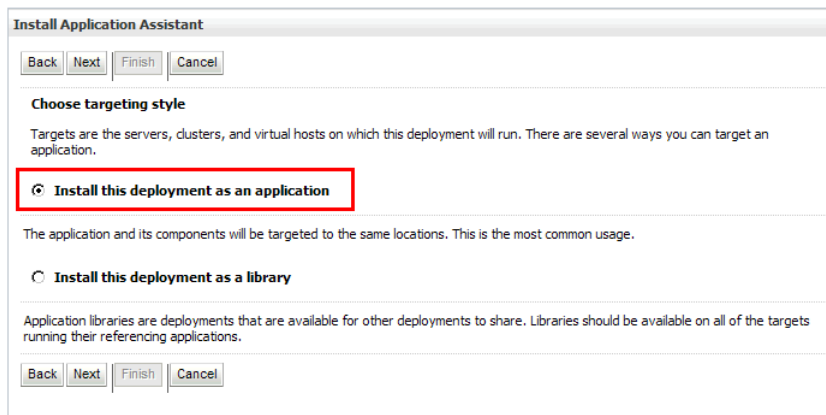
5. Click **upload your file(s)**. The next installation page is displayed.

The screenshot shows the 'Install Application Assistant' dialog box. At the top, there are navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'. The main heading is 'Upload a Deployment to the admin server'. Below this, there is a text box with instructions: 'Click the Browse button below to select an application or module on the machine from which you are currently browsing. When you have located the file, click the Next button to upload this deployment to the Administration Server.' There are two sections: 'Deployment Archive:' and 'Deployment Plan Archive:'. Each section has a text input field and a 'Browse...' button. At the bottom, there are navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

6. Select the file to be uploaded:
 - 6.1. In the **Deployment Archive** field, click **Browse** and navigate to the `CardholderEventPublisher_v8.1.3.ear` file.
 - 6.1. Select the file and click **Open**.
7. Click **Next**. The next installation page is displayed.



8. Select the `CardholderEventPublisher_v8.1.3.ear` file from the list.
9. Click **Next**. The next installation page is displayed.



10. Select **Install this deployment as an application**.

11. Click **Next**. The next installation page is displayed.

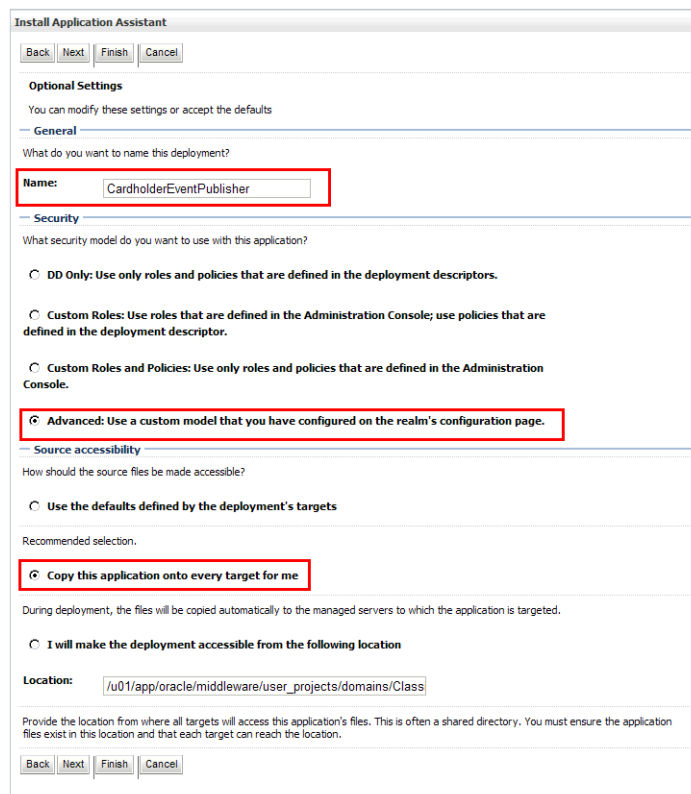
11.1. In some domains, the following page is displayed. Select the server where the Publisher should be deployed and go to step 12.

The Publisher must be installed in the instance where the Banner Translation Service and the Banner Web Service Adapter for Campus Card Systems are installed.

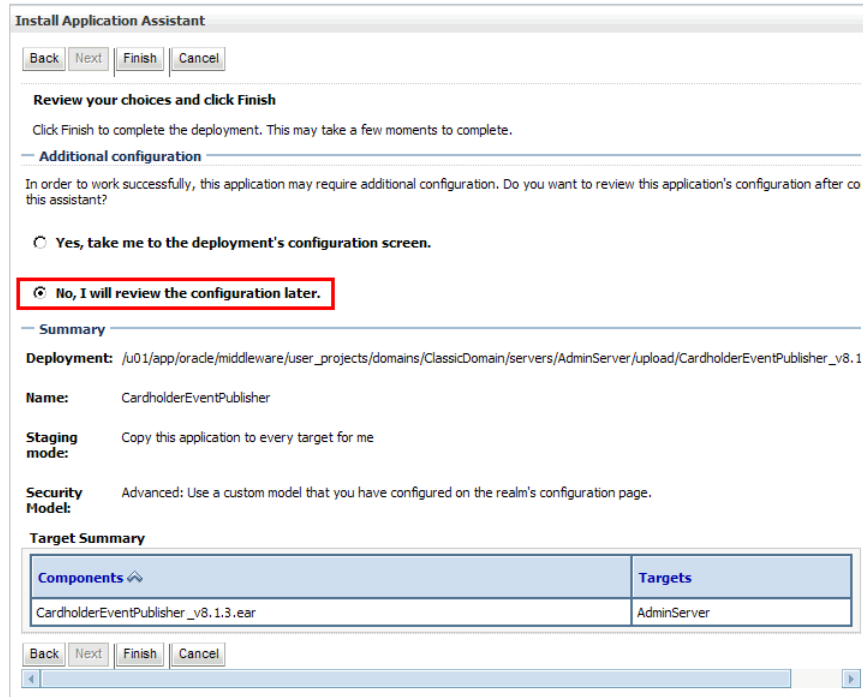


11.2. In some domains, the preceding page is skipped. Go directly to step 12.

12. Click **Next**. The next installation page is displayed.



13. Enter a name for the application (for example, *CardholderEventPublisher*) in the **Name** field.
14. Select **Advanced: Use a custom model that you have configured on the realm's configuration page**.
15. Select **Copy this application onto every target for me**.
16. Click **Next**. The next installation page is displayed.



17. Select **No, I will review the configuration later**.

18. Click **Finish** to start the deployment. When deployment is completed, the Summary of Deployments page is redisplayed with the newly deployed Publisher.

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Customize this table

Deployments

Install Update Delete Start Stop Previous Next

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input type="checkbox"/>	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	CardholderEventPublisher	distribute Initializing		Enterprise Application	100
<input type="checkbox"/>	DMS Application (11.1.1.1.0)	Active	OK	Web Application	5

19. In the Change Center pane, click **Activate Changes**.

20. Start the newly deployed application as follows:

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Customize this table

Deployments

Install Update Delete Start Stop Previous Next

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input type="checkbox"/>	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
<input checked="" type="checkbox"/>	CardholderEventPublisher	distribute Initializing		Enterprise Application	100
<input type="checkbox"/>	DMS Application (11.1.1.1.0)	Active	OK	Web Application	5

20.1. Select the newly deployed Publisher.

20.1. Click **Start** -> **Servicing all requests**. The Start Application Assistant page is displayed.

Start Application Assistant

Yes No

Start Deployments

You have selected the following deployments to be started. Click 'Yes' to continue, or 'No' to cancel.

- CardholderEventPublisher

Yes No

20.2. Click **Yes**.

Step 8 Configure the security group and user

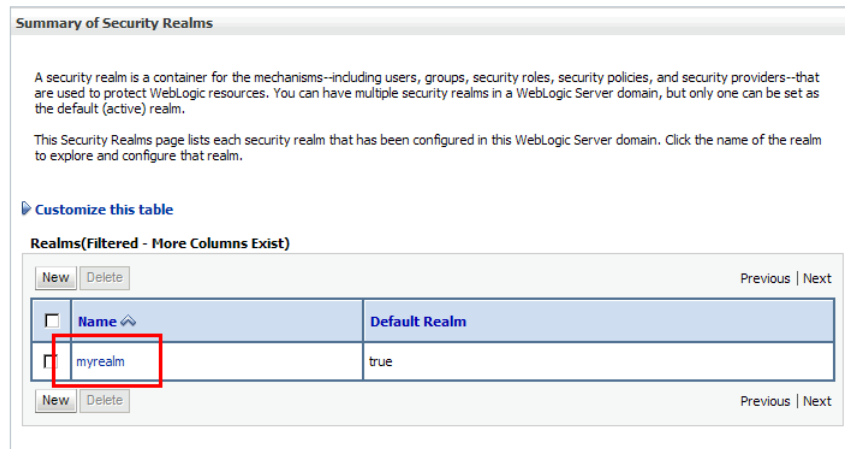
Before beginning this step, refer to the security configuration for your version of the Oracle WebLogic Server.

Use the following steps to add the `chepAdminGroup` group and an administrative user to the Banner Cardholder Event Publisher application. This group and user are required for accessing the Banner Cardholder Event Publisher administrative interface.

1. In the Domain Structure pane, click **Security Realms**.



The Summary of Security Realms page is displayed.



2. Click **myrealm**. The Settings for myrealm page is displayed.
3. Select the **Users and Groups** tab.

4. Select the **Groups** sub-tab. A table of existing groups is displayed.

The screenshot shows the 'Settings for myrealm' interface. The 'Users and Groups' sub-tab is selected and highlighted with a red box. Below the sub-tab, there is a 'Groups' section with a table of existing groups. The table has columns for Name, Description, and Provider. The groups listed are AdminChannelUsers, Administrators, AppTesters, bannerws, DemoGroup, Deployers, idpadmin, and Monitors. Each group has a checkbox in the first column and a 'Provider' column with the value 'DefaultAuthenticator'.

<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	AdminChannelUsers	AdminChannelUsers can access the admin channel.	DefaultAuthenticator
<input type="checkbox"/>	Administrators	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
<input type="checkbox"/>	AppTesters	AppTesters group.	DefaultAuthenticator
<input type="checkbox"/>	bannerws	bannerws	DefaultAuthenticator
<input type="checkbox"/>	DemoGroup	Demo group created for demo purpose	DefaultAuthenticator
<input type="checkbox"/>	Deployers	Deployers can view all resource attributes and deploy applications.	DefaultAuthenticator
<input type="checkbox"/>	idpadmin	Enterprise Identity Proxy Services Group	DefaultAuthenticator
<input type="checkbox"/>	Monitors	Monitors can view and modify all resource attributes and perform operations not restricted by roles.	DefaultAuthenticator

5. Click **New**. The Create a New Group page is displayed.

The screenshot shows the 'Create a New Group' page. It has a form with the following fields:

- Name:** * Name: chepAdminGroup
- Description:** Description: Banner Cardholder Event Publisher Administrative Group
- Provider:** Provider: DefaultAuthenticator (dropdown menu)

There are 'OK' and 'Cancel' buttons at the top and bottom of the form.

6. Enter the following information to create a group:

Name *chepAdminGroup*
Description *Banner Cardholder Event Publisher Administrative Group*
Provider *DefaultAuthenticator*

7. Click **OK**. The table of groups is redisplayed with the new group.

The screenshot shows the 'Settings for myrealm' interface with the 'Users and Groups' tab selected. Under the 'Groups' sub-tab, there is a table of configured groups. The 'chepAdminGroup' entry is highlighted with a red border. The table has three columns: Name, Description, and Provider. The 'chepAdminGroup' row shows the name 'chepAdminGroup', the description 'Banner Cardholder Event Publisher Administrative Group', and the provider 'DefaultAuthenticator'.

<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	AdminChannelUsers	AdminChannelUsers can access the admin channel.	DefaultAuthenticator
<input type="checkbox"/>	Administrators	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
<input type="checkbox"/>	AppTesters	AppTesters group.	DefaultAuthenticator
<input type="checkbox"/>	bannerwsGroup	Banner Web Services Administrative Group	DefaultAuthenticator
<input type="checkbox"/>	bnigAdminGroup	Banner Identity Gateway Administrative Group	DefaultAuthenticator
<input type="checkbox"/>	chepAdminGroup	Banner Cardholder Event Publisher Administrative Group	DefaultAuthenticator
<input type="checkbox"/>	CrossDomainConnectors	CrossDomainConnectors can make inter-domain calls from foreign domains.	DefaultAuthenticator
<input type="checkbox"/>	DemoGroup	Demo group created for demo purpose	DefaultAuthenticator
<input type="checkbox"/>	Deployers	Deployers can view all resource attributes and deploy applications.	DefaultAuthenticator
<input type="checkbox"/>	idpadmin	Enterprise Identity Proxy Services Group	DefaultAuthenticator

8. Select the **Users** sub-tab. A table of existing users is displayed.

The screenshot shows the 'Settings for myrealm' interface. The 'Users and Groups' tab is selected, and the 'Users' sub-tab is highlighted with a red box. Below the sub-tabs, there is a description: 'This page displays information about each user that has been configured in this security realm.' A link 'Customize this table' is present. The main content area is titled 'Users' and contains a table with columns for 'Name', 'Description', and 'Provider'. The table lists several users, including 'bannerwsUser', 'idproxy', 'OracleSystemUser', 'transsvc', and 'weblogic'. Each user has a checkbox in the 'Name' column. Navigation buttons 'New' and 'Delete' are located at the top and bottom of the table area.

<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	bannerwsUser		DefaultAuthenticator
<input type="checkbox"/>	idproxy	Enterprise Identity Proxy Services User	DefaultAuthenticator
<input type="checkbox"/>	OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
<input type="checkbox"/>	transsvc		DefaultAuthenticator
<input type="checkbox"/>	weblogic		DefaultAuthenticator

9. Click **New**. The Create a New User page is displayed.

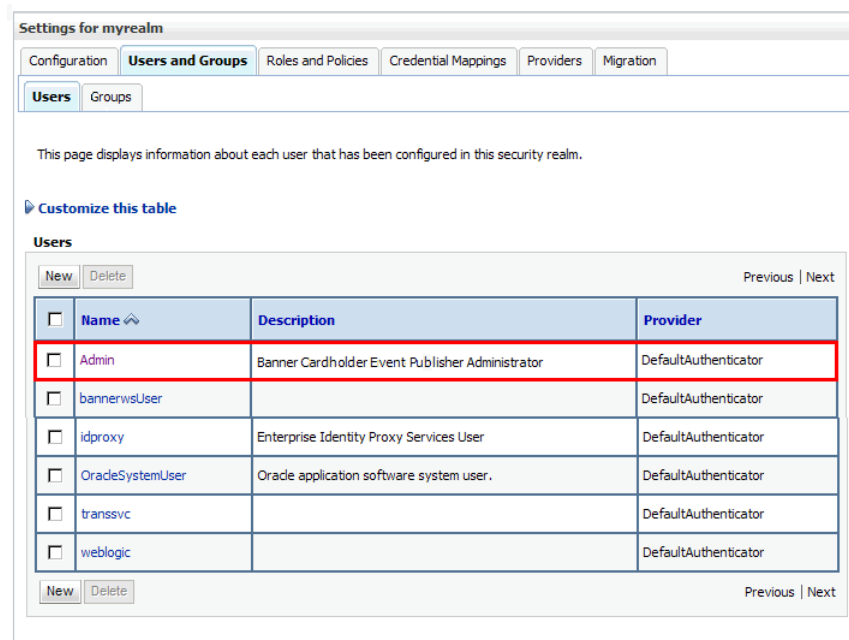
The screenshot shows the 'Create a New User' form. It includes 'OK' and 'Cancel' buttons at the top and bottom. The form is titled 'User Properties' and contains the following fields:

- Name:** A text input field containing 'Admin'.
- Description:** A text input field containing 'Banner Cardholder Event Publisher Administrator'.
- Provider:** A dropdown menu with 'DefaultAuthenticator' selected.
- Password:** A password input field with 10 dots.
- Confirm Password:** A password input field with 10 dots.

10. Enter the following information to create a user:

Name	<i>Admin</i> (This is an example. Enter the name of your choice.)
Description	<i>Banner Cardholder Event Publisher Administrator</i>
Provider	<i>DefaultAuthenticator</i>
Password	Password used to log in to the Banner Cardholder Event Publisher administrative interface
Confirm Password	Confirmation of the password

11. Click **OK**. The table of users is redisplayed with the new user.

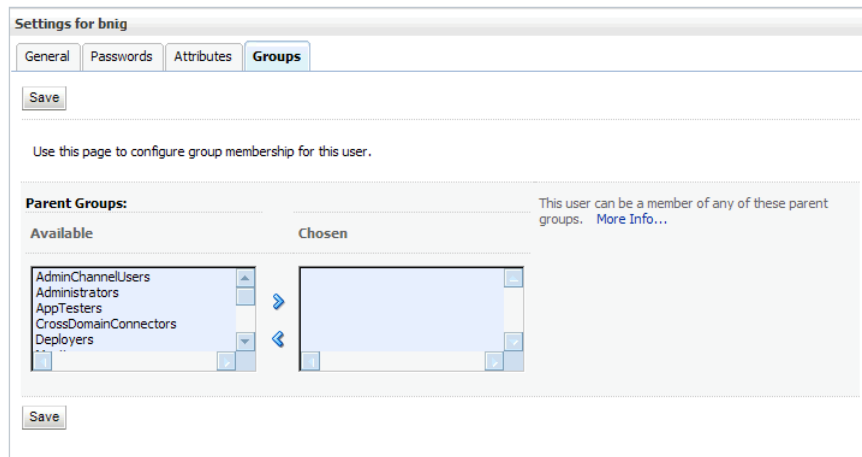


The screenshot shows the 'Settings for myrealm' interface with the 'Users and Groups' tab selected. Under the 'Users' sub-tab, there is a table of users. The 'Admin' user is highlighted with a red border. The table has columns for Name, Description, and Provider. There are 'New' and 'Delete' buttons at the top and bottom of the table, and 'Previous' and 'Next' links on the right side.

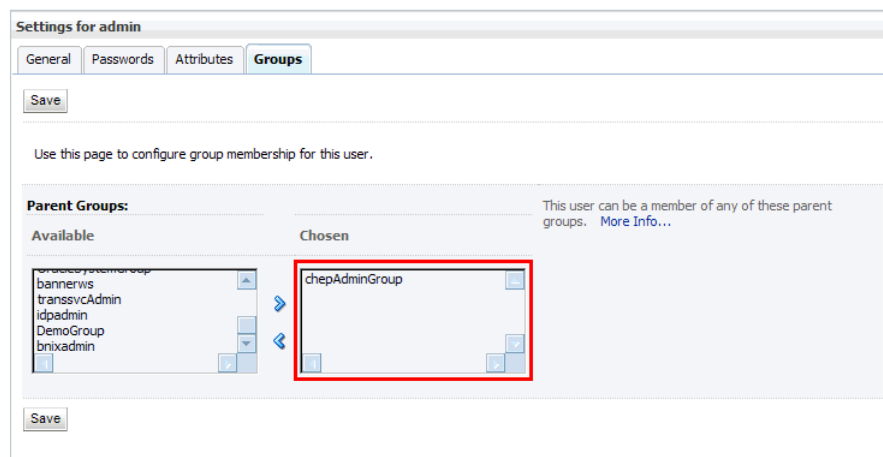
<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	Admin	Banner Cardholder Event Publisher Administrator	DefaultAuthenticator
<input type="checkbox"/>	bannerwsUser		DefaultAuthenticator
<input type="checkbox"/>	idproxy	Enterprise Identity Proxy Services User	DefaultAuthenticator
<input type="checkbox"/>	OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
<input type="checkbox"/>	transsvc		DefaultAuthenticator
<input type="checkbox"/>	weblogic		DefaultAuthenticator

12. Click the name of the user you just created. The Settings page for the user is displayed.

13. Select the **Groups** tab.



14. In the Parent Groups section, select *chepAdminGroup* in the **Available** list and move it to the **Chosen** list.



15. Click **Save**.

16. Restart the server for the changes to take effect.

Configuration

The Banner Cardholder Event Publisher must be configured before it can start publishing events. See “Appendix A, Administering Banner Cardholder Event Publisher” in the *Banner Web Services Handbook* for information on configuring the Publisher.



8 Test Banner Cardholder Event Publisher

Testing the Banner® Cardholder Event Publisher before integrating with a third-party campus card systems is highly recommended, but not required. Testing ensures proper configuration and provides experience publishing SyncEligibleCardholder messages.

To test the publication of SyncEligibleCardholder messages by the Banner Cardholder Event Publisher, a test application must be able to perform the following:

- Expose an endpoint that can receive SyncEligibleCardholder messages via SOAP over HTTP.
- Preferably, display the SyncEligibleCardholder messages it receives.

This chapter describes soapUI, an open source tool that meets these criteria. Other testing means are available, including direct testing with your institution's chosen campus card system.

Setup and use of soapUI

The open source soapUI tool, available from eviware (www.soapui.org), can be used to test exposed Web services. It can also be used to generate the mock implementation of a Web service based on a WSDL definition. Mock implementations are simulations of a Web service that can receive a SOAP message and reply with a predefined response.

A soapUI project is delivered with the Banner Web services download. This project provides a quick start for exposing a SyncEligibleCardholder endpoint for testing the Banner Cardholder Event Publisher.

Note

Refer to the soapUI documentation for more details on MockService. ■

Use the following steps to set up and use the soapUI tool:

- [Step 1, “Download and install soapUI”](#)
- [Step 2, “Open the testing workspace”](#)
- [Step 3, “Import the soapUI project”](#)
- [Step 4, “Start the MockService”](#)
- [Step 5, “Send a test message”](#)
- [Step 6, “Add accessible URL for the MockService”](#)

- [Step 7, “Reconfigure the Banner Cardholder Event Publisher”](#)
- [Step 8, “Test the Banner Cardholder Event Publisher”](#)

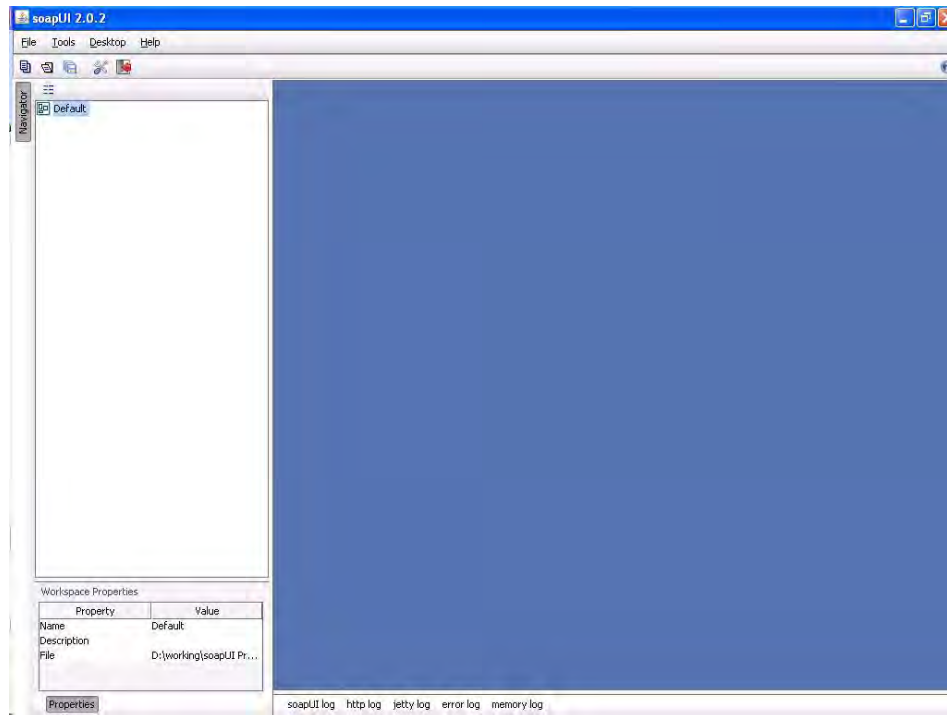
Step 1 Download and install soapUI

Download soapUI from eviware (www.soapui.org) and install it.

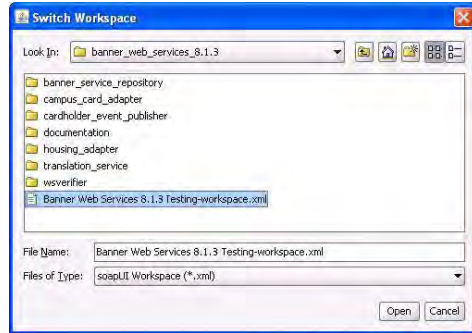
Step 2 Open the testing workspace

In soapUI, workspaces contain projects. Projects contain Web services definitions. The Banner Web services testing workspace is located in the root directory of the unzipped Banner Web services download. Use the following steps to open the testing workspace.

1. Open soapUI. A default workspace is displayed in the Navigator.



2. Select Switch Workspace from the File menu. The Switch Workspace window is displayed.
3. Navigate to Banner Web Services 8.1.3 Testing-workspace.xml in the root directory of the unzipped Banner Web services download:

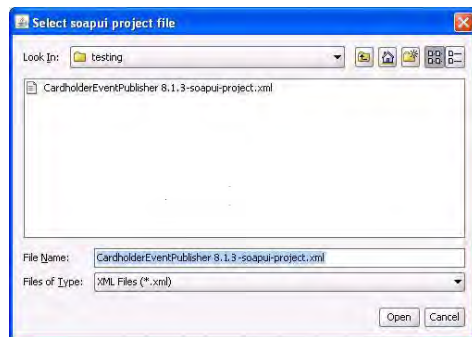


4. Click **Open**. The testing workspace is displayed.

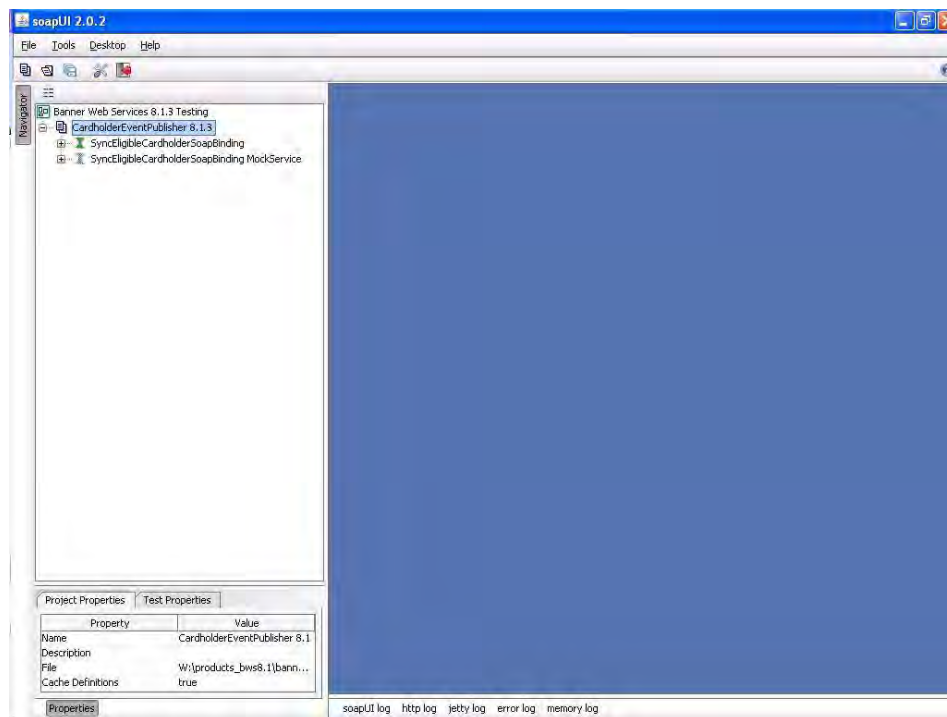
Step 3 Import the soapUI project

Use the following steps to import the soapUI project.

1. Select Import Project from the File menu. The Select soapui project file window is displayed.
2. Navigate to CardholderEventPublisher 8.1.3-soapui_project.xml in the cardholder_event-publisher\testing subdirectory of the unzipped Banner Web services download.



3. Click **Open**. The workspace for the project is displayed with two nodes:
 - **SyncEligibleCardholderSoapBinding**
 - **SyncEligibleCardholderSoapBinding MockService**

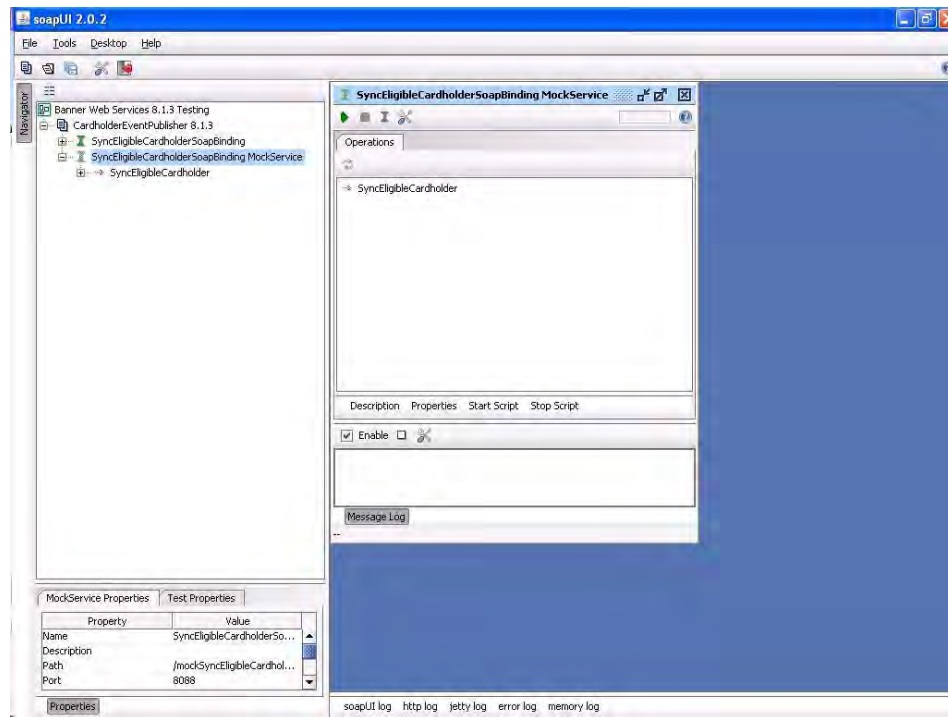


Step 4 Start the MockService

Use the following steps to start the MockService.

1. Select the **SyncEligibleCardholderSoapBinding MockService** node in the project.

2. Press Enter. The MockService Editor is displayed with three sections:
 - Toolbar
 - Operations pane, which displays the Web service operations supported by the MockService
 - Message Log pane, which displays messages as they are received

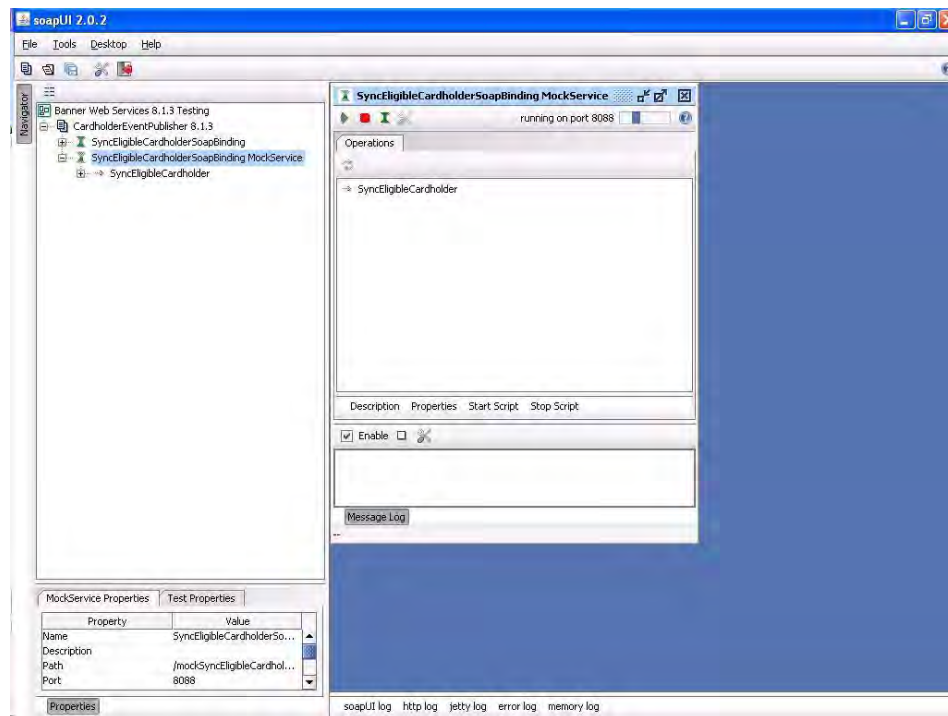


 **Note**

Tools and options for the Operations pane and the Message Log pane are not described in this document. ■

3. From the toolbar of the MockService Editor, press the green arrow icon to start the MockService. The following changes indicate that the MockService Editor is running:

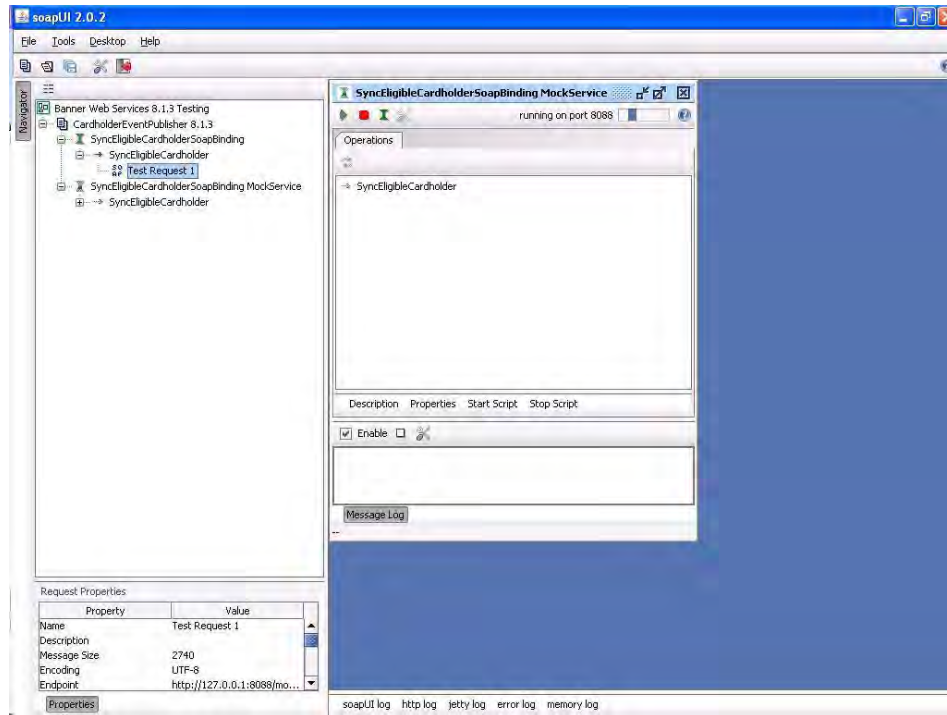
- The following message is visible in the toolbar: *running on port 8088*.
- The red stop icon is displayed.
- The green arrow icon is dimmed.



Step 5 Send a test message

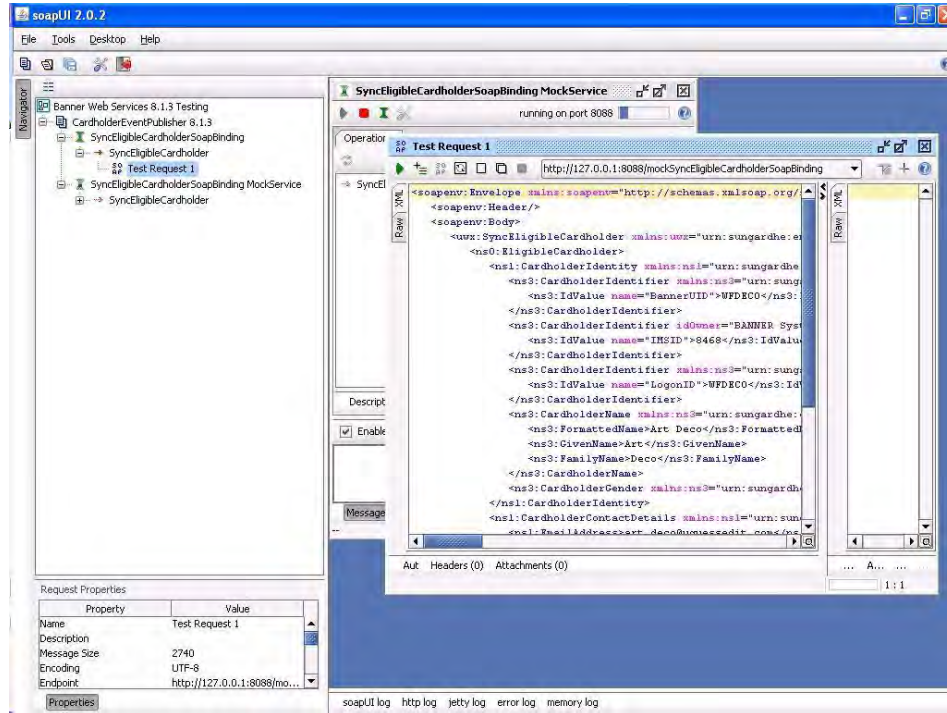
The MockService can now receive SyncEligibleCardholder messages. A message can be sent from soapUI itself. A test request message is provided in the project for this purpose. Use the following steps to send a test message.

1. Expand the SyncEligibleCardholderSoapBinding node, and then expand the SyncEligibleCardholder node to see Test Request 1.



2. Double-click Test Request 1. The Request Editor is displayed with three sections:

- Toolbar - The URL field displays the default endpoint for the MockService, which uses the computer's loopback IP address and default HTTP port rather than a specific hostname and port (`http://127.0.0.1:8088/mockSyncEligibleCardholderSoapBinding`). This URL can be used as a quick test to see how the MockService works.
- Request pane with a sample SOAP SyncEligibleCardholder message
- Response pane (initially empty)

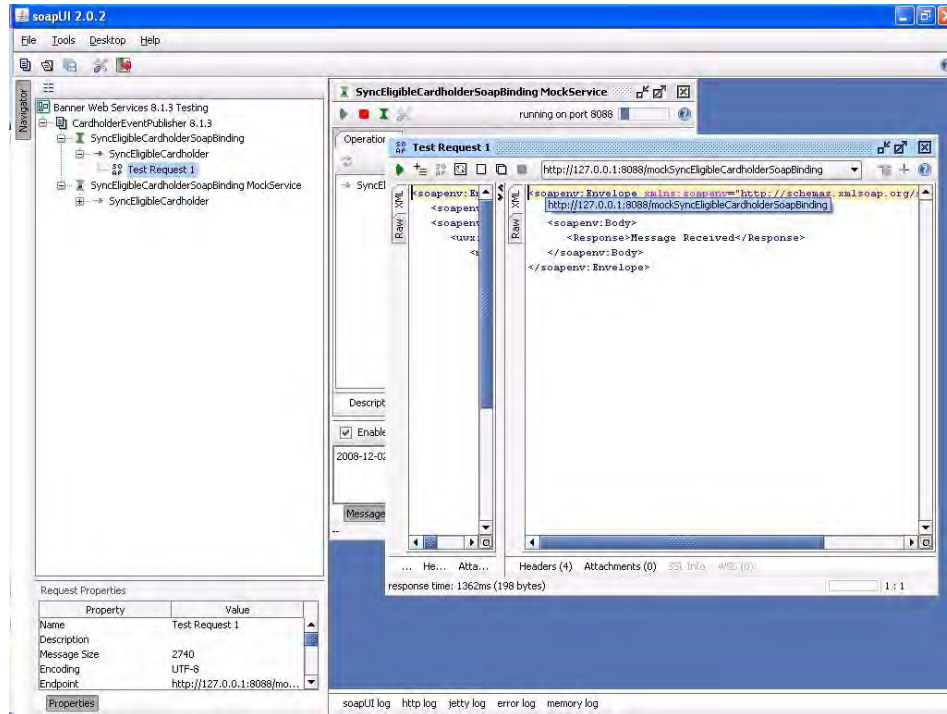


Note

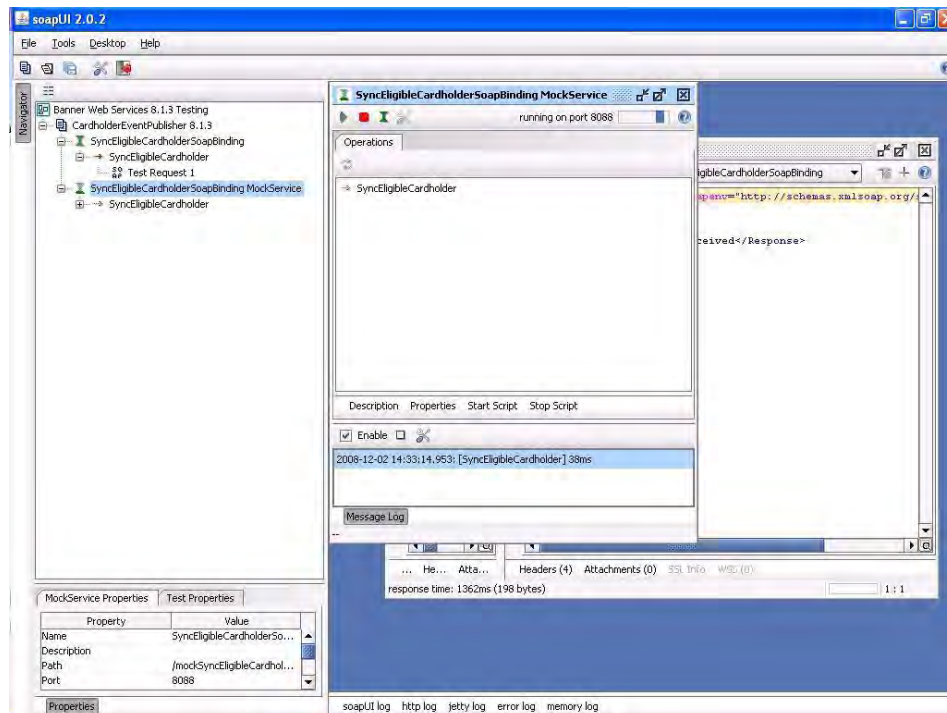
To use the MockService as a destination for messages from the Banner Cardholder Event Publisher, you must determine the appropriate hostname and port number for your computer. Additionally, you might need to disable VPN software and/or open the default HTTP port (8088) using Windows Firewall.

3. Click the green arrow icon in the toolbar of the Request Editor to send the SyncEligibleCardholder message to the MockService.

A Message Received message is displayed in the Response pane of the Request Editor.



A new entry is displayed in the Message Log pane of the MockService Editor.



4. Double-click the message in the Message Log pane. A Message Viewer displays the SyncEligibleCardholder message that was received.

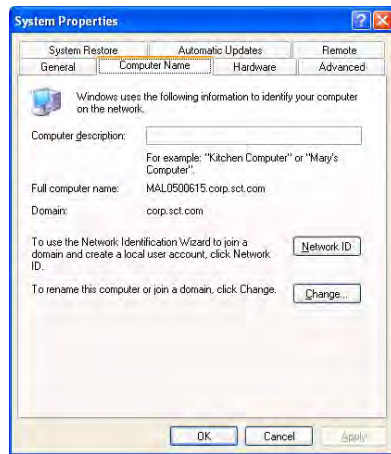
 **Note**

If the complete message is not visible, right-click in the Message Viewer and select Format XML from the context menu. This reformats the message to a displayable format. ■

Step 6 Add accessible URL for the MockService

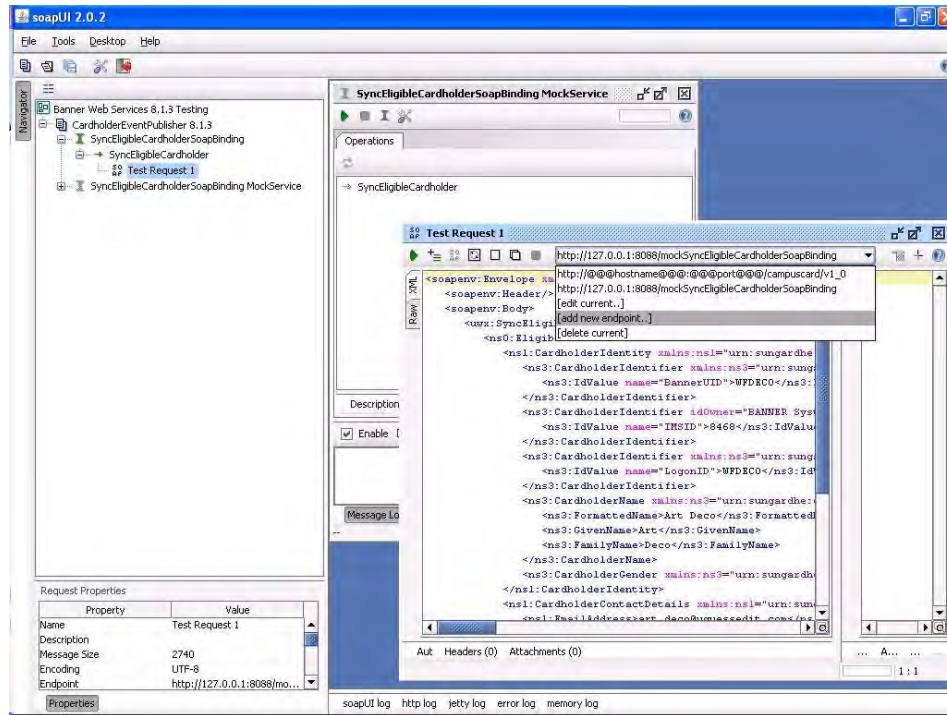
Use the following steps to determine and add the accessible URL for the MockService so that other computers can communicate with it.

1. Determine the hostname component of the URL by opening the Windows System Properties application from the Windows Control Panel. The value in the **Full computer name** field on the **Computer Name** tab can be used as the hostname component of the URL.



2. Determine the port number component of the URL. It can be kept as 8088, or it can be changed using the soapUI MockService Editor.
3. Add an exception in Windows Firewall that opens the port for inbound requests.

- Using the same test request (Test Request 1) or a clone of it, open the endpoint dropdown list from the toolbar of the Request Editor and select the **[add new endpoint]** option.



- On the Add new endpoint window, edit the endpoint to reflect the hostname and port number (if changed).



- Click **OK**.
- Click the green arrow icon in the toolbar of the Request Editor to send the SyncEligibleCardholder message to the MockService.

A Message Received message should be displayed in the Response pane of the Request Editor, and a new entry should be displayed in the Message Log pane of the MockService Editor. If you do not see these results, verify that any installed VPN software is disabled. Additional troubleshooting might be required.

Step 7 Reconfigure the Banner Cardholder Event Publisher

When the MockService can be reached using your computer’s hostname, reconfigure the Banner Cardholder Event Publisher. See “Appendix A, Administering Banner Cardholder

Event Publisher” in the *Banner Web Services Handbook* for information on configuring the Publisher.

Step 8 Test the Banner Cardholder Event Publisher

Test event publication by changing data in Banner and watching for messages in the Message Log pane of the MockService Editor.

Test cases

Once the Publisher is deployed to the Oracle Application Server and configured to post messages to the exposed endpoint, testing can begin. This entails creating data in Banner and monitoring the results. Typical test cases include the following:

- Create a person
- Update a person’s name
- Add an address for a person
- Add a telephone number for a person
- Add an e-mail address for a person
- Enroll a person as a student
- Add a student’s expected graduation year, term, and/or date
- Enroll a student in an active meal plan
- Assign an employee to a primary job that has an associated job location

Some of these test cases can have multiple contexts. For example, residence location addresses can be derived from information entered on the General Person Identification Form (SPAIDEN) and stored in the SPRADDR table or from information entered on the Room Assignment Form (SLARASG) and stored in the SLRRASG table. Be sure to conduct tests that are appropriate for your environment.