SUNGARD HIGHER EDUCATION

BANNER ENTERPRISE IDENTITY SERVICES HANDBOOK

Release 8.1.5, Revision 2
January 2012

Trademark, Publishing Statement and Copyright Notice

Prepared by: SunGard Higher Education

4 Country View Road
Malvern, Pennsylvania 19355
United States of America

Customer Support Center Website

http://connect.sungardhe.com

Documentation Feedback

http://education.sungardhe.com/survey/documentation.html

Distribution Services E-mail Address

distserv@sungardhe.com

Revision History Log

Publication Date	Summary
October 2011	New version that supports Banner Enterprise Identity Services 8.1.5 software.
January 6, 2012	Revised installation instructions for Oracle WebLogic Server 11g.
January 9, 2012	Revised configuration instructions for Banner Identity Gateway.

Banner Enterprise Identity Services 8.1.5 Handbook

Contents

Chapter 1 Overview 1-1

Definitions 1-1

Account provisioning 1-1

Access management 1-2

BEIS support for identity and access management 1-2

Common identity 1-2

Common identifier 1-3

Common architecture 1-3

Example configurations 1-5

Outbound account provisioning 1-5

Inbound account provisioning 1-6

Single sign on 1-7

Requirements 1-7

Professional Services engagement 1-8

Banner products 1-8

Oracle application server 1-8

Oracle database 1-9

Oracle Streams 1-9

Scope of this handbook 1-10

Chapter 2 UDCIdentity 2-1

Data exchange constraints 2-1

Child attributes and elements 2-1

UDCIdentifier element 2-13

InstitutionRoles element 2-13

Supported institution roles 2-13

Use of rules to determine institution roles 2-14

Roles in the GORIROL table 2-17

Role administration 2-18

Extension element. 2-20

Sample UDCIdentity operations 2-21

Create operation 2-21

Query operation 2-21

Update operation 2-22

Delete operation. 2-26

Chapter 3 Components That Support Account Provisioning and SSO 3-1

Outbound account provisioning 3-1

Oracle Streams 3-2

Streams Rules Configuration Form (GUASADM) 3-3

Crosswalk Validation Form (GTVSDAX) 3-3

Business Rules Form (GORRSQL) 3-4

BannerIdentity Topic 3-4

UDCIdentity Topic 3-4

BEIS components 3-4

Identity management system 3-6

Inbound account provisioning 3-7

Identity management system 3-8

Banner Identity Gateway 3-8

Banner APIs 3-8

Single sign on 3-10

Central access manager 3-10

SSO Manager 3-10

Banner Web Tailor 3-10

baniam.jar 3-11

Chapter 4 Implementation Road Maps 4-1

Banner outbound account provisioning 4-1

Banner inbound account provisioning 4-2

Single sign on to Banner 4-3

Chapter 5 Banner Configuration 5-1

BEIS_Check_script 5-1

Configuration of your environment 5-6

Step 1 Validate the Oracle initialization parameters and database configuration . 5-6
Step 2 Verify the registration of identity_domain XML schema 5-8
Step 3 Populate the Business Rule Process Parameters Form (GORSQPA) . . . 5-8
Step 4 Validate Oracle Streams metadata 5-9
Step 5 Define attributes to be used as the LDAP username and password 5-11
Step 6 Create a population selection to select IDs to provision into LDAP 5-13
Step 7 Configure element content via GTVSDAX 5-13
Step 8 Configure and start the capture process and apply process 5-16
Step 9 Validate Oracle user accounts for fully privileged INB access 5-17

Customization of the UDCIdentity XML structure 5-17

Step 1 Define the attribute selection rule 5-18
Step 2 Define the attribute change capture rule 5-19
Step 3 Enable the capture rule 5-20

Definition of rules to monitor database changes 5-20

GUASADM - Key block 5-21
GUASADM - Capture Tables block 5-21
GUASADM - Capture Rules block 5-22
GUASADM - Capture Columns block 5-23

Chapter 6 Identity Data Export Utilities 6-1

Installation on Oracle Application Server 6-1

Step 1 Configure the database objects 6-2
Step 2 Specify the working directory 6-2

Step 3 Install the Identity Data Export Utilities 6-3
Step 4 Configure the security role and user 6-7
Step 5 Configure logging 6-11

Installation on Oracle WebLogic Server 11g 6-12

Recommended configuration 6-12
Installation steps 6-12
Step 1 Configure the database objects 6-13
Step 2 Customize properties 6-13
Step 3 Install the Identity Data Export Utilities 6-15
Step 4 Configure the security group and user 6-22

Configuration 6-27

Configure the UDCIdentifier Assigner 6-28
Configure the UDCIdentity Extractor 6-29
Configure the LDIF Generator 6-31
Configure the SPML Publisher 6-33

Using Identity Data Export Utilities 6-34

Assign a UDCIdentifer to persons in Banner 6-35
Extract UDCIdentity information from the database 6-35
Generate LDIF files 6-36
Publish SPML files 6-37
Perform file operations 6-37

Chapter 7 Banner Identity Gateway 7-1

Installation options 7-1

Manual installation 7-1
Automated installation 7-1

Installation on Oracle Application Server 7-2

Step 1 Configure the database user and schema 7-3
Step 2 Configure the integmgr user password 7-4
Step 3 Define the data source for Oracle Advanced Queuing 7-5
Step 4 Define the data source for the Banner Identity Gateway 7-10
Step 5 Define the data source for the Oracle Streams administrator 7-11
Step 6 Define the data source for the Banner security administrator 7-13

Step 7 Configure the security role and user 7-14
Step 8 Configure the JMS queues and topics 7-19
Step 9 Deploy the Banner Identity Gateway 7-23
Step 10 Configure logging 7-26

Installation on Oracle WebLogic Server 11g 7-26

Recommended configuration 7-26
Installation steps 7-27
Step 1 Configure the database user and schema 7-28
Step 2 Customize properties 7-29
Step 3 Configure the authentication provider 7-31
Step 4 Define the data source for Oracle Advanced Queuing 7-39
Step 5 Define the data source for the Banner Identity Gateway 7-45
Step 6 Define the data source for the Oracle Streams administrator 7-47
Step 7 Define the data source for the Banner security administrator 7-49
Step 8 Configure the security group and user 7-50
Step 9 Create a JMS server 7-56
Step 10 Create a JMS module 7-59
Step 11 Configure a JMS topic and connection factory 7-63
Step 12 Deploy the Banner Identity Gateway 7-72

Configuration 7-78

Configure Gateway metadata 7-78
Modify Gateway metadata 7-79

Administration 7-80

Search for and edit errors 7-80
Display all edits for an error 7-81
Remove an error log 7-81
View the status of Oracle Streams processes 7-82
Start Oracle Streams IAM processes 7-82
Stop Oracle Streams IAM processes 7-83
Load updated capture rules 7-83

Chapter 8 Enterprise Identity Proxy Services 8-1

Installation options 8-1
Manual installation 8-1
Automated installation 8-2

Installation on Oracle Application Server 8-2
Step 1 Configure the database user and schema 8-3
Step 2 Change processing parameters (optional) 8-4
Step 3 Define the data source 8-5
Step 4 Configure the security role and user 8-10
Step 5 Configure the JMS queues 8-15
Step 6 Install the Identity Proxy 8-19
Step 7 Configure logging 8-22

Installation on Oracle WebLogic Server 11g 8-22
Recommended configuration 8-22
Installation steps 8-23
Step 1 Configure the database user and schema 8-24
Step 2 Customize properties 8-25
Step 3 Define the data source 8-27
Step 4 Configure the security group and user 8-34
Step 5 Create a JMS server 8-39
Step 6 Create a JMS module 8-39
Step 7 Configure a JMS queue and connection factory 8-40
Step 8 Install the Identity Proxy 8-48

Configuration of PSPs 8-54
Add or update a PSP configuration 8-55
Delete a PSP configuration 8-57

Administration 8-57
View message details 8-58
Delete a message that is creating an error 8-58
Search for messages 8-59

Chapter 9 Automated Installer 9-1

Prerequisites 9-1

Modes for using the installer 9-2

Installation on Oracle Application Server 9-2

Step 1 Note the deployer URL 9-2
Step 2 Configure and install the applications 9-3
Step 3 Configure security roles and users 9-11
Step 4 Configure logging 9-16

Installation on Oracle WebLogic Server 11g 9-17

Recommended configuration 9-17
Installation steps 9-18
Step 1 Configure the domain security model 9-18
Step 2 Configure the authentication provider 9-20
Step 3 Configure and install the applications 9-30

Chapter 10 SPML LDAP Adapter 10-1

Installation on Oracle Application Server 10-1

Step 1 Extract the ear file 10-2
Step 2 Configure the SPML LDAP Adapter 10-2
Step 3 Rebuild the ear file 10-6
Step 4 Install the SPML LDAP Adapter 10-6

Installation on Oracle WebLogic Server 11g 10-10

Recommended configuration 10-10
Installation steps 10-11
Step 1 Extract the ear file 10-11
Step 2 Configure the SPML LDAP Adapter 10-11
Step 3 Rebuild the ear file 10-12
Step 4 Install the SPML LDAP Adapter 10-13

Chapter 11 SSO Manager 11-1

Definitions 11-1

How the SSO Manager facilitates SSO 11-1

SSO for Self-Service Banner (SSB) 11-1
SSO for Internet-native Banner (INB) 11-4

Implementation of the SSO Manager 11-6
Prerequisites. 11-7
Establish the environment 11-7
Configure the domain security model. 11-8
Configure the authentication provider 11-11
Run the configuration utility 11-19
Complete the installation on Oracle Application Server 11-31
Step 1 Create a database user and objects 11-32
Step 2 Define the integmgr data source 11-33
Step 3 Define the data source for the Banner security administrator 11-38
Step 4 Define the data source for connecting to the SSO Manager schema . . 11-39
Step 5 Create a security user 11-41
Step 6 Deploy the ear file 11-45
Complete the installation on Oracle WebLogic Server 11g 11-48
Step 1 Create a database user and objects 11-49
Step 2 Define the integmgr data source 11-50
Step 3 Define the data source for the Banner security administrator 11-56
Step 4 Define the data source for connecting to the SSO Manager schema . . 11-58
Step 5 Create a security user 11-60
Step 6 Deploy the ear file 11-66
Migrate credentials (optional) 11-73
Configure the supporting components 11-73
Verify the configuration 11-91

Application development with the SSO Manager 11-92
Deep-linking 11-92
Using the Ticketing Web service 11-95
Using the Credential Web service. 11-96
Incorporating CAS service validation 11-99

Use of the SSO Manager without BEIS account provisioning 11-103

Chapter 12 Middleware Validation 12-1

Prerequisites 12-1

Oracle Streams 12-1

Validation 12-1

Troubleshooting 12-3

Banner Identity Gateway 12-3

Validation 12-3

Troubleshooting 12-5

Enterprise Identity Proxy Services 12-6

Chapter 13 Luminis Platform Configuration 13-1

Prerequisite 13-1

Information flow for account provisioning 13-2

Luminis Platform configuration 13-3

Step 1 Configure Luminis Platform properties 13-3

Step 2 Set up and enable SSO 13-5

Step 3 Verify the URL for the CAS server 13-7

Step 4 Access the Luminis Provisioning Gateway 13-9

Step 5 Migrate current users to be enterprise users 13-9

Chapter 14 Banner Workflow Configuration 14-1

Prerequisites 14-1

Account provisioning 14-1

Information flow 14-1

Configuration of the Banner Workflow Provisioning Gateway 14-7

Step 1 Install the Banner Workflow Provisioning Gateway 14-7

Step 2 Configure the Banner Workflow Provisioning Gateway 14-8

Step 3 Build ear file 14-11

Step 4 Deploy ear file 14-12

Step 5 Configure Enterprise Identity Proxy Services 14-12

Single sign on (SSO) authentication 14-13
SSO under CAS 14-13
Step 1 Configure Banner Workflow for CAS 14-14
Step 2 Configure <LogoffUrl> element (optional) 14-15
Step 3 Set up Oracle Application Server SSL 14-15
Step 4 Register Banner Workflow with CAS server 14-15
Step 5 Configure Luminis Platform for Banner Workflow 14-15
Step 6 Modify Banner Forms technology type 14-16
SSO under a third-party access manager 14-17
Step 1 Configure IDM Gateway 14-17
Step 2 Configure Banner Workflow for IDM Gateway 14-17
Step 3 Configure <LogoffUrl> element (optional) 14-18
Step 4 Configure Luminis Platform for Banner Workflow 14-18
Step 5 Modify Banner Forms technology type 14-19

Chapter 15 Banner Document Management Suite Configuration 15-1

Prerequisites 15-1

Account provisioning 15-1
Information flow 15-2
Installation of BDMS User Provisioning Web service 15-6
Configuration of Enterprise Identity Proxy Services 15-7
Configuration of the BDMS role 15-8
Step 1 Create a new Additional ID Type 15-8
Step 2 Create the BDMS role 15-9
Step 3 Define a selection rule 15-9

Single sign on (SSO) authentication 15-10
Installation of SSO files 15-10
Configuration of SSO under CAS 15-11
Step 1 Modify login.aspx 15-11
Step 2 Modify bdms.sso.config 15-11
Step 3 Add AX Web Access as a protected service on the CAS server 15-12
Step 4 Configure Banner 15-12
Configuration of SSO under a third-party access manager 15-12
Step 1 Modify login.aspx 15-13
Step 2 Modify bdms.sso.config 15-13

Step 3 Configure BDMS on the third-party access manager server 15-14
Step 4 Configure Banner 15-15

Appendix A Oracle Streams A-1

Stop Oracle Streams processes A-1

Start Oracle Streams processes A-1

Create attribute selection rule A-2

Modify capture definitions A-2

Appendix B UDCIdentity Message Sample B-1

Sample message B-1

Appendix C EduPerson Data Mapping C-1

LDAP C-1
objectClass attribute C-2
EduPerson higher education attributes C-2

EduPerson and BEIS C-3
Banner identity components C-4
UDCIdentity XML structure standards C-4

Appendix D CAS Installation and Configuration D-1

Introduction to CAS D-1

Prerequisites D-2

Configuration of the CAS server D-2
Step 1 Download JA-SIG CAS server distribution D-3
Step 2 Download CAS extensions jar file D-3
Step 3 Modify WEB-INF/web.xml D-4
Step 4 Modify WEB-INF/cas.properties D-4
Step 5 Modify WEB-INF/spring-configuration/uniqueIdGenerators.xml D-5

Step 6 Modify WEB-INF/spring-configuration/argumentExtractorsConfiguration.xml D-5
Step 7 Modify WEB-INF/cas-servlet.xml D-5
Step 8 Modify WEB-INF/deployerConfigContext.xml D-6
Step 9 Modify WEB-INF/classes/default_views.properties D-15
Step 10 Deploy application D-15
Step 11 Test CAS Web application D-15
Step 12 Modify Tomcat (if needed) D-16
Step 13 Define CAS managed services D-17

Configuration of Self-Service Banner for CAS D-19

Configuration of Internet-native Banner for CAS D-20

Configuration of Luminis Platform for CAS D-20

Step 1 Configure Luminis Platform with CAS server information D-21
Step 2 Add Luminis service to CAS server D-22

Configuration of Banner Workflow for CAS D-23

Configuration of Banner Document Management Suite for CAS D-23

1 Overview

Institutions are under increasing pressure to manage the security of their computer and network systems with enhanced protection for sensitive data. An essential element of security and privacy is identity and access management. Processes, technologies, and policies are combined to manage digital identities and specify how they are used to access digital resources such as your institution's information systems.

Definitions

It is important to understand two terms: *account provisioning* and *access management*.

Account provisioning

A central component of identity and access management is user account provisioning. User account provisioning refers to the creation, maintenance, and deactivation of user objects and user attributes in one or more directories or applications as a result of automated or interactive business processes. The objective is to provide an individual with the appropriate access to enterprise applications based on the individual's affiliation with the institution (for example, faculty, staff, or student).

User account provisioning is enabled by establishing or implementing a central identity vault that stores the enterprise definition of identity. ERP applications, such as those for human resources or student information, populate the identity vault with person data as the data is created, updated, and deleted during standard business processes.

Changes to data in the identity vault result in subsequent provisioning or de-provisioning of user accounts in other applications. This is particularly true when the central identity vault is part of a commercial enterprise identity management system (EIMS). These systems provide rule-based account provisioning capabilities as well as adapters or tool kits for integrating with other applications. Alternatively, the central identity vault can be implemented as a directory or application database. These lightweight solutions, however, often require more extensive in-house development.

Regardless of how the central identity vault is implemented, its existence and the existence of an enterprise definition of identity are critical first steps in establishing a centralized identity management infrastructure.

With these capabilities in place, institutions can implement centralized access management leading to trust-based authentication and the ability to participate in identity federations. (A federation is an association of organizations that use common attributes, practices, and policies to exchange information about their users and resources to enable collaboration and transactions.)

Access management

Another fundamental component of identity and access management is centralized authentication for purposes of access control. This centralized access management is commonly called single sign on (SSO).

Authentication is the process of verifying the digital identity of the sender of a communication, such as a request to log in. Once the identity of the sender is verified, access to system resources can be granted. Traditionally, individual computer systems granted access to valid users and excluded unauthorized users. SSO, on the other hand, enables a user to authenticate once and gain access to multiple software systems.

By centralizing access management, clients can affect access at one location and choose from various authentication methods beyond the normal username and password that are common in legacy authentication methods.

BEIS support for identity and access management

SunGard® Higher Education recognizes the importance of centralized identity and access management. A collection of common software components and embedded capabilities in Banner® support your institution's identity and access management architecture. Collectively, these components and capabilities are called Banner Enterprise Identity Services (BEIS).

The following BEIS features support your institution's identity and access management infrastructure:

- Common definition of identity across SunGard Higher Education applications
- Common identifier across SunGard Higher Education applications
- Common architecture and components for account provisioning and access management

Common identity

The foundation of the BEIS architecture is a common definition of identity that is shared among SunGard Higher Education applications. Implemented in W3C XML Schema, the

UDCIdentity XML structure collects and packages identity data about a user of a SunGard Higher Education application. The UDCIdentity XML structure also provides the basis for exchanging user data between SunGard Higher Education applications and external provisioning components such as a third-party EIMS or a central directory.

If Banner is the authoritative source for identity data, the UDCIdentity XML structure is used to provision enterprise user accounts into an identity vault. This data structure is then used to provision other SunGard Higher Education applications as appropriate. If Banner is not the authoritative source for identity data, Banner can be provisioned using the UDCIdentity XML structure.

Common identifier

Part of the UDCIdentity XML structure is a globally unique identifier (GUID) that is assigned to each person record published in a UDCIdentity message.The UDCIdentifier is an unchanging, system-generated, 32-character, alphanumeric value. The following example shows the UDCIdentifier in the UDCIdentity XML structure:

```
<UDCIdentifier>36BE6D6D18560C44E0440003BA33B440</UDCIdentifier>
```

The UDCIdentifier is the primary identifier for each user that is synchronized with the EIMS or central directory, ensuring that user data objects are properly located, created, and updated. The UDCIdentifier also provides the basis of claims-based authentication.

Common architecture

BEIS components support different configurations for identity and access management. This flexibility is a fundamental feature of the common BEIS architecture. Other fundamental features of the architecture include the following:

- Reliance on a central identity vault
- Use of Service Provisioning Markup Language (SPML)
- A defined set of roles for Banner and other SunGard® Higher Education applications
- Authentication from a central access manager

Central identity vault

An identity vault serves as a central repository for identity in the enterprise. Whether implemented as a third-party EIMS, as a central directory, or even with Banner itself, the existence of the vault establishes standard patterns for the exchange of identity data. Changes to data in the identity vault result in the provisioning or de-provisioning of accounts in enterprise applications. Enterprise applications that are authoritative for person data, such as Banner or another ERP, can feed the identity vault, stimulating provisioning activities.

The actual flow of identity data among applications at your institution is determined by your institution's chosen configuration.

Service Provisioning Markup Language (SPML)

The use of Service Provisioning Markup Language, specifically SPML 2.0, is a fundamental feature of the BEIS architecture. SPML is an XML-based OASIS standard that defines a protocol for exchanging user, resource, and service provisioning requests. SPML defines the roles of applications that exchange provisioning data and the structure and content of provisioning request messages. The use of SPML 2.0 allows BEIS components to package UDCIdentity XML content within defined SPML request messages.

Role of SunGard Higher Education applications

The BEIS architecture establishes standard roles for Banner and other SunGard Higher Education applications. These roles are defined by SPML.

Banner can be the authoritative source for person data in the enterprise, or Banner can be the recipient of changes that occur in the central identity vault.

- If Banner is the authoritative source, identity events in Banner cause a central SPML Request Authority (RA) to make provisioning requests to one or multiple Provisioning Service Providers (PSPs). In this role, Banner feeds the central identity vault.
- If Banner is the recipient, data changes in the central identity vault are sent to Banner. Banner is a Provisioning Service Target (PST) that receives provisioning requests from its PSP (the Banner Identity Gateway).

Other SunGard Higher Education products never feed the central identity vault. They are always the recipients of provisioning requests. In terms of SPML, Luminis® Platform 4.x, Banner Workflow, and Banner Document Management Suite are always Provisioning Service Providers (PSPs). Their respective data stores are Provisioning Service Targets (PSTs).

Central access manager

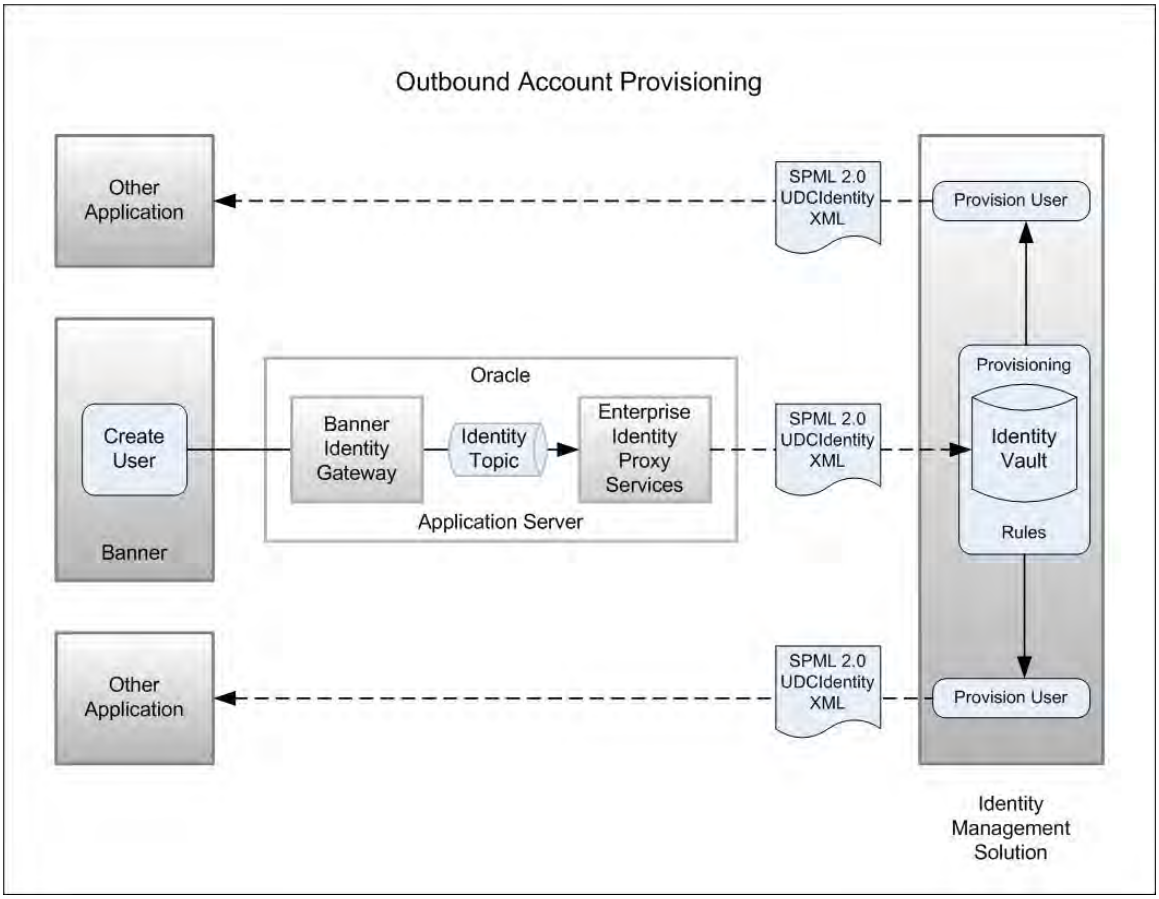
With the UDCIdentifier provisioned via the UDCIdentity XML structure, trust-based authentication (single sign on) between a central access manager and SunGard Higher Education applications is possible. After a user successfully authenticates to a central access manager, the user's digital identity is asserted through the UDCIdentifier. Applications can use this natively to identify the user and grant access. If this is not possible, applications can use BEIS components for storing and retrieving application-specific credentials.

Example configurations

The following example configurations highlight the flexibility of the BEIS architecture. BEIS supports other possible configurations.

Outbound account provisioning

In this example, Banner is the authoritative source for person data. An EIMS houses the identity vault and acts as a Provisioning Service Provider in relation to the SPML messages sent to it by Enterprise Identity Proxy Services.



A business process results in the addition or change of identity-related data in Banner. As a result, the following provisioning activities are performed:

1. The addition or change is published as an XML message.
2. The XML message is transformed into a standard XML representation used by all compliant SunGard Higher Education applications.

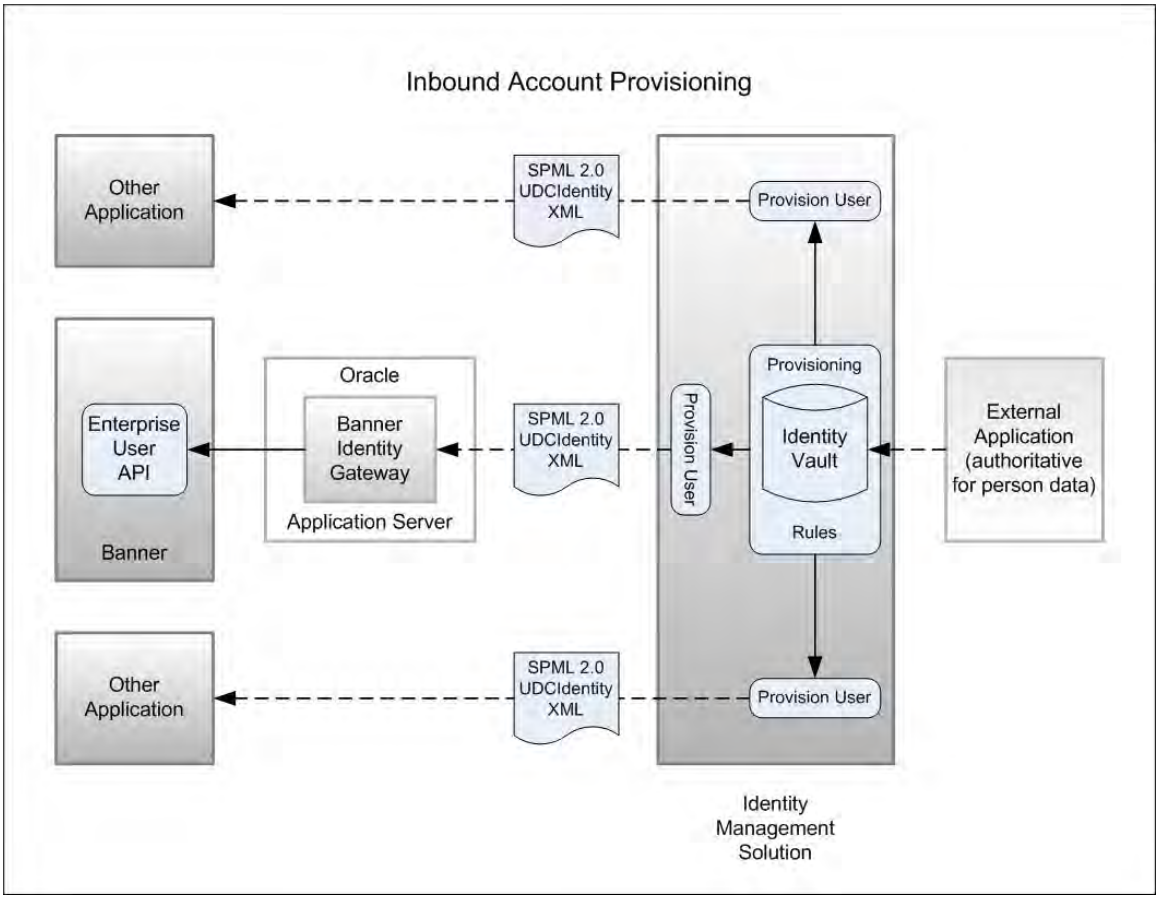
3. The data is packaged in a standard SPML 2.0 request message and sent to the implemented EIMS.

4. The EIMS updates its identity vault.

5. Based on the rules established, the EIMS provisions or de-provisions other applications as appropriate. For SunGard Higher Education non-Banner applications, the EIMS communicates changes using the same standard identity XML format used by Banner, packaged in an SPML 2.0 request message, and communicated through the common Enterprise Identity Proxy Services.

Inbound account provisioning

The BEIS architecture also supports configurations in which Banner is not the authoritative source for identity. An example of this type of configuration follows.



In this example, an EIMS is implemented and an external application, such as a third-party HR system, is the authoritative source for person data. Provisioning activities similar to the preceding example are performed:

1. Additions and changes to identity-related information update the identity vault of the EIMS.

2. The EIMS provisions or de-provisions other applications based on established rules. For SunGard Higher Education applications, including Banner, this is accomplished by sending SPML 2.0 request messages to the Web services that are exposed by each application.

Single sign on

The SSO Manager, a component of BEIS, serves as a single sign on gateway for Self-Service Banner (SSB) and Internet-native Banner (INB). The SSO Manager relies on a central access manager to authenticate a user and assert the user's identity. Once a user is authenticated, the SSO Manager performs the operations that allow the user to access Banner. The SSO Manager exposes these operations as services that other applications can use, as needed, to implement single sign on.

The SSO Manager supports the following SSO configurations:

- **Third-party authentication**. BEIS allows Banner to participate in environments that are controlled by a third-party enterprise identity management system (EIMS). The SSO Manager is protected by the EIMS, which authenticates the user and asserts the user's identity to the SSO Manager via a cookie or HTTP header. (The cookie or header is configured in the SSO Manager.)

- **CAS-based SSO**. The SSO Manager can be configured to operate in CAS mode. The SSO Manager becomes a service that is protected by CAS, which authenticates the user. After the user is authenticated, the SSO Manager invokes a validation service (/samlValidate or /bannerValidate) that is exposed by the CAS server. These services validate the CAS session and provide the identity of the user to the SSO Manager in a defined XML format.

- **CAS-based SSO embedded in Luminis Platform 5.x**. Single sign on with Luminis Platform 5.x is a variant of CAS-based SSO. Luminis Platform 5.x includes a CAS server and supports CAS-based authentication natively. All configuration is done on the embedded CAS server.

Requirements

SunGard Higher Education recommends a Professional Services engagement in addition to certain required software components.

Professional Services engagement

Banner plays an important role in how an institution manages identities. To position Banner as a friendly citizen of identity and access management, a firm understanding of identity provisioning and single sign on (SSO) is required.

The “Identity Provisioning Service” is a Professional Services engagement that is strongly recommended before installing, using, and receiving customer support for BEIS. This service provides the following benefits:

- Structural and functional overview of the database and application server components of BEIS
- Firm understanding of how to apply the technology to meet institution-specific needs
- Effective maintenance and troubleshooting techniques required to maintain the database and application server components

Contact your SunGard Higher Education Account Manager to arrange for the “Identity Provisioning Service” service engagement.

Banner products

If you license any Banner product that uses the BEIS components, you can download, install, and use BEIS without any additional licensing.

BEIS requires the following Banner products:

Product	Minimum Version
Banner General	8.3.1
Banner Web Tailor	If Cascade is enabled, version 8.4.2 is the minimum version required. If Cascade is not enabled, version 8.3.1 plus patch `p1-dc06g9_twb8030101` is the minimum version required.
Banner Intcomp	8.0.1

Oracle application server

The BEIS application requires one of the following application servers.

Oracle Application Server (OAS)

BEIS was developed and tested on OAS versions 10.1.3.4/5 with Java 1.6.

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 SunGard Higher Education 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
SunGard Higher Education FAQ:	1-AXZ803
Oracle Doc ID:	351476.1

Oracle WebLogic Server 11g

BEIS was developed and tested on Oracle WebLogic Server 11g version 10.3.2. This handbook documents installation on version 10.3.2.

Note

Additional testing was performed on version 10.3.5. There are minor documentation differences between installation on versions 10.3.2 and 10.3.5.

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 (10.3.2)	Oracle Database 11g

Oracle Streams

An Oracle Streams environment for BEIS is required.

Scope of this handbook

This handbook describes the installation, configuration, and use of the BEIS components that support account provisioning and authentication. This includes details on native Banner capabilities and middle tier components.

This handbook also describes the use of BEIS with Luminis Platform 4.x, Banner Workflow, and Banner Document Management Suite (BDMS).

Note

Luminis Platform 5.x uses Banner Integration for eLearning, rather than Banner Enterprise Identity Services, for inbound account provisioning.

This handbook does not cover specific EIMSs.

2 UDCIdentity

The foundation of the Banner® Enterprise Identity Services (BEIS) architecture is a common definition of identity that is shared among SunGard® Higher Education applications. Implemented in W3C XML Schema, the UDCIdentity XML structure collects and packages identity data about a user of a SunGard Higher Education application. The UDCIdentity XML structure also provides the basis for exchanging user data between SunGard Higher Education applications and external provisioning components such as a third-party enterprise identity management system (EIMS) or a central directory.

If Banner is the authoritative source for identity data, the UDCIdentity XML structure is used to provision enterprise user accounts into an identity vault. This data structure is then used to provision other SunGard Higher Education applications as appropriate. If Banner is not the authoritative source for identity data, Banner can be provisioned using the UDCIdentity XML structure.

Data exchange constraints

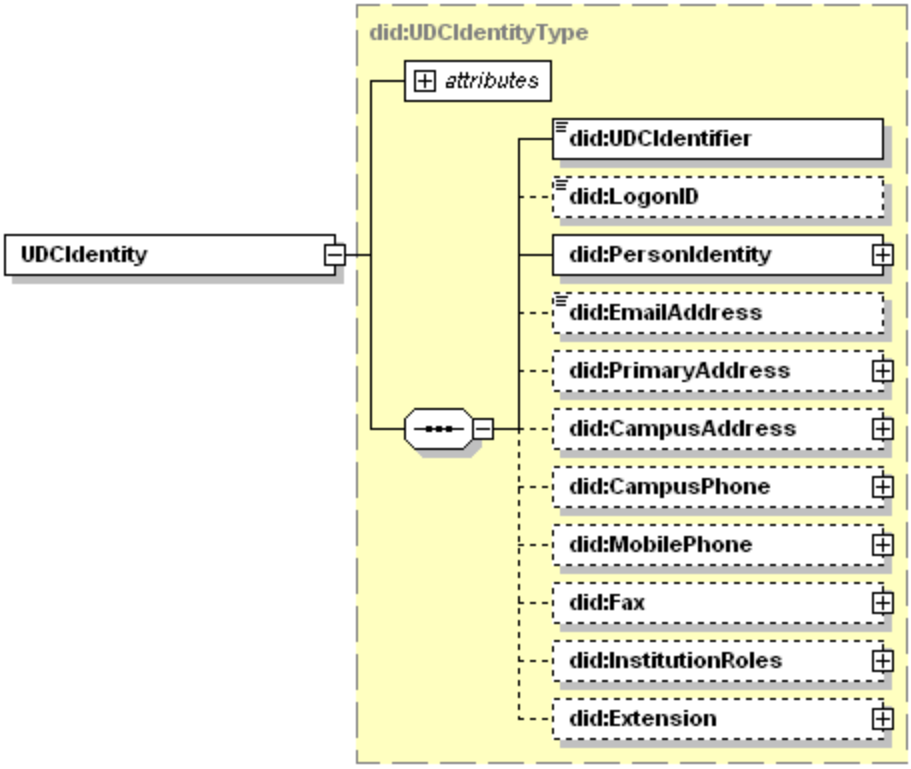
Data exchange between applications using the UDCIdentity XML structure is constrained by the following rules:

- A UDCIdentity message always carries a full snapshot image of the user.
- User roles are the only exception to the “full snapshot” rule. Applications that consume UDCIdentity messages may know about more user roles than those contained in the UDCIdentity XML structure.
- The UDCIdentity representation of a user’s profile is binding on any application that is creating application users based on it.

Child attributes and elements

The UDCIdentity XML structure includes person information that is normally stored in a central directory. Person information includes name, gender, birth date, email address, primary address, campus address, campus phone, mobile phone, fax, and institutional roles. Some structures are complex, having child attributes.

The following graphic depicts the immediate children of the UDCIdentity XML structure.



Appendix B, "UDCIdentity Message Sample" provides a sample XML message.

The following table describes all child attributes and elements of the UDCIdentity XML structure and maps their content to columns in the Banner database.

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
/UDCIdentity	XML object holding common attributes of a person that are shared across SunGard Higher Education applications.		
./@PUBLISHER_NAME	Publisher of the UDCIdentity message. Used to resolve the application context of data (for example, roles from Banner, roles from Banner Workflow). Should provide enough application context information for the consuming applications.	Banner	Populated by the eventing infrastructure. Not used for inbound account provisioning to Banner.
./@action	Action or operation to be performed on the object (for example, add, update, or delete).	ADD	Populated by the eventing infrastructure based on the action that triggered publication. Not used for inbound account provisioning to Banner.
./UDCIdentifier	Unique, 32-character alphanumeric identifier for the person.	36BE6D6D18560C44E0440003BA33B440	Generated by the Banner GUID Service. See "UDCIdentifier element" on page 2-13 for details.
./PersonIdentity	Identifying attributes of the person.		
././PersonName	Names of the person.		
././.FormattedName	Fully formatted name, contained in one string, with all pieces in their proper place. Includes all necessary punctuation. Cannot be easily parsed.	Mrs. Elizabeth Ann Croton, MD	Available elements concatenated in the following order: Prefix, GivenName, MiddleName, FamilyName, Suffix.

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./../LegalName	Name used for legal documentation or other legal purposes. Contains, in one string, a fully formatted name with all pieces in their proper place. Includes all necessary punctuation.	Mrs. Elizabeth Ann Croton, MD	SPRIDEN_CURRENT PERS_LEGAL_NAME
./../GivenName	Given or chosen name. Also known as first name.	Elizabeth	SPRIDEN_FIRST_NAME
./../PreferredGivenName	Name by which the person prefers to be addressed. May be a name other than a given name, such as a nickname.	Liz	SPRIDEN_CURRENT PERS_PREF_FNAME
./../MiddleName	Middle name or initial.	Ann	SPRIDEN_MI
./../FamilyName	Non-chosen or inherited name. Also known as last name in the Western context.	Croton	SPRIDEN_LAST_NAME
./../Affix @type="formOfAddress"	Form of address for the person. Also known as the prefix.	Mrs.	SPRIDEN_PREFIX
./../Affix @type="qualification"	Name suffix, such as a professional designation or generational qualification.	MD	SPRIDEN_SUFFIX
./../Gender	Gender.	Female	SPBPERS_SEX
./../Birthdate	Denormalized birth date.		
./../BirthDay	Day portion of the birth date.	02	SPBPERS_BIRTH_DATE
./../BirthMonth	Month portion of the birth date.	02	SPBPERS_BIRTH_DATE

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./../BirthYear	Year portion of the birth date.	1970	SPBPERS_BIRTH_DATE
./TaxId	Tax identifier of the person.	555-99-6666	Published from SPBPERS_SSN based on GTVSDAX setting TAXID, which determines whether tax IDs are included or excluded in messages
./EmailAddress	Email address assigned by the institution.	lcroton@myinstitution.edu	Published from GOREMAL based on GTVSDAX setting EMAIL_CODE, which establishes the email type code that maps to EmailAddress. For inbound account provisioning to Banner, the code in the GTVSDAX setting is used to populate GOREMAL_EMAL_CODE.
./PrimaryAddress	Primary contact address. May or may not be the person's address while attending or working at the institution, and may or may not be the person's permanent contact address depending on institutional policies.		Published from SPRADDR based on GTVSDAX setting PRIM_ADDR_CODE, which establishes the address type code that maps to PrimaryAddress. For inbound account provisioning to Banner, the code in the GTVSDAX setting is used to populate SPRADDR_ATYP_CODE.
./@validFrom	Date when the address begins to be valid.	2008-01-12	SPRADDR_FROM_DATE
./@validTo	Date when the address is no longer valid.	2011-09-15	SPRADDR_TO_DATE

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./CountryCode	Country code, which can include the ISO 3166-1 two-character country code, the descriptive name of the country, or another accepted value.	157	Initially populated from SPRADDR_NATN_CODE. However, it can be translated to a valid enterprise value. In the default configuration, no translation of Banner values takes place.
./PostalCode	Code established by postal authorities for sorting and delivering mail.	89099	SPRADDR_ZIP
./Region	State or province portion of the address.	NE	SPRADDR_STAT_CODE
./Municipality	City, town, village, or hamlet portion of the address.	Omaha	SPRADDR_CITY
./AddressLine (repeatable)	One formatted address line with all pieces in their proper place. Includes all necessary punctuation. Cannot be easily parsed. Used for delivery by the postal service. Can contain the name or number of the building, house, and street.	12 Main Street Apartment 6d	SPRADDR_STREET_LINE1 SPRADDR_STREET_LINE2 SPRADDR_STREET_LINE3 SPRADDR_STREET_LINE4
./CampusAddress	Primary contact address while the person is attending or working at the institution.		Published from SPRADDR based on GTVSDAX setting CAMP_ADDR_CODE, which establishes the address type code that maps to CampusAddress. For inbound account provisioning to Banner, the code in the GTVSDAX setting is used to populate SPRADDR_ATYP_CODE.

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./@validFrom	Date when the address begins to be valid.	2008-01-12	SPRADDR_FROM_DATE
./@validTo	Date when the address is no longer valid.	2011-09-15	SPRADDR_TO_DATE
./CountryCode	Country code, which can include the ISO 3166-1 two-character country code, the descriptive name of the country, or another accepted value.	157	Initially populated from SPRADDR_NATN_CODE. However, it may be translated to a valid enterprise value. In the default configuration, no translation of Banner values takes place.
./PostalCode	Code established by postal authorities for sorting and delivering mail.	89099	SPRADDR_ZIP
./Region	State or province portion of the address.	NE	SPRADDR_STAT_CODE
./Municipality	City, town, village, or hamlet portion of the address.	Omaha	SPRADDR_CITY
./AddressLine (repeatable)	One formatted address line with all pieces in their proper place. Includes all necessary punctuation. Cannot be easily parsed. Used for delivery by the postal service. Can contain the name or number of the building, house, and street.	12 Main Street	SPRADDR_STREET_LINE1 SPRADDR_STREET_LINE2 SPRADDR_STREET_LINE3 SPRADDR_STREET_LINE4

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./CampusPhone	Primary contact phone while the person is attending or working at the institution.		Published from SPRTELE based on GTVSDAX setting CAMP_TELE_CODE, which establishes the phone type code that maps to CampusPhone. For inbound account provisioning to Banner, the code in the GTVSDAX setting is used to populate SPRTELE_TELE_CODE.
./InternationalCountryCode	ITU's country code that identifies a specific country, group of countries in an integrated numbering plan, or a specific geographic area.	44	SPRTELE_CTRY_CODE_PHONE
./AreaCityCode	Telephone area code. Defines either a numbering area within a country (or group of countries in an integrated numbering plan or a specific geographic area) or network/service.	803	SPRTELE_PHONE_AREA
./SubscriberNumber	"Local" phone number. Identifies a subscriber in a network or numbering area. *Must* contain at least one digit. May contain hyphens, periods, and spaces in addition to digits.	760-7890	SPRTELE_PHONE_NUMBER
./Extension	PBX extension, pager PIN, fax subaddress, or other extended addressing information.	416	SPRTELE_PHONE_EXT

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./MobilePhone	Mobile phone number.		Published from SPRTELE based on GTVSDAX setting MOBILE_TELE_CODE, which establishes the phone type code that maps to MobilePhone. For inbound account provisioning to Banner, the code in the GTVSDAX setting is used to populate SPRTELE_TELE_CODE.
./../InternationalCountryCode	ITU's country code that identifies a specific country, group of countries in an integrated numbering plan, or a specific geographic area.	44	SPRTELE_CTRY_CODE_PHONE
./../AreaCityCode	Telephone area code. Defines either a numbering area within a country (or group of countries in an integrated numbering plan or a specific geographic area) or network/service.	803	SPRTELE_PHONE_AREA
./../SubscriberNumber	"Local" phone number. Identifies a subscriber in a network or numbering area. *Must* contain at least one digit. May contain hyphens, periods, and spaces in addition to digits.	760-7890	SPRTELE_PHONE_NUMBER
./../Extension	PBX extension, pager PIN, fax subaddress, or other extended addressing information.	416	SPRTELE_PHONE_EXT

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./Fax	Fax number.		Published from SPRTELE based on GTVSDAX setting FAX_TELE_CODE, which establishes the phone type code that maps to Fax. For inbound account provisioning to Banner, the code in the GTVSDAX setting is used to populate SPRTELE_TELE_CODE.
./InternationalCountryCode	ITU's country code that identifies a specific country, group of countries in an integrated numbering plan, or a specific geographic area.	44	SPRTELE_CTRY_CODE_PHONE
./AreaCityCode	Telephone area code. Defines either a numbering area within a country (or group of countries in an integrated numbering plan or a specific geographic area) or network/service.	803	SPRTELE_PHONE_AREA
./SubscriberNumber	"Local" phone number. Identifies a subscriber in a network or numbering area. *Must* contain at least one digit. May contain hyphens, periods, and spaces in addition to digits.	760-7890	SPRTELE_PHONE_NUMBER
./Extension	PBX extension, pager PIN, fax subaddress, or other extended addressing information.	416	SPRTELE_PHONE_EXT
./InstitutionRoles	Relationships of the person to the institution.		

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./institutionrole (repeatable)	Relationship that the person has with the institution. Can be system-derived or otherwise assigned to a person to determine access rights.		See "InstitutionRoles element" on page 2-13 for details.
./../role	Text-based description of the relationship that the person has with the institution.	STUDENT FACULTY STAFF ALUMNI FRIENDS DEVELOPMENTOFFICER FINANCE PROSPECT APPLICANT INSTITUTIONACCEPT APPLICANTACCEPT BANNERINB BASICPERSON	The BASICPERSON role is created when a person does not have any other valid role. This role is replaced when the person acquires a valid role.
./../context	Context or source of the role. Allows for the use of the same role name with different semantics.	INTCOMP Campus Card	Only INTCOMP roles are published from Banner.
./../institution	Name or identifier of the institution with which the person has the identified relationship. Used in multi-institution configurations.	INST_A	
./../fromDate	Date when the role begins to be valid. Used only for the BANNERINB role.	2008-01-12	Allows external systems to provision BANNERINB roles with a valid date range to Banner Workflow.

XML Tag (Element or Attribute as XPATH Expression)	Definition	Example	System-specific Data Comments
./../toDate	Date when the role is no longer valid. Used only for the BANNERINB role.	2008-09-15	Allows external systems to provision BANNERINB roles with a valid date range to Banner Workflow.
./Extension	Structure used to extend the UDCIdentity XML structure to include other undefined attributes.		See "Extension element" on page 2-20 for details.
./../Attribute	Name-value pair of an identifying characteristic of the person.		
./../name	Identifying name of the attribute.	firstName lastName emailAddress Major	Name of the variable used in an SQL query defined by an institution using GORRSQL to extract a custom attribute. **Note:** Oracle Streams configuration rules must be defined on GUASADM to support custom attributes.
./../value	Value of the attribute.	Joe Jones jjones@somewhere.edu POLSCI	Value derived by an SQL query defined by an institution using GORRSQL for a custom attribute. When provisioning an INB user into Banner, two Attribute elements are required: • One with the `<name>` element set to *BANNERINB_USER* should have a `<value>` element containing the user ID to be created in Banner. • One with the `<name>` element set to *BANNERINB_PASSWORD* should have a `<value>` element containing the password to be created in Banner for the user.

UDCIdentifier element

The UDCIdentifier element is a globally unique identifier (GUID) that is assigned to each person record published in a UDCIdentity message. The UDCIdentifier is an unchanging, system-generated, 32-character, alphanumeric value. The following example shows the UDCIdentifier in the UDCIdentity XML structure:

```
<UDCIdentifier>36BE6D6D18560C44E0440003BA33B440</UDCIdentifier>
```

The UDCIdentifier is the primary identifier for each user that is synchronized with the EIMS or central directory, ensuring that user data objects are properly located, created, and updated. The UDCIdentifier also provides the basis of claims-based authentication.

InstitutionRoles element

The InstitutionRoles element contains one or more child elements that specify the relationships that a person has with the institution. Each relationship can be system-derived or assigned to a person to help determine access rights.

If Banner is the authoritative source for person data, it is responsible for publishing a person's relationships with the institution (that is, the person's "institution roles").

The following sections describe the supported roles, how rules are used to determine roles, how roles are assigned and maintained, and how roles are administered. For additional information about role publication and maintenance, refer to the *Banner Integration for eLearning Administration Guide*.

Supported institution roles

The InstitutionRoles element can be populated with the following *types* of institution roles:

- *Person* roles are defined for and used by Banner Intcomp, a collection of database objects for integration with Luminis® Platform and Banner Integration for eLearning. Examples of person roles include Student, Faculty, Staff, and Alumni.
- The *BannerINB* role is assigned to a user who has access to Internet-native Banner forms. This role is used specifically with BEIS.
- The *BasicPerson* role is assigned to a person who has none of the other defined roles.

Note

Custom roles that your institution creates are also sent in messages without the need for attribute extensions.

The InstitutionRoles element supports the following institution roles. The “Role” is the value that is published in a UDCIdentity message via messaging. The “Role Code” is the corresponding value that is delivered in the seed data.

Role	Role Code	Banner Product
Alumni	ALUMNI	Advancement
Applicant	APPLICANT	Student
ApplicantAccept	APPACCEPT	Student
BannerINB	BANNERINB	General
BasicPerson	BASICPERSON	Banner Identity Gateway
DevelopmentOfficer	DEVELOPMENTOFFICER	Advancement
Faculty	FACULTY	Student
Finance	FINANCE	Finance
Friends	FRIENDS	Advancement
InstitutionAccept	INTACCEPT	Student
Prospect	PROSPECT	Student
Staff	EMPLOYEE	Human Resources
Student	STUDENT	Student

Use of rules to determine institution roles

Roles in Banner are assigned to a person based on the existence of data in specific tables. For example, the existence of a PEBEMPL record with specific values could qualify a person for a Staff role; the existence of a SIBINST record could qualify a person for a Faculty role.

Banner uses rules to evaluate data in tables and determine whether a role should be assigned to a person. These rules are stored in the SQL Process Rules Table (GORRSQL) and maintained on the Business Rules Form (GORRSQL). Rules used to populate the InstitutionRoles element are defined on GORRSQL under two process codes:

`INTCOMP`	Defines rules for the institution person roles
`IAM`	Defines the BannerINB role

The following table describes the GORRSQL rules used to determine institution roles.

Note

GORRSQL does not define a rule for the BasicPerson role. This role is created in the UDCIdentity message when a person does not have a valid role as defined on GORRSQL. The role of BasicPerson is automatically removed when the person acquires a role that is defined on GORRSQL. ■

Role	Rule That Determines Whether This Role Is Assigned
Alumni	Person has a record in the Alumni Category Table (APRCATG), and the category represented by that record is “alumni.”
Applicant	Person has a record in the Admissions Application Table (SARADAP), and that record and any others for the same person, term, and level have no associated records in the Student Application Decision Table (SARAPPD) that indicate an institutional accept decision.
ApplicantAccept	Person has a record in the Student Application Decision Table (SARAPPD), and that record indicates an applicant accept decision.
BannerINB	Person has a record in the Enterprise Oracle Access Table (GOBEACC). This role supports users who have access to Banner, typically institution administrative personnel. The associated rule is used specifically for BEIS. SunGard Higher Education recommends that this rule not be modified.
DevelopmentOfficer	Person has a record in the Web Tailor User Roles Table (TWGRROLE), and the role represented by that record is “DevelopmentOfficer.”
Faculty	Person has a record in the Faculty Member Table (SIBINST), that record is effective for a term indicated by an `ACTIVE_TERM` record on the Integration Configuration Settings Form (GORICCR), and the person has an active faculty status. Banner Intcomp supports a second GORRSQL rule (delivered inactive) to limit the publication of faculty data. Refer to the *Banner Integration for eLearning Administration Guide* for detailed information. Best practice is not to use this rule when using BEIS.
Finance	Person has a record in the Enterprise Oracle Access Table (GOBEACC), and the person’s username matches the user ID on a record in the User Profile Validation Table (FOBPROF). Selecting the **Self Service Access** check box on the User Profile Maintenance Form (FOMPROF) creates or removes a person’s Finance role.
Friends	Person who has a record in the Alumni Category Table (APRCATG), and the category represented by that record is “alumni friend.”

Role	Rule That Determines Whether This Role Is Assigned
InstitutionAccept	Person has a record in the Admissions Application Table (SARADAP) and an associated record in the Student Application Decision Table (SARAPPD) that indicates an *institutional* accept decision. **Note:** This role is *not* assigned to a person who has a record in SARADAP, and that record or any others for the same person, term, and level has an associated record in SARAPPD that indicates an *applicant* accept decision.
Prospect	Person has a record in the Recruiting Table (SRBRECR), and there are no records in the Admissions Application Table (SARADAP) for the same person, term, and level.

Staff	Person has a record in the Employee Table (PEBEMPL), and the class code on that record is not specified in any `LDIEMPEX` record on the Integration Configuration Settings Form (GORICCR). In other words, a person is excluded from being assigned the Staff role if the employee's class code on PEBEMPL matches a `LDIEMPEX` record on GORICCR.
	
	To exclude terminated employees, you can modify the GORRSQL rule by adding the following line to check for a NULL or future termination date:
	`AND NVL (pebempl_term_date, SYSDATE) >= SYSDATE`
	
	Another option is to modify the GORRSQL rule by adding the following line to check for a termination date that is after a certain date:
	`AND NVL (pebempl_term_date, '01-JAN-2011') >= '01-JAN-2011'`

Student	Person has a record in the Student Table (SGBSTDN), that record is effective for a term indicated by an `ACTIVE_TERM` record on the Integration Configuration Settings Form (GORICCR), and the student status allows the student to register. **Note:** If you prefer to synchronize inactive and active students, you can modify the GORRSQL rule so that the select statement does not check the student status. The GORRSQL rule that defines the active student role is delivered as active. Banner Intcomp supports a second GORRSQL rule (delivered inactive) to limit the publication of student data. Refer to the *Banner Integration for eLearning Administration Guide* for detailed information. Best practice is not to use this rule when using BEIS.

Roles in the GORIROL table

Roles assigned to a person are stored in the Institution Role Table (GORIROL). This table contains one record for each assigned role. Changes to data in the GORIROL table stimulate the production of UDCIdentity messages.

Initial assignment of roles

Initially, the GORIROL table is populated by the Institution Role Maintenance Process (GURIROL), which is run via Banner Job Submission. GURIROL loops through all person records in the Banner database, executes the GORRSQL role definitions for each record, and adds role records to the GORIROL table as appropriate.

Realtime role updates

Once the GORIROL table is populated, the Banner Intcomp LDIPERSON event is used to update a person's role information. At runtime, when an LDIPERSON (Banner Intcomp) event is triggered, the `GB_INSTITUTION_ROLE.P_MAINTAIN_ROLE` API is called. This API looks at the state of the role before the event fired and compares it to the current state of the role. If there is a difference, the GORIROL table is updated to reflect the current state of the role. Any role change for a person produces a UDCIdentity message.

Note

There are no event triggers on the GORIROL table. Therefore, if you create a custom role, you must create the necessary triggers in the database so that they are included in any extract processing. ■

Whenever message production is stimulated, role information is extracted from the GORIROL table and used to populate the InstitutionRoles element. In addition, the GORRSQL rule governing the BannerINB role is executed to determine if this role should be included in InstitutionRoles. If neither step produces a role (for example, if a person only has a valid SPRIDEN record), the BasicPerson role is added to InstitutionRoles. The BasicPerson role is automatically removed when the person gets a valid institution role.

Role updates with GURIROL

The GURIROL process can be run at any time to update role data in the GORIROL table. The process *must* be run in the following situations:

- GURIROL must be run when Banner Intcomp is installed. Banner Intcomp uses person roles in the GORIROL table for integration with Luminis Platform and Banner Integration for eLearning.
- GURIROL must be run when you activate or inactivate a role rule on GORRSQL. Roles in the GORIROL table are added or removed, based on whether you are activating or inactivating the role rule on GORRSQL.

- GURIROL must be run when you modify the SQL logic for an active role rule on GORRSQL. Roles in the GORIROL table are added or removed, based on the new selection criteria in the modified role rule.
- Student and Faculty roles on GORRSQL are delivered with rules that uses the `ACTIVE_TERM` settings on the Integration Configuration Settings Form (GORICCR). GURIROL should be run whenever an `ACTIVE_TERM` setting on GORICCR is added or removed. Roles in the GORIROL table are added or removed, based on the addition or removal of the `ACTIVE_TERM` setting on GORICCR.
- If Banner Intcomp is not installed or if Banner Intcomp event triggers are not enabled, GURIROL must be run periodically to update the GORIROL table with current person roles.

Tip

Whenever GURIROL is run, you should subsequently run the LDIS Extract (ICGORLDI) for user objects and import the resulting XML extract into the external systems that integrate with Banner. This ensures that the role changes are communicated to those external systems en masse. ■

GURIROL provides two options to limit the number of person records processed:

- You can run the process with a predefined population selection (POPSEL).
- You can provide a specific role code (for example, FACULTY) to indicate that the process should only update the GORIROL table with changes to this role. This capability can be used to update the table when a new role is activated.

Both options are available on the Process Submission Controls Form (GJAPCTL) for GURIROL. Refer to the *Banner General User Guide* for more details on running GURIROL.

Role administration

GORRSQL rules that are delivered as seed data cannot be deleted or modified, except to inactivate them. However, you can override, inactivate, and activate roles, as described in the following sections.

Override role definitions

Use the following steps to override a delivered role rule.

1. Access the Business Rules Form (GORRSQL).
2. Enter the following data:

Process	*INTCOMP*
Rule	Rule code to be overridden

3. Go to the Rule Data block.
4. Clear the **Active** check box.
5. Save.
6. Go to the next record and insert a new rule with a higher sequence number. Make sure the **Active** check box is selected.
7. Save.

If there is more than one active select statement for a given process/rule (each with a unique sequence number), a person in Banner need only be selected by one of the select statements to be given that role.

Note

Custom institution rules are also supported. ■

Inactivate roles

If your institution does not want to use certain roles, you can inactivate them. For example, your institution might not want to include the Prospect role when person data is extracted or synchronized. GORRSQL rules that define roles are delivered as active. If you clear the **Active** check box for a role on GORRSQL, then the GURIROL process and the LDIPERSON (Banner Intcomp) event do not evaluate or insert the role into the GORIROL table.

Activate roles

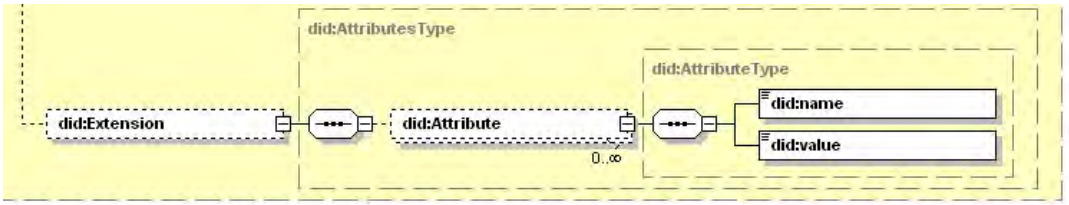
GORRSQL rules that define roles are delivered as active. However, your institution might not license the Banner products associated with certain roles. For example, your institution might not have Banner Human Resources installed, and therefore cannot generate LDIPERSON (Banner Intcomp) events that contain a Staff role. If a GORRSQL rule depends on a Banner product that is not installed at your institution, the rule is installed as inactive. If you subsequently add a Banner product that supports a rule previously made

inactive, you must reactivate the rule on GORRSQL. You can also reactivate a role that your institution previously did not need and was manually inactivated.

After activating a role rule, you should run GURIROL to re-seed the GORIROL table.

Extension element

The optional Extension element in the UDCIdentity XML structure allows institutions to add additional data elements to the base set of published information. The element can consist of any number of Attribute elements, each containing a Name element and a Value element:



To populate this structure, a system user (normally an administrator) creates a new attribute selection rule in Banner using the Business Rules Form (GORRSQL). The Streams Rules Configuration Form (GUASADM) also needs to be updated to capture the specific table changes.

When changes to data in Banner trigger publication of a new UDCIdentity message, the attribute selection rule fires and, if data is returned, the result is incorporated into the published message. The Name element contains the variable name used in the SQL statement of the attribute selection rule, while the Value element contains the actual data returned from the successful execution of the rule. This execution pattern is followed for all custom attribute selection rules defined by your institution.

For specific information on customizing the UDCIdentity XML structure to include new data elements, see "Customization of the UDCIdentity XML structure" on page 5-17.

The Extension element is also used in specific scenarios of inbound account provisioning to Banner. For example, when creating a Banner Internet-native Banner (INB) user, two Attribute elements are required as Extension element children. These Attribute elements provide Banner with a user ID and password for creating the Oracle user account and INB account.

Sample UDCIdentity operations

The following sections provide sample UDCIdentity operations. Where appropriate, multiple cases are shown to highlight specific operation scenarios. Additional operations are included to show the state of a UDCIdentity object after successful completion of an operation. For example, Lookup samples are included to show the state of the object following a successful Update operation.

Create operation

The following sample shows a UDCIdentity create operation, which is published upon creation of the user in Banner. Logic in the General Person Identification Form (SPAIDEN) results in the publication of a message containing minimal person data.

```
<UDCIdentity action="CREATE"
xmlns="urn:sungardhe:enterprise:domain:identity:1.0">
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
   <PersonIdentity>
      <PersonName>
         <GivenName>Elizabeth</GivenName>
         <FamilyName>Croton</FamilyName>
         <Affix type="formOfAddress">Dr.</Affix>
      </PersonName>
   </PersonIdentity>
   <InstitutionRoles>
      <institutionrole>
         <role>BASICPERSON</role>
      </institutionrole>
   </InstitutionRoles>
</UDCIdentity>
```

Query operation

If an application consumes and successfully processes the preceding create message, a query request directed at the application would result in the following reply message:

```
<UDCIdentity action="CREATE"
xmlns="urn:sungardhe:enterprise:domain:identity:1.0">
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
   <PersonIdentity>
```

```
      <PersonName>
         <FormattedName/>
         <GivenName>Elizabeth</GivenName>
         <FamilyName>Croton</FamilyName>
         <Affix type="formOfAddress">Dr.</Affix>
      </PersonName>
   </PersonIdentity>
   <InstitutionRoles>
      <institutionrole>
         <role>BASICPERSON</role>
      </institutionrole>
   </InstitutionRoles>
</UDCIdentity>
```

Update operation

Update operations are published with the UDCIdentity/@action attribute set to "Update." This message reflects the addition of data on the General Person Form (SPAPERS).

```
<UDCIdentity action="UPDATE"
xmlns="urn:sungardhe:enterprise:domain:identity:1.0">
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
   <PersonIdentity>
      <PersonName>
         <GivenName>Elizabeth</GivenName>
         <FamilyName>Croton</FamilyName>
         <Affix type="formOfAddress">Dr.</Affix>
      </PersonName>
      <Gender>Female</Gender>
      <Birthdate>
         <BirthDay>23</BirthDay>
         <BirthMonth>08</BirthMonth>
         <BirthYear>1977</BirthYear>
      </Birthdate>
   </PersonIdentity>
   <InstitutionRoles>
      <institutionrole>
         <role>BASICPERSON</role>
      </institutionrole>
   </InstitutionRoles>
</UDCIdentity>
```

Update messages are published with the full snapshot of the data object known at the time of publication. Messages are not published with only changed data. Consuming systems must understand that not all data contained in the UDCIdentity message changed. Applications are responsible for identifying the data that changed and updating the state of their corresponding entities accordingly.

The following example provides a case in which only the BirthDay of the person is modified. The published message, however, has all information pertaining to the person.

```
<UDCIdentity action="UPDATE"
xmlns="urn:sungardhe:enterprise:domain:identity:1.0">
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
   <PersonIdentity>
      <PersonName>
         <GivenName>Elizabeth</GivenName>
         <FamilyName>Croton</FamilyName>
         <Affix type="formOfAddress">Dr.</Affix>
      </PersonName>
      <Gender>Female</Gender>
      <Birthdate>
         <BirthDay>25</BirthDay>
         <BirthMonth>08</BirthMonth>
         <BirthYear>1977</BirthYear>
      </Birthdate>
   </PersonIdentity>
   <InstitutionRoles>
      <institutionrole>
         <role>BASICPERSON</role>
      </institutionrole>
   </InstitutionRoles>
</UDCIdentity>
```

Another form of an update message is deleting a child element of UDCIdentity. The absence of an attribute or element indicates that this data is no longer available for the user. Removed data can be simple attributes such as an address component, or a complex object itself, such as an address or telephone number. Removal of complex objects indicates that there no longer is an affiliation or association between the person entity and these subordinate entities. Applications are responsible for identifying changes to UDCIdentity and changing the state of application entities accordingly.

The following UDCIdentity object illustrates a more complex update.

```
<UDCIdentity action="UPDATE"
xmlns="urn:sungardhe:enterprise:domain:identity:1.0"
xmlns:null="urn:sungardhe:enterprise:domain:identity:1.0">
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
   <PersonIdentity>
      <PersonName>
         <GivenName>Elizabeth</GivenName>
         <FamilyName>Croton</FamilyName>
         <Affix type= "formOfAddress">Dr.</Affix>
      </PersonName>
      <Gender>Female</Gender>
      <Birthdate>
         <BirthDay>25</BirthDay>
         <BirthMonth>08</BirthMonth>
         <BirthYear>1977</BirthYear>
      </Birthdate>
   </PersonIdentity>
   <EmailAddress>ecroton@myinstitution.edu</EmailAddress>
   <PrimaryAddress validFrom="1992-12-12">
      <PostalCode>19108</PostalCode>
      <Region>PA</Region>
      <Municipality>Manyunk</Municipality>
      <AddressLine>349 Chestnut Avenue</AddressLine>
      <AddressLine>Apartment 16</AddressLine>
   </PrimaryAddress>
   <CampusAddress validFrom="1992-12-12">
      <PostalCode>19085</PostalCode>
      <Region>PA</Region>
      <Municipality>Philadelphia</Municipality>
      <AddressLine>2608 Locust Walk</AddressLine>
      <AddressLine>#3323</AddressLine>
   </CampusAddress>
   <InstitutionRoles>
      <institutionrole>
         <role>BASICPERSON</role>
      </institutionrole>
   </InstitutionRoles>
</UDCIdentity>
```

Upon removal of the address associated with the CampusAddress tag, the following XML message would be published.

```
<UDCIdentity action="UPDATE"
xmlns="urn:sungardhe:enterprise:domain:identity:1.0"
xmlns:null="urn:sungardhe:enterprise:domain:identity:1.0">
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
   <PersonIdentity>
      <PersonName>
         <GivenName>Elizabeth</GivenName>
         <FamilyName>Croton</FamilyName>
         <Affix type="formOfAddress">Dr.</Affix>
      </PersonName>
      <Gender>Female</Gender>
      <Birthdate>
         <BirthDay>25</BirthDay>
         <BirthMonth>08</BirthMonth>
         <BirthYear>1977</BirthYear>
      </Birthdate>
   </PersonIdentity>
   <EmailAddress>ecroton@myinstitution.edu</EmailAddress>
   <PrimaryAddress validFrom="1992-12-12">
      <PostalCode>19108</PostalCode>
      <Region>PA</Region>
      <Municipality>Manyunk</Municipality>
      <AddressLine>349 Chestnut Avenue</AddressLine>
      <AddressLine>Apartment 16</AddressLine>
   </PrimaryAddress>
   <InstitutionRoles>
      <institutionrole>
         <role>BASICPERSON</role>
      </institutionrole>
   </InstitutionRoles>
</UDCIdentity>
```

The absence of the CampusAddress complex element indicates the address is no longer associated with the user. A consuming application must update its person entity by deleting, disabling, or expiring the address so it is no longer part of the user profile.

Delete operation

Banner forms do not allow the deletion of person-related data. However, if data is deleted by other means, outbound provisioning can capture the deletion and publish a UDCIdentity message with the action attribute set to *DELETE*. For example, the following SQL*Plus statement can delete person-related data and produce an SPML outbound message:

```
delete from spriden where spriden_pidm = <PIDM> and
spriden_change_ind is null and spriden_entity_ind = ‘P’
```

The resulting SPML outbound message includes the UDCIdentifier only.

```
<UDCIdentity action=“DELETE”
xmlns=“urn:sungardhe:enterprise:domain:identity:1.0”>
   <UDCIdentifier>36BE6D6D18560C44E0440003BA33B440
   </UDCIdentifier>
</UDCIdentity>
```

Not all Provisioning Service Targets (PSTs) support the delete operation. Banner as a PST does not support delete requests received from the Banner Identity Gateway, acting as a Provisioning Service Provider (PSP). To deprovision accounts in Banner, an SPML modify request without a role can be sent to the Banner Identity Gateway. This request disables Banner forms access for the user. The Oracle account is not modified and can be used for other applications, but access to Banner forms is removed.

3 Components That Support Account Provisioning and SSO

The installation, configuration, and administration of Banner® Enterprise Identity Services (BEIS) depend on the overall configuration of your institution's identity and access management enterprise. Several components work together to facilitate the exchange of identity information (account provisioning) and claims-based authentication (single sign on). Some components are available within SunGard® Higher Education applications. Others are specific to BEIS and are only available if deployed within your institution.

For account provisioning, the required components and their corresponding responsibilities depend on whether Banner is authoritative for identity data (outbound account provisioning) or not authoritative (inbound account provisioning). The components for single sign on are the same, whether Banner is authoritative or not.

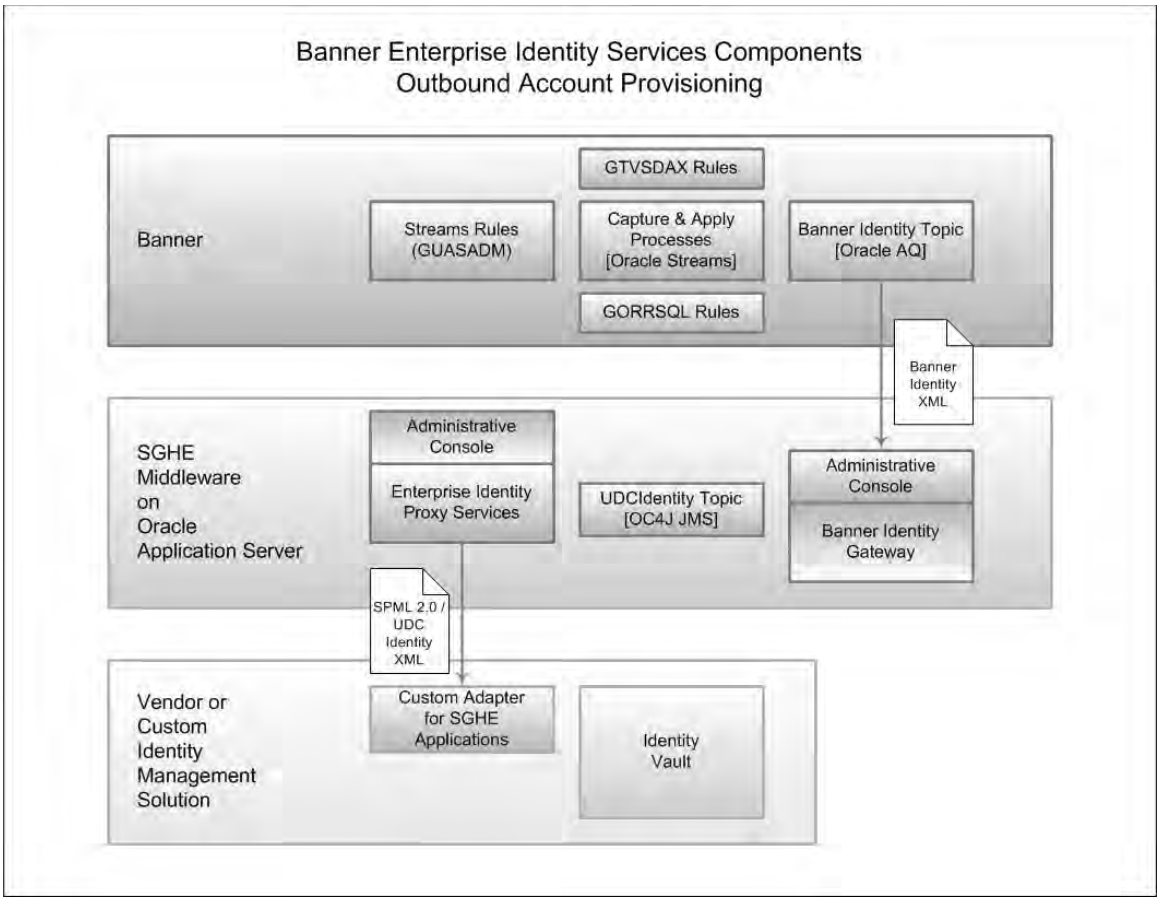
This chapter provides separate overviews of the components used in outbound account provisioning, inbound account provisioning, and single sign on. For each scenario, the required components are listed with their responsibilities, deployment platform, and relationship to other components. Other chapters of this handbook provide more details about some components.

Outbound account provisioning

"Outbound account provisioning" describes configurations in which Banner is the authoritative source for person information within the enterprise and, as a result, identity. Person information is created and modified in Banner and subsequently published as XML messages for consumption by other applications. This configuration allows for flexibility in what is published. There is also flexibility in the target applications, which can be one of the following:

- A compliant third-party enterprise identity management system (EIMS)
- An LDAP or active directory store
- Other SunGard® Higher Education applications

The following figure shows the components that are required for this configuration.

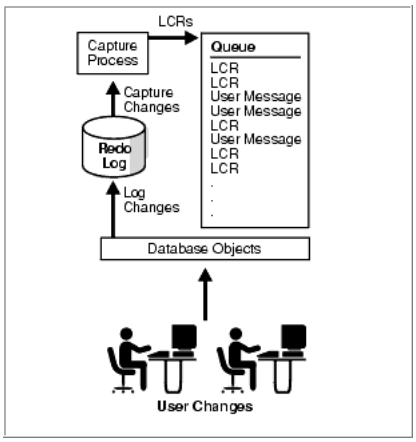


Oracle Streams

Oracle Streams, a feature of the Oracle database, is the engine for capturing and publishing changes to identity information in Banner:

- The capture process is an Oracle background process that reads the database redo log to capture changes made to database objects. The capture process is a rule-driven PL/SQL package that runs in the Banner database. Rules defining which database changes to capture and publish are defined in Banner using the Streams Rules Configuration Form (GUASADM).
- The apply process formats the captured changes into events called Logical Change Records (LCRs) and queues them for processing by the Banner Identity Gateway.

The following figure depicts the flow of data using Oracle Streams and the capture and apply processes.



Refer to Chapter 5, “Banner Configuration” for more details.

Streams Rules Configuration Form (GUASADM)

The Streams Rules Configuration Form (GUASADM) provides a central facility for administering Banner event streams. GUASADM can be used to modify or extend the rules that Oracle Streams uses to capture changes in the Banner database.

Refer to “Definition of rules to monitor database changes” on page 5-20 for details on GUASADM.

Crosswalk Validation Form (GTVSDAX)

GTVSDAX settings in Banner are used to customize specific parts of the UDCIdentity XML structure for your institution’s needs. Such customization is required because the UDCIdentity message contains specific tags that do not map directly to Banner data.

For example, Banner allows a person to have multiple address records. The UDCIdentity XML structure, however, includes only two tags for addresses: PermanentAddress and CampusAddress. You can use GTVSDAX rules to specify which addresses should be retrieved from Banner to populate these tags by. Similar GTVSDAX rules exist for email and telephone tags.

Refer to “Configure element content via GTVSDAX ” on page 5-13 for details on GTVSDAX.

Business Rules Form (GORRSQL)

While the UDCIdentity XML structure is fairly static in its composition and data content, it does provide an extension mechanism that allows you to publish additional data elements out of Banner. The Business Rules Form (GORRSQL) is used to define these additional data elements.

Refer to "Define the attribute selection rule" on page 5-18 for details on GORRSQL.

BannerIdentity Topic

When relevant data in Banner is created, updated, or deleted, the Oracle capture and apply processes publish a corresponding Banner Identity XML message to the BannerIdentity Topic. This JMS destination is implemented on Oracle Advanced Queuing (AQ), a component of the Oracle database. The Banner Identity Gateway subscribes to the BannerIdentity Topic and consumes Banner Identity XML messages for further processing.

UDCIdentity Topic

The UDCIdentity Topic provides a store-and-forward mechanism for UDCIdentity messages published by the Banner Identity Gateway. If OAS is used for deployment, the topic is configured in OC4J JMS (also known as OracleAS), which is a file and memory-based JMS provider in the OAS. If Oracle WebLogic is used for deployment, the topic is configured in JMS configurations provider in the Oracle WebLogic Server.

The Enterprise Identity Proxy Services application subscribes to this topic and consumes UDCIdentity messages for routing to a Provisioning Service Provider (PSP), which acts as an SPML front-end to one or more Provisioning Service Targets (PSTs) for provisioning actions.

BEIS components

Four middleware components are used to populate a central identity vault and synchronize identities between Banner and any SPML 2.0-enabled application that supports this product's SPML profile. These components are customized for each SunGard Higher Education application.

Identity Data Export Utilities

Identity Data Export Utilities populate a central identity vault from person-related data in the Banner database. The utilities are used to assign UDCIdentifiers to people in Banner, extract UDCIdentity information from the database, and bulk load data via LDIF or SPML batch provisioning. For more information, refer to Chapter 6, "Identity Data Export Utilities".

Banner Identity Gateway

The Banner Identity Gateway (Gateway) is a Java-based component that runs in the Oracle Application Server. The Gateway subscribes to the BannerIdentity Topic and performs the following operations:

- Invokes the Banner GUID service to generate a globally unique identifier (GUID) for the entity (if required)
- Transforms the Banner Identity XML message into a UDCIdentity message
- Publishes the UDCIdentity message to an SPML topic for consumption by other applications

The Gateway is configured to detect identity changes on Banner by reading Banner events from the Oracle AQ. This component provides a Web service to add and modify users in Banner.

The Gateway administrative interface provides the following administrative functions:

- Manage error logs created by the Gateway
- Administer the capture and apply processes
- Configure Gateway metadata

The Gateway serves as a host for several services that are required in the identity management environment. These services include a GUID service for creating a globally unique identifier and a lookup service that can cross-reference the UDCIdentifier and Banner-specific identifiers as required.

For more information, refer to Chapter 7, “Banner Identity Gateway”.

Enterprise Identity Proxy Services

The Enterprise Identity Proxy Services (Identity Proxy) application is the central component of BEIS. It is a Java-based JEE application that is deployed to the Oracle Application Server. It provides data exchange capabilities and an interface for administrative tasks.

The Identity Proxy acts as a conduit between SunGard Higher Education applications and a central identity vault using the industry-standard Service Provisioning Markup Language (SPML). For outbound account provisioning from Banner, the Identity Proxy acts as an SPML Request Authority (RA). In this capacity, it issues well-formed SPML requests to a compliant enterprise identity management system (EIMS) Provisioning Service Provider (PSP) whenever it consumes UDCIdentity messages from the UDCIdentity Topic. These SPML requests, which encapsulate the UDCIdentity XML structure, populate the identity vault with identity data as it is created in Banner and keep it in sync as subsequent data changes are made in Banner.

The Identity Proxy includes an administrative interface that is used to configure Provisioning Service Providers (PSPs). A PSP is an application that understands SPML and can update a Provisioning Service Target (PST). A PST can be the identity store of a third-party enterprise identity management system (EIMS), a central directory server, or, in a lightweight configuration, other non-Banner SunGard Higher Education applications.

Note

In many cases the distinction between a PSP and a PST is subtle. This is especially true where applications such as Luminis® Platform 4.x and Banner Workflow expose an SPML endpoint. As such, they act as a PSP, managing user accounts in their own data store, which is a PST. From the standpoint of SPML, however, and the Enterprise Identity Proxy Services component, they are referred to and treated as PSPs because of their understanding of SPML. ■

The administrative interface also includes an SPML provisioning message log.

For more information, refer to Chapter 8, "Enterprise Identity Proxy Services".

SPML LDAP Adapter

The SPML LDAP Adapter creates user accounts in an LDAP V3 compliant directory server. The adapter can be used with the SPML Publisher, delivered with the Identity Data Export Utilities, as one way to initially load accounts from Banner to an LDAP directory. A secondary use of the adapter is on-demand account provisioning from Banner. For more information, refer to Chapter 10, "SPML LDAP Adapter"

Identity management system

The final components that support outbound account provisioning are the components from your chosen enterprise identity management system (EIMS) or custom identity vault and adapter or gateway. These components serve as the hub of your institution's identity management infrastructure, receiving provisioning requests from the Identity Proxy, and, in turn, populating the identity vault and provisioning user accounts in other applications as necessary.

BEIS provides flexibility through the use of standard technologies such as XML, SPML, Java, Web services, and Service Oriented Architecture (SOA). Use of these technologies ensures a seamless integration with existing identity and access management systems. BEIS is a flexible framework that can be customized to meet the specific needs of your institution.

SunGard Higher Education has established relationships with several leading EIMS vendors. These vendors typically develop application-specific adapters through which their central identity repositories can be populated.

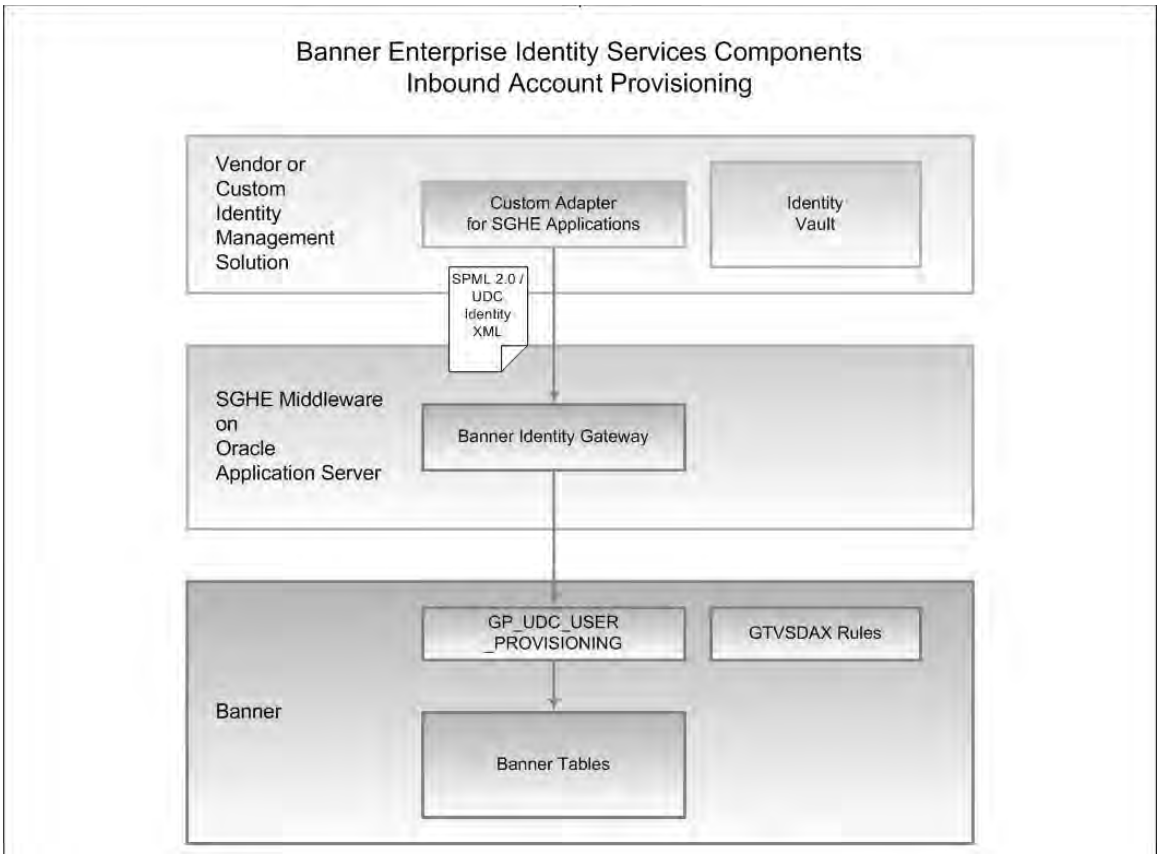
SunGard Higher Education Professional Services can help you leverage the power of BEIS within your campus environment. Specific services include the following:

- Installation and configuration
- Integration with the following third-party EIMs:
  - Oracle Identity Management
  - Novell Identity Manager
  - Microsoft Identity Integration Server
- Custom identity and access management integration

Inbound account provisioning

"Inbound account provisioning" describes configurations in which an application other than Banner is the authoritative source for person information within the enterprise. Person information is created and modified in the identity vault. User accounts in enterprise applications such as Banner are subsequently provisioned, based on rules established by the institution.

The following figure shows the components that are required for this configuration.



Identity management system

Your chosen enterprise identity management system (EIMS) or custom identity vault and adapter serve as the hub of your institution's identity management infrastructure. This component acts as a Request Authority (RA), requesting the creation or modification of data in Banner.

Banner Identity Gateway

The Gateway acts as a Provisioning Service Provider (PSP), receiving well-formed SPML 2.0 requests that contain the XML structure. The Gateway exposes Banner APIs as SPML 2.0-compliant Web services.

Banner APIs

Banner APIs are used to create or modify data in Banner, which serves as a Provisioning Service Target (PST).

Create requests

At a minimum, inbound account provisioning to Banner creates an identification record in the SPRIDEN table and a record of the entity's UDCIdentifier in the GOBUMAP table. If appropriate information is provided in the inbound request message, inbound provisioning also creates a record in the following tables:

- SPBPERS - specific person details
- SPRADDR - contact addresses
- SPRTELE - contact phone numbers
- GOREMAL - contact e-mail

Inbound account provisioning cannot create role information in Banner. A person's roles in Banner are determined by administrative processes such as establishing the person as a student, faculty member, or staff. These administrative processes require additional data that is outside the scope of account provisioning or the UDCIdentity XML structure.

The automatic creation of a third-party ID and PIN on the Third Party Access Table (GOBTPAC) is tied to the defined role (such as student or faculty). For this reason, the creation of third-party IDs and PINs is also outside the scope of inbound account provisioning.

An exception to this functionality is the ability to create a person with a specific role that allows access to Banner forms. To create a person record in Banner with default security privileges, two custom attributes in the extension element are required:

- One with the Name element set to *BANNERINB_USER* and a Value element containing the user ID to be created in Banner
- One with the Name element set to *BANNERINB_PASSWORD* and a Value element containing the password to be created in Banner for the user

Refer to "Extension element" on page 2-20 for more information.

Delete requests

Banner as a PST does not support the deletion of person data. Consequently, the Banner Identity Gateway, acting as the PSP for Banner, does not support SPML delete requests received from an SPML Request Authority (RA). In this situation, sending an SPML modify request to the Gateway without a role in the UDCIdentity message disables Banner forms access for the user. The Oracle account is not modified and can be used for other applications, but access to Banner forms is removed.

Single sign on

Single sign on (SSO) centralizes the process of authenticating the digital identity of a user. Once authenticated, a user can access multiple software systems without having to sign on to each system.

BEIS provides a single sign on gateway for Internet-native Banner (INB) and Self-Service Banner (SSB), allowing these applications to participate in a claims-based authentication environment. BEIS also provides services that other SunGard Higher Education applications can use to facilitate claims-based authentication based on the UDCIdentifier.

The following sections describe the components that are required for single sign on.

Central access manager

The central access manager for SSO can be the JA-SIG Central Authentication Service (CAS) or a third-party access manager.

If CAS is used as the central access manager, the CAS attribute assertion features facilitate single sign on. Attributes that identify the user are retrieved via a proprietary validation service, /bannerValidate, or via the CAS samlValidate service.

SSO Manager

The SSO Manager is the component of BEIS that serves as an SSO gateway for Banner. In this role, the SSO Manager proxies all requests for access to Self-Service Banner and Internet-native Banner. The SSO Manager exposes URLs that are used to access Banner. These URLs are protected by a central access manager. Once a user is authenticated, the SSO Manager performs necessary operations that allow the user to access Banner.

For example, the SSO Manager uses its Ticketing and Credential services to obtain the Oracle credentials that are required for a user to access Internet-native Banner. The SSO Manager also exposes these services as Web services for use by other applications that require the user's actual credentials for authentication.

Banner Web Tailor

This component applies to Self-Service Banner only. Banner Web Tailor accepts the identity assertion from the SSO Manager, determines the user based on the assertion, and creates a session for the user.

baniam.jar

This component applies to Internet-native Banner only. The baniam.jar contains Java components that communicate with the Credential Web service (exposed by the SSO Manager) to obtain the user credentials that are required to log a user in to Internet-native Banner. Refer to "Configure the Oracle Forms server for Internet-native Banner SSO" on page 11-74 for more information.

4 Implementation Road Maps

This chapter lists high-level tasks for installing and configuring Banner® Enterprise Identity Services (BEIS) for the following deployment scenarios:

- Outbound account provisioning, where Banner is authoritative for identity
- Inbound account provisioning, where Banner is not authoritative for identity
- Single sign on to Banner

Details for each task listed in this chapter are provided in subsequent chapters of this handbook.

Banner outbound account provisioning

If Banner is the authoritative source for person data at your institution, the creation and update of person data in Banner is the catalyst for provisioning identity in the enterprise. Changes to person data are published as XML messages and transferred to your institution's identity vault. These changes result in account provisioning to other applications.

As introduced in Chapter 3, "Components That Support Account Provisioning and SSO", several SunGard® Higher Education software components collaborate to transfer identity data from Banner to the identity vault. The following tasks are required to install and configure these components. Each task is described in more detail in the chapters that follow.

Note

If any of the following middleware components are already present, the associated installation step can be skipped.

1. Run a script that evaluates your Banner environment and reports on several criteria that are essential for a fully functioning BEIS environment. (Chapter 5, "Banner Configuration").
2. Configure Banner to publish data for account provisioning (Chapter 5, "Banner Configuration").
3. Install and configure the Identity Data Export Utilities (Chapter 6, "Identity Data Export Utilities").

4. Install and configure the Banner Identity Gateway (Chapter 7, “Banner Identity Gateway” if you are doing a manual installation, or Chapter 9, “Automated Installer” if you are doing an automated installation).

5. Install and configure the Enterprise Identity Proxy Services (Chapter 8, “Enterprise Identity Proxy Services” if you are doing a manual installation, or Chapter 9, “Automated Installer” if you are doing an automated installation).

6. Install the SPML LDAP Adapter, if you are integrating with an LDAP directory (Chapter 10, “SPML LDAP Adapter”).

7. Validate the installation and configuration (Chapter 12, “Middleware Validation”).

8. Identify and configure the identity management solution (LDAP/CAS) or third-party enterprise identity management system (EIMS).

9. Use Identity Data Export Utilities (Chapter 6, “Identity Data Export Utilities”) to assign UDCIdentifiers, extract UDCIdentity information, and populate the identity vault (LDAP or third-party EIMS). There are three ways to populate the vault:

   - Use the UDCIdentityList.xml extract natively.
   - Convert UDCIdentity information to LDIF and perform an LDIF import.
   - Use the SPML Publisher to generate SPML create request messages targeted at a specific endpoint.

Banner inbound account provisioning

If an application other than Banner is the authoritative source for person data at your institution, the creation and update of person data in that system triggers account provisioning in the enterprise. Changes to person data update your institution’s identity vault, which stimulates account provisioning in other applications.

To support this configuration, SunGard Higher Education applications expose SPML-based Web services that respond to provisioning requests. These services require the inclusion of the UDCIdentity message as their payload.

Inbound provisioning to Banner is supported through the collaboration of several software components. The following tasks are required to install and configure these components. Each task is described in more detail in the chapters that follow.

Note

If the following middleware components are already configured to support Banner inbound account provisioning, no further configuration is necessary.

1. Run a script that evaluates your Banner environment and reports on several criteria that are essential for a fully functioning BEIS environment. (Chapter 5, "Banner Configuration").

2. Configure Banner to support inbound account provisioning (Chapter 5, "Banner Configuration").

3. Install and configure the Banner Identity Gateway (Chapter 7, "Banner Identity Gateway" if you are doing a manual installation, or Chapter 9, "Automated Installer" if you are doing an automated installation).

4. Install and configure the Enterprise Identity Proxy Services (Chapter 8, "Enterprise Identity Proxy Services" if you are doing a manual installation, or Chapter 9, "Automated Installer" if you are doing an automated installation).

5. Validate the installation and configuration (Chapter 12, "Middleware Validation").

Single sign on to Banner

Whether Banner is the authoritative source for person data or not, single sign on to Banner is accomplished through the SSO Manager. As the SSO gateway for Self-Service Banner (SSB) and Internet-native Banner (INB), the SSO Manager performs the following services:

- Validates the SSO session and retrieves identity attributes from the central identity vault
- Starts the SSB session
- Retrieves user credentials for INB

The SSO Manager insulates Banner from varying authentication protocols.

The following chapters and documents provide details for implementing single sign on for Banner:

- Chapter 11, “SSO Manager” - Provides details for implementing the SSO Manager and its supporting components.
- Chapter 13, “Luminis Platform Configuration” - Provides details for configuring Luminis® Platform 4.x for SSO.
- Luminis Platform Integration Setup Guide - Provides details for configuring Luminis Platform 5.x for SSO.
- Chapter 14, “Banner Workflow Configuration” - Provides details for configuring Banner Workflow for SSO.
- Chapter 15, “Banner Document Management Suite Configuration” - Provides details for configuring Banner Document Management Suite for SSO.
- Appendix D, “CAS Installation and Configuration” - Provides installation and configuration details if you are using CAS as the central authentication server

5 Banner Configuration

Banner® must be configured properly before you implement Banner Enterprise Identity Services (BEIS). This chapter tells you how to run a script that evaluates your Banner environment for potential problems, configure the environment for BEIS, customize the UDCIdentity XML structure, and define custom rules for monitoring database changes.

Note

If your institution is implementing BEIS in a RAC environment, you must understand the best practices for implementing Oracle Streams in a RAC. Review Oracle Doc ID 413353.1 (What are the Best Practices For Oracle Streams in RAC Environment), available on the Oracle support Web site. If your institution contracts with SunGard® Higher Education for Oracle support, you can also review FAQ 1-7Q0BNU, available on the Customer Support Center.

BEIS_Check_script

The `BEIS_Check_script_.sql` script evaluates the Banner environment and reports on several criteria that are essential for a fully functioning BEIS environment. You can use the script before and during installation to quickly evaluate your environment and identify potential problems. You can use the script to evaluate both new installations and upgrades. Depending on the current state of your Banner/Oracle rdbms environment, you might need to make changes based on the output of the script.

Use the following steps to run the script, evaluate the results, and fix any problems.

1. Connect to the database as BANINST1 (primary node connection if RAC).
2. Increase the size of the output buffer by entering the following at an SQL prompt:

   ```
   exec DBMS_OUTPUT.ENABLE(1000000);
   ```

3. Run the script:

   ```
   BEIS_Check_script.sql
   ```

4. Review and evaluate the output:

Result	Evaluation
Database name	Verify that you are connected to the correct database.
Banner General version	Verify that the version is Banner General 8.3.1 or higher.
Banner Web Tailor version	Verify that the version is one of the following: • If Cascade is enabled, the version must be Banner Web Tailor 8.4.2 or higher. • If Cascade is not enabled, the version must be Banner Web Tailor 8.3.1 plus patch `p1-dc06g9_twb8030101,` or higher.
Banner Intcomp version	Verify that the version is Banner Intcomp 8.0.1 or higher. For information on the way BEIS uses Banner Intcomp, refer to "InstitutionRoles element" on page 2-13.
Archive log mode	Verify that the value is *Pass*. This indicates that the database is running in archive log mode. Oracle Streams must be operational in the database, and Oracle Streams requires the database to be in archive log mode.
Database initialization parameters	Verify that all parameters fall within the range of recommended values. For a list of recommended values, refer to Step 1, "Validate the Oracle initialization parameters and database configuration" on page 5-6.
DBA Apply Instantiated Objects	Verify that all DBA Apply Instantiated Objects belong to the database to which you are connected. The following message is displayed if an objects does not belong to the correct database: `Fail. Check DBA Apply Instantiated Objects Rows for Incorrect Database.`

Result	Evaluation
Registration of the identity_domain.xsd schema	Verify that the schema URL is shown (`http://xmlns.sungardhe.com/idm/schemas/identity_domain.xsd`). This indicates that the schema is registered in the database. The following message is displayed if the schema is not registered: `None Found. Fail.`
Banner_Identity_Topic	Verify that the values are *Exists* and *Yes*. This indicates that the topic is enabled for the ENQUEUE and DeQUEUE options.
Banner_Identity_Table	Verify that the value is *Exists*. This indicates that the underlying table for the `Banner_Identity_Topic`. exists.
IAM Process Code (GTVSQPR)	Verify that the value is *Pass*. This indicates that the IAM process code exists in GTVSQPR. For more information, refer to Step 3, "Populate the Business Rule Process Parameters Form (GORSQPA)" on page 5-8.
GORCTAB Data	Verify the number of tables that are being monitored for identity events. This number is *8* for a baseline configuration. This number is higher if additional tables are being monitored for extension attributes. For more information on the tables for a baseline configuration, refer to Step 4, "Validate Oracle Streams metadata" on page 5-9.
GORCCOL Data	Verify the number of columns that are being monitored for identity events. This number is *36* for a baseline configuration. This number is higher if additional columns are being monitored for extension attributes. For more information on the columns for a baseline configuration, refer to Step 4, "Validate Oracle Streams metadata" on page 5-9.

Result	Evaluation
GORCRUL Data	Verify the number of columns that are defined for filtering identity events. This number is *2* for a baseline configuration. This number is higher if additional column rules are defined. For more information on the column rules for a baseline configuration, refer to Step 4, "Validate Oracle Streams metadata" on page 5-9.
BEIS Configuration in GTVSDAX	Verify that all eight IDM settings exist on GTVSDAX. For those that have the *UPDATE ME* value, determine the appropriate values and enter changes on GTVSDAX. BEIS relies on settings in GTVSDAX for specific processing parameters and for data translation between Banner and the UDCIdentity XML structure. Seed data inserts the text *UPDATE ME* in the `GTVSDAX_EXTERNAL_CODE` field for some GTVSDAX settings. These default settings should be changed to meet your institution's requirements. For more information on the GTVSDAX settings used in BEIS, refer to the UDCIdentity XML structure mapping in Chapter 2, "UDCIdentity", "Crosswalk Validation Form (GTVSDAX)" on page 3-3, and Step 7, "Configure element content via GTVSDAX " on page 5-13.
GORRSQL rules for IAM - Extension Elements	Determine whether the GORRSQL rules that are associated with process code IAM are defined and active. The UDCIdentity XML structure can be extended by using GORRSQL rules that are associated with the process code IAM. For example, when populating an LDAP repository with user login credentials for CAS-based single sign on, GORRSQL rules can be used to extract and provide the required principal (username) and credential (password) attributes. For more information on GORRSQL rules, refer to the UDCIdentity XML structure mapping of the Extension element on page 2-12, Step 5, "Define attributes to be used as the LDAP username and password" on page 5-11, and "Customization of the UDCIdentity XML structure" on page 5-17.

Result	Evaluation
Active GORRSQL rules for INTCOMP - Roles	Determine whether the GORRSQL rules that your institution requires for Banner Intcomp roles are defined and active. The InstitutionRoles element of the UDCIdentity XML structure is populated with roles that are defined for Banner Intcomp on GORRSQL.
Address, phone, and email codes	Verify the total number of addresses, phones, and emails of specific Banner-defined types.
Streams-related views	Verify that all Streams-related views exist and that each view has a row related to IAM streams. Acceptable value is *Pass*.
Oracle Streams capture process	Verify that the status is *ENABLED*.
Oracle Streams apply process	Verify that the status is *IDLE*.

5. Once all results are evaluated against the expected results, connect as the STREAMSADMIN user and run the following command:

```
SQL>select * from session_privs;
```

6. Make sure that STREAMSADMIN has all required privileges.
7. Connect as the INTEGMGR user and run the following command:

```
SQL>select * from session_privs;
```

8. Make sure that INTEGMGR has all required privileges.
9. Connect as the BANSECR user and run the following command:

```
SQL>select * from session_privs;
```

10. Make sure that BANSECR has all required privileges.

Configuration of your environment

Use the following steps to configure your environment for BEIS:

- Step 1, “Validate the Oracle initialization parameters and database configuration”
- Step 2, “Verify the registration of identity domain XML schema”
- Step 3, “Populate the Business Rule Process Parameters Form (GORSQPA)”
- Step 4, “Validate Oracle Streams metadata”
- Step 5, “Define attributes to be used as the LDAP username and password”
- Step 6, “Create a population selection to select IDs to provision into LDAP”
- Step 7, “Configure element content via GTVSDAX ”
- Step 8, “Configure and start the capture process and apply process”
- Step 9, “Validate Oracle user accounts for fully privileged INB access”

Step 1 Validate the Oracle initialization parameters and database configuration

Oracle Streams uses two processes to capture and apply data changes that are associated with Banner identity attributes:

- The capture process monitors the Banner database for changes. Changes that meet the criteria of the capture rules defined on the Streams Rules Configuration Form (GUASADM) are converted to messages and posted in a queue.
- The apply process reads the messages that are posted by the capture process, generates Banner Identity XML messages, and calls a PL/SQL package (handler) that processes the captured changes.

You must adjust the Banner settings that the Oracle Streams capture and apply processes use to populate specific elements in the UDCIdentity XML structure. Not doing so limits the data contained in the UDCIdentity XML structure. SunGard Higher Education recommends that the Oracle DBA set the following Oracle initialization parameters:

Parameter	Configured Value
`compatible`	Database version: *10.2.0* or *11.1.0.0.0*
`global_names`	*true*
`job_queue_processes`	Maximum number of `dbms_job` processes. Increase the value by 4.

Parameter	Configured Value
streams_pool_size	**If Oracle initialization parameter SGA_TARGET is set:** Do not specify the streams_pool_size parameter. Oracle auto tunes the memory based on the SGA_TARGET parameter. **If Oracle initialization parameter SGA_TARGET is not set:** Specify the streams_pool_size based on the following recommendations. This parameter specifies the size of the streams pool, in bytes. The streams pool contains captured events and is part of the System Global Area (SGA). If this parameter is set, Oracle assigns a new pool of memory within the SGA that is dedicated to the use of Oracle Streams buffers. If this parameter is greater than 0, any SGA memory used by Oracle Streams is allocated from the streams pool. If this parameter equals zero, the SGA memory used by Oracle Streams is allocated from the shared pool and can use up to 10% of shared pool. Recommendations from Oracle are: • 10MB for *each* capture process parallelism • 1MB for *each* apply process parallelism • 10MB or more for *each* queue that stages captured events Insufficient memory can cause contention.
timed_statistics	Set to *true* to collect elapsed time statistics in dynamic performance views related to Oracle Streams.
shared_pool_size	Increase by 20MB.
parallel_max_servers	Derived from the following values: • CPU_COUNT • PARALLEL_AUTOMATIC_TUNING • PARALLEL_ADAPTIVE_MULTI_USER Refer to Oracle documentation for details.
sga_max_size	If you run multiple capture processes on a single database, consider using this setting to increase the size of the System SGA for each instance.

Step 2 Verify the registration of identity_domain XML schema

Use the following steps to verify that the `identity_domain` XML schema is registered to Oracle XDB.

1. Run SQL*Plus and connect as `BANINST1`.
2. Execute the following query:

```
select schema_url from user_xml_schemas where schema_url like
'%identity_domain.xsd';
```

3. One of the following occurs, depending on whether the schema is registered:
   - 3.1. If the schema is registered, one row is returned with the following data:

```
SCHEMA_URL
-----------------------------------------------------------
http://xmlns.sungardhe.com/idm/schemas/identity_domain.xsd
```

     You can continue with the next step of the installation.

   - 3.2. If the schema is not registered, the following message is displayed:

```
no rows selected
```

     Execute the `geniam.sql` script from the correct Banner General directory.

Step 3 Populate the Business Rule Process Parameters Form (GORSQPA)

The GORRSQL criteria for identity management are bound to a PIDM. All GORRSQL criteria that are supplied by seed data are system required and do not need the SQL bind variable PIDM to be declared. To add a new GORRSQL statement, you must declare the PIDM as a bind variable using the Business Rule Process Parameters Form (GORSQPA). Use the following steps to declare PIDM as a bind variable on GORSQPA.

1. Access the Business Rule Process Parameters Form (GORSQPA).
2. Enter *IAM* (IAM Process Code) in the **Process Code** field.
3. Navigate to the next block and enter *PIDM* in the **Parameter Code** field.
4. Save.

Note

The capture and apply processes produce a Banner Identity XML message, which the Banner Identity Gateway transforms into the UDCIdentity message. The mapping between these two structures is predefined and static. Instructions for customizing the XML produced by Banner are given in relation to the UDCIdentity XML structure, the primary means for transferring person attributes between systems.

Step 4 Validate Oracle Streams metadata

Oracle Streams uses metadata in the GORCTAB, GORCCOL, and GORCRUL tables to monitor the Banner database for changes. Seed data scripts for these tables were previously delivered, so the BEIS metadata may already exist in Banner. Use the following steps to validate and correct, if necessary, the BEIS metadata.

1. Use one of the following methods to validate the Oracle Streams metadata:

   - Run `BEIS_Check_script_.sql` and review the counts for the GORCTAB, GORCCOL, and GORCRUL tables. Refer to "BEIS_Check_script" on page 5-1 for information about running the script.
   - Use the Streams Rules Configuration Form (GUASADM) to validate the rules for the IAM process code. Refer to "Definition of rules to monitor database changes" on page 5-20 for information about the form.
   - Execute an SQL query.

   For a baseline configuration, the following metadata must be loaded:

Table	Data Loaded for Baseline Configuration	
GORCTAB	SPRIDEN	SPBPERS
	SPRADDR	GORIROL
	SPRTELE	GOBEACC
	GOREMAL	GOBTPAC

Table	Data Loaded for Baseline Configuration	
GORCCOL	SPRIDEN_CHANGE_IND	SPRTELE_PHONE_NUMBER
	SPRIDEN_ENTITY_IND	GOREMAL_EMAIL_ADDRESS
	SPRIDEN_FIRST_NAME	GOREMAL_EMAL_CODE
	SPRIDEN_LAST_NAME	GOREMAL_PREFERRED_IND
	SPRIDEN_MI	GOREMAL_STATUS_IND
	SPRADDR_ATYP_CODE	SPBPERS_BIRTH_DATE
	SPRADDR_CITY	SPBPERS_LEGAL_NAME
	SPRADDR_NATN_CODE	SPBPERS_NAME_PREFIX
	SPRADDR_STATUS_IND	SPBPERS_NAME_SUFFIX
	SPRADDR_STAT_CODE	SPBPERS_PREF_FIRST_NAME
	SPRADDR_STREET_LINE1	SPBPERS_SEX
	SPRADDR_STREET_LINE2	SPBPERS_SSN
	SPRADDR_STREET_LINE3	GORIROL_ROLE
	SPRADDR_STREET_LINE4	GORIROL_ROLE_GROUP
	SPRADDR_ZIP	GOBEACC_PIDM
	SPRTELE_CTRY_CODE_PHONE	GOBEACC_USERNAME
	SPRTELE_PHONE_AREA	GOBTPAC_EXTERNAL_USER
	SPRTELE_PHONE_EXT	GOBTPAC_PIN
GORCRUL	SPRIDEN_CHANGE_IND	
	SPRIDEN_ENTITY_IND	

2. If there is no metadata in the tables, use the following steps to load the baseline metadata for the UDCIdentity XML structure:

   2.1. Connect to the database as baninst1.

   2.2. Run the `load_iam_streams_meta_data.sql` script.

   The script depends on the following files: `gorccoli_iam_meta_data.sql`, `gorcruli_iam_meta_data.sql`, and `gorctabi_iam_meta_data.sql`. The script and its dependent files are located in the `\db-scripts\scripts` folder of the Banner Identity Gateway zip file.

3. If some of metadata exists but is inconsistent with the baseline configuration, use the following steps to delete the existing metadata and reload the baseline metadata for the UDCIdentity XML structure.

   3.1. Connect to the database as `BANINST1`.

   3.2. Run the `repair_iam_streams_meta_data.sql` script.

   The script depends on the following files:

   ```
   gorccoli_iam_meta_data.sql
   gorcruli_iam_meta_data.sql
   gorctabi_iam_meta_data.sql
   ```

   The script and its dependent files are located in the `\db-scripts\scripts` folder of the Banner Identity Gateway zip file.

   3.3. Follow the prompts to delete and restore the default IAM Streams metadata.

Step 5 Define attributes to be used as the LDAP username and password

Note

Use this step only if you are integrating with an LDAP directory. ■

Data in Banner can be included in the UDCIdentity message for use as the LDAP enterprise username and password. Because no specific elements exist in the UDCIdentity message for this additional data, it is included in the UDCIdentity Extension element.

Introduced in "Extension element" on page 2-20, the child Extension element is used to publish data elements other than the base set of attributes. The element can include any number of Attribute elements, each containing a name element and a value element. The name element contains text that defines what the attribute is (for example, "username"), and the value element contains the data queried from Banner for the person (for example, "jsmith").

The following fields on the Third Party Access Audit Form (GOATPAD) capture data that can be used for the LDAP username and password:

- **Third Party ID** - LDAP username (principal)
- **PIN** - password (credential)

These data items are stored on the GOBTPAC table.

To include a new Extension/Attribute element, you must define an attribute selection rule using the GTVSQRU and GORRSQL forms. Use the following steps to create rules for "username" and "password" attributes using data from the GOBTPAC table.

1. Access the Business Rule Code Validation Form (GTVSQRU).

2. Enter the following data for the two attribute selection rules:

   2.1. Username Attribute Selection Rule

Code	`IAM_USERNAME_RULE`
Description	Username for UDCIdentity provisioning to LDAP
System Required	Unchecked
Start Date	Start date (default is current date)

   2.2. Password Attribute Selection Rule

Code	`IAM_PASSWORD_RULE`
Description	Password for UDCIdentity provisioning to LDAP
System Required	Unchecked
Start Date	Start date (default is current date)

3. Save.

4. Access the Business Rules Form (GORRSQL).

5. Enter the following data in the key block:

Process	`IAM`
Rule	`IAM_USERNAME_RULE`

6. In the Rule Data block, enter the following SQL SELECT statement to extract the GOBTPAC third party ID field to be included as the "username" (PRINCIPAL) attribute in UDCIdentity messages.

   ```
   SELECT GOBTPAC_EXTERNAL_USER PRINCIPAL FROM GOBTPAC WHERE
   GOBTPAC_PIDM = :PIDM
   ```

7. Enter the date when the rule is to be valid in the **Start Date** field.

8. Click **Validate** to validate the SQL statement.

9. Select the **Active** check box to activate the rule.

10. Save.

11. Return to the key block.

12. Enter the following data:

Process	IAM
Rule	IAM_PASSWORD_RULE

13. In the Rule Data block, enter the following SQL SELECT statement to extract the PIN field to be included as the “password” (CREDENTIAL) attribute in UDCIdentity messages:

```
SELECT GB_THIRD_PARTY_ACCESS.F_EXTERNAL_PIN( :PIDM ) CREDENTIAL
FROM DUAL
```

Note

The Banner GOBTPAC_PIN value is stored in the SSHA-1 storage format requiring retrieval via an API. ■

14. Enter the date when the rule is to be valid in the **Start Date** field.

15. Click **Validate** to validate the SQL statement.

16. Select the **Active** check box to activate the rule.

17. Save.

Step 6 Create a population selection to select IDs to provision into LDAP

Note

This step is used only if you are integrating with an LDAP directory. ■

To ensure that the LDAP identity vault is only populated with identities that have all required information (specifically, Principal and Credential data), create a population selection in Banner to select only those IDs. The Identity Data Export Utilities application uses this population selection to extract Banner identities.

Refer to the *Banner General User Guide* for information on creating population selections.

Step 7 Configure element content via GTVSDAX

Persons in Banner can have an unlimited number of email addresses, physical addresses, and telephone numbers. The UDCIdentity XML structure allows for a limited number of email addresses, physical addresses, and telephone numbers. Consequently, institutions must select the email address, address, and telephone types in Banner to use for the defined child elements of the UDCIdentity XML structure.

Your institution must establish the content of the following elements of the UDCIdentity XML structure:

- EmailAddress
- PrimaryAddress
- CampusAddress
- CampusPhone
- MobilePhone
- Fax

To populate the UDCIdentity XML structure, the capture and apply processes retrieve values for address, telephone, and email address type codes from the Banner GTVSDAX table. A GTVSDAX rule exists for each of the preceding elements. When properly configured, each rule specifies the appropriate type code (that is, the appropriate address, telephone, or email address type) to use when retrieving data from Banner and populating the corresponding element.

Example

A GTVSDAX rule specifies that the address type code *CA* should be used to populate the CampusAddress tag. When data associated with a person in Banner is changed, the capture and apply processes are invoked and query the SPRADDR table for an address of type *CA* which, if returned, is used by the processes to populate CampusAddress.

The following table describes the GTVSDAX settings that control the population of the UDCIdentity XML structure.

Warning

Carefully determine what values are most appropriate for your institution's business needs. Although Banner supports multiple GTVSDAX settings with different values, only one type of each is produced in the Banner Identity XML message. Use the following table to understand the settings and exactly how and where they must be set to meet your operational expectations.

Note

GTVSDAX rules use an internal process code of IDM, short for Identity Management. Other forms, such as GORRSQL and GUASADM, use the code IAM, short for Identity and Access Management. While these terms differ slightly, there is no significant difference to internal Banner processing. Be careful to use the appropriate code for the specific context.

Setting	What It Controls	How It Is Set
Campus Address	Address that is defined as a person's campus address	Internal Code: *IDM* Internal Group: *CAMP_ADDR_CODE* Translation Code: null External Code: Value defined on Address Type Code Validation Form (STVATYP)
Campus Telephone	Telephone number that is defined as a person's campus telephone	Internal Code: *IDM* Internal Group: *CAMP_TELE_CODE* Translation Code: null External Code: Value defined on Telephone Type Validation Form (STVTELE)
Common Matching Source	Common Matching rules that are used for inbound account provisioning of person data	Internal Code: *IDM* Internal Group: *CM_SOURCE_CODE* Translation Code: null External Code: Value defined on Common Matching Source Code Validation Form (GTVCMSC)
Email Code	Email address that is defined as a person's email address	Internal Code: *IDM* Internal Group: *EMAIL_CODE* Translation Code: null External Code: Value defined on E-mail Address Type Validation Form (GTVEMAL)
Fax Number	Telephone number that is defined as a person's fax telephone	Internal Code: *IDM* Internal Group: *FAX_TELE_CODE* Translation Code: null External Code: Value defined on Telephone Type Validation Form (STVTELE)

Setting	What It Controls	How It Is Set
Mobile Number	Telephone number that is defined as a person's mobile telephone	Internal Code: *IDM* Internal Group: *MOBILE_TELE_CODE* Translation Code: null External Code: Value defined on Telephone Type Validation Form (STVTELE)
Primary Address	Address that is defined as a person's primary address	Internal Code: *IDM* Internal Group: *PRIM_ADDR_CODE* Translation Code: null External Code: Value defined on Address Type Code Validation Form (STVATYP)

Step 8 Configure and start the capture process and apply process

Use the following steps to configure and start Oracle Streams capture and apply processes.

1. Run SQL*Plus and connect as the `streamsadmin` Oracle user.
2. Execute `exec gp_streams_util.p_create_streams('IAM');`

   This step accomplishes the following:

   - Creates buffered queues and queue tables.
   - Configures the DML callback handler for the apply process.
   - Sets the instantiation SCN for the tables in the apply process.
   - Starts the Oracle Streams capture and apply processes.
3. Execute `exec gp_streams_util.p_configure_rules('IAM');`

   This step accomplishes the following:

   - Stops the Oracle Streams capture and apply processes.
   - Creates supplemental, primary key, and unique key log groups for the configured tables.
   - Generates the streams rules defined from the meta data on GUASADM.
   - Re-starts the Oracle Streams capture and apply processes.

Refer to Appendix A, "Oracle Streams" for information about Oracle Streams administration.

Step 9 Validate Oracle user accounts for fully privileged INB access

Note

This step is required only to support inbound account provisioning to Banner.

Inbound account provisioning to Banner can create two kinds of Banner users:

- A Banner administrative user, who has access to Banner Self-Service pages and Banner forms. This user has an Oracle user account to access Internet-native Banner (INB).
- A Banner person, such as a student or faculty member, who does not have access to Banner forms, but may be allowed to access Banner Self-Services pages.

Note

While inbound provisioning can create a person, it cannot create the person's role(s). Roles must be created in Banner via specific processes, such as student matriculation.

A default user profile is used to create a new INB (Oracle) user. The `'IAM_TEMPLATE'` user has minimal Banner privileges (`ban_default_connect`). Use the Security Maintenance Form (GSASECR) to validate the `'IAM_TEMPLATE'` user so that new Oracle user accounts created for INB access have the following:

- Correct table space
- Required Banner privileges
- Required Oracle privileges

Customization of the UDCIdentity XML structure

As introduced in “Extension element” on page 2-20, the content of the UDCIdentity XML structure can be customized to include additional data from Banner. This capability allows your institution to provision additional identity attributes to your institution’s enterprise identity management system (EIMS).

Populating the Extension element is based on attribute selection rules defined by your institution on the Business Rules Form (GORRSQL). When changes to Banner data trigger publication of a new UDCIdentity message, the capture and apply processes invoke any existing attribute selection rules. If a rule returns data, the data is added as a child Attribute element of Extension in the published message. The name element of the Attribute tag contains the variable name used in the SQL statement of the attribute selection rule, and the value element contains the actual data returned from the successful execution of the rule. This execution pattern is followed for all custom attribute selection rules defined by your institution.

You can define an attribute selection rule to get additional data for inclusion in the UDCIdentity element.An attribute selection rule does not cause the Oracle Streams processes to monitor and capture changes to this data. To trigger the publication of a UDCIdentity message, you must define a corresponding capture rule using the GUASADM form. Then you must enable the capture rule. The creation of extension attributes and capture rules are independent of each other. You can create either or both.

Use the following steps to include optional, custom identity attributes in the UDCIdentity XML structure:

- Step 1, "Define the attribute selection rule"
- Step 2, "Define the attribute change capture rule"
- Step 3, "Enable the capture rule"

The example of adding an attribute for a student's participation in ROTC, defined as an activity in Banner, is used to illustrate the process.

Step 1 Define the attribute selection rule

Use the following steps to create a rule defining the data that needs to be retrieved from Banner for publication.

1. Create the rule code on the Business Rule Code Validation Form (GTVSQRU).

   Example

   For a rule that extracts ACT data, create a rule code entitled *IAM_SGRSACT_RULE*.

2. Access the Business Rules Form (GORRSQL).

3. In the key block, enter the following data:

   Process *IAM* (required for the capture and apply processes to invoke the rule)
   Rule Name of the rule created in step 1

4. In the Rule Data block, enter an SQL SELECT statement that will select the data you want to create as an attribute in the UDCIdentity XML structure.

   Example

```
SELECT sgrsact_actc_code from sgrsact
where sgrsact_pidm = :pidm
and sgrsact_actc_code = '190'
```

5. Enter the date when the rule is to be valid in the **Start Date** field.

   Example

   21-AUG-2008

6. Click the **Validate** button to validate the SQL statement.

7. Select the **Active** check box to activate the rule.

8. Save.

Step 2 Define the attribute change capture rule

Use the following steps to modify the Oracle Streams configuration so that changes to Banner data are monitored to stimulate the capture and apply processes. Data capture rules are defined using the Streams Rules Configuration Form (GUASADM). See "Definition of rules to monitor database changes" on page 5-20 for more details about using this form.

1. Access the Streams Rules Configuration Form (GUASADM).

2. In the key block, enter *IAM* in the **Capture Name** field.

3. In the Capture Tables block, enter the name of the table to be monitored and the table owner.

   Example

   Table Name *SGRSACT*
   Table Owner *SATURN*

4. Enter *GP_IDENT_HANDLER.P_HANDLE_IAM_EVENT* in the **Apply Handler** field.

5. In the Capture Columns block, enter the name of the column to be monitored.

   Example

   SGRSACT_ACTC_CODE

Note

More granular rules can be defined in the Capture Rules block. These rules must evaluate to TRUE for a database change to be captured and written to the capture process' queue.

6. Save.

Step 3 Enable the capture rule

After change capture rules are defined, they must be enabled in the capture and apply processes. This is done through the administrative interface of the Banner Identity Gateway, which is deployed to and runs in the Oracle Application Server. Use the following steps to enable the new capture rule.

1. Access the URL for the Banner Identity Gateway administrative interface:

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

2. Click **Streams Admin**.

3. Select *Stop Capture and Apply* in the **Banner Streams Ops.** field.

4. Click **Apply**. The capture and apply processes stop.

5. Select *Load Capture Rules* in the **Banner Streams Ops.** field.

6. Click **Apply**. The new capture rule is loaded.

7. Select *Start Capture and Apply* in the **Banner Streams Ops.** field.

8. Click **Apply**. The capture and apply processes restart.

9. Execute a test case that modifies the Banner data that is to be captured.

10. Check the resulting UDCIdentity message produced by the Banner Identity Gateway.

In the case of the example provided, an activity code of 190, ROTC would be added to a student's record in Banner. The UDCIdentity message produced as a result of this update should include the following in the Extension element.

Example

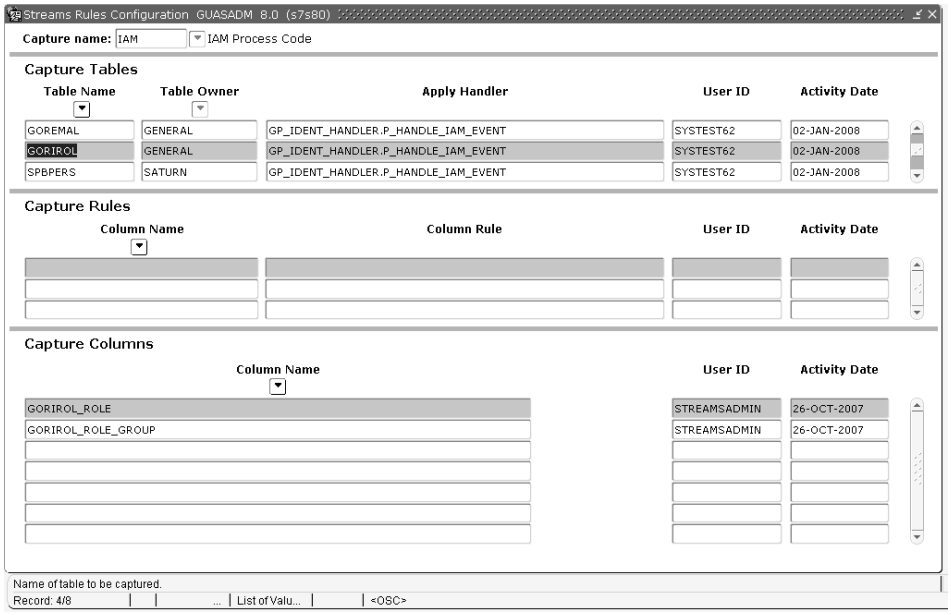
```
<Attribute>
          <name>SGRSACT_ACTC_CODE</name>
          <value>190</value>
</Attribute>
```

Definition of rules to monitor database changes

The Streams Rules Configuration Form (GUASADM) defines the rules that Oracle Streams uses to monitor changes in the Banner database. GUASADM is delivered with

several rules that monitor changes associated with Banner identity information (for example, name, address, and email address).

You can define additional rules for monitoring changes in your Banner database. A rule can evaluate a specific column condition and capture changes if the condition is *true*. For example, you can define a rule that captures address changes only if the addresses are active and have a specific address type.



GUASADM - Key block

This block provides a name for the set of rules used to capture database changes.

Field	Description
Capture Name	Process code that identifies the set of capture rules. Code *IAM* is used to identify rules for capturing Banner identity information. List Capture Processes (GTVSQPR)

GUASADM - Capture Tables block

This block identifies the tables that Oracle Streams monitors for database changes. The capture of specific table changes depends on the corresponding rules defined in the Capture Rules and Capture Columns blocks.

Field	Description
Table Name	Table that Oracle Streams monitors for changes. Any Banner table can be monitored. List — Table Names (all tables in Banner database)
Table Owner	User ID that owns the table. If there is a single table for **Table Name,** the **Table Owner** defaults. If there are multiple tables with the same **Table Name** but with different owners, select the correct owner from the drop-down list.
Apply Handler	Command that runs the capture in the background. The following command is used to capture changes for Banner identity information: `GP_IDENT_HANDLER.P_HANDLE_IAM_EVENT`
User ID	User ID that created the rule. Display only.
Activity Date	Date when the rule was created. Display only.

GUASADM - Capture Rules block

This block defines the rules that Oracle Streams uses to evaluate columns for specific conditions. If a condition is evaluated to be *true*, the columns in the Capture Columns block are monitored for changes.

Field	Description
Column Name	Column that is evaluated for a specific condition. You can evaluate any column in the table that is selected in the Capture Tables block. List — Table Columns (all columns in the table that is selected in the Capture Tables block)
Column Rule	Column condition to be evaluated. If evaluated to be *true*, columns in the Capture Columns block are monitored for changes. ***Example 1*** **Column Name** = *SPRIDEN_ENTITY_IND* **Column Rule** = *IN ('P')* If the value of the `SPRIDEN_ENTITY_IND` column is *P* (person), then the columns in the Capture Columns block are monitored for changes.

Field	Description
	Example 2 **Column Name** = *GOREMAL_PREFFERED_IND* **Column Rule** = *IN ('Y')* If the value of the `GOREMAL_PREFERRED_IND` column is *Y* (email address is marked as preferred), then the columns in the Capture Columns block are monitored for changes
User ID	User ID that created the rule. Display only.
Activity Date	Date when the rule was created. Display only.

GUASADM - Capture Columns block

This block identifies the columns that Oracle Streams monitors for database changes. How columns are monitored depends on whether rules are defined in the Capture Rules block:

- If rules are defined in the Capture Rules block, then the columns in the Capture Columns block are monitored only if the corresponding rules are evaluated to be *true*.
- If no rules are defined in the Capture Rules block, then the columns in the Capture Columns block are always monitored for changes.

Field	Description
Column Name	Column that Oracle Streams monitors for changes. You can monitor any column for the table that is selected in the Capture Tables block. List — Table Columns (all columns for the table selected in the Capture Tables block)
User ID	User ID that created the rule. Display only.
Activity Date	Date when the rule was created. Display only.

6 Identity Data Export Utilities

The Identity Data Export Utilities application is used to initially populate a central identity vault with person-related data from the Banner® database. Components of the application populate a central identify vault as follows:

- The UDCIdentifier Assigner assigns a UDCIdentifier to all persons in Banner who do not have a UDCIdentifier.
- The UDCIdentity Extractor extracts UDCIdentity information from the database and creates a single UDCIdentityList.xml file.
- The LDIF Generator parses the UDCIdentityList.xml file to create an LDIF (LDAP Data Interchange Format) file for import to LDAP (Lightweight Directory Access Protocol).
- The SPML Publisher uses information in the UDCIdentityList.xml file to publish UDCIdentity information to SPML compliant endpoints.

Note

The SPML LDAP Adapter can be used with the SPML Publisher as another way to create user accounts in an LDAP V3 compliant directory server. Refer to Chapter 10, "SPML LDAP Adapter".

The application is packaged in two zip files:

- `Deployables\OC4J\Identity_Data_Export_Utilities_full_release.zip` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\Identity_Data_Export_Utilities_full_release.zip` is used for installation on Oracle WebLogic Server 11g.

This chapter gives instructions for installing the application on both servers, configuring the application, and using the application to perform administrative tasks.

Installation on Oracle Application Server

`Deployables\OC4J\Identity_Data_Export_Utilities_full_release.zip` is used for installation on Oracle Application Server 10.1.3.4/5.This zip file contains an archive file named `IdentityDataExportUtilities.ear`.

The Identity Data Export Utilities application can be installed on an existing Oracle Application Server. The application can be deployed into a new or existing OC4J instance. The application can be deployed in the same OC4J instance as other Banner Enterprise

Identity Services (BEIS) applications or in its own OC4J instance. In either case, BEIS applications should be deployed separately from other applications so they can be managed independently.

Although the application runs on an application server, it requires a database for several of its features. Use the following steps to install the Identity Data Export Utilities application on OAS 10.1.3.4/5:

- Step 1, “Configure the database objects”
- Step 2, “Specify the working directory”
- Step 3, “Install the Identity Data Export Utilities”
- Step 4, “Configure the security role and user”
- Step 5, “Configure logging”

Step 1 Configure the database objects

Use the following steps to create the packages and synonyms that are required by the Identity Data Export Utilities.

1. Extract the `Identity_Data_Export_Utilities` archive file, which is in a tar or zip format.
2. Go to the directory where the file was extracted.
3. Navigate to the sql folder and open a command prompt.
4. Run SQL*Plus and connect as `BANINST1`.
5. Execute the following command at the SQL prompt:

   ```
   @setup.sql
   ```

   Required database objects are created with the required privileges.

Step 2 Specify the working directory

The Identity Data Export Utilities application includes components that perform file operations. These components require a working directory for writing and reading XML and LDIF files. The components use the following default working directory:

```
<OAS_HOME>/j2ee/<INSTANCE_NAME>/applications/
IdentityDataExportUtilities/IdentityDataExportUtilities/
generatedFiles
```

The following steps are optional. They are required only if the working directory needs to be changed. If the working directory is not changed, all files (XML and LDIF) that are

generated by the Identity Data Export Utilities will be deleted during redeployment of the application.

Use the following steps to customize the working directory.

1. Copy `IdentityDataExportUtilities.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 IdentityDataExportUtilities.ear
   ```

   The extract contains a Web archive named `IdentityDataExportUtilities.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>\IdentityDataExportUtilities.war
   ```

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

6. Change the `udcBatchApp.working.dir` property to the location to be used as the directory for file operations.

7. Save the changes.

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

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

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

   ```
   jar cvf IdentityDataExportUtilities.ear *.war META-INF/*
   ```

   The rebuilt `IdentityDataExportUtilities.ear` is used for installation.

Step 3 Install the Identity Data Export Utilities

Use the following steps to install the Identity Data Export Utilities to the Oracle Application Server.

1. Connect to the Oracle Enterprise Manager:

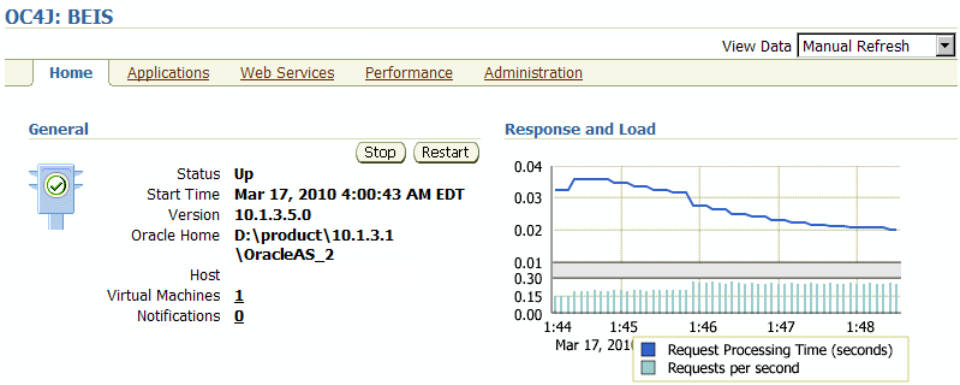
   `http://<host>:<port>/em`

   The console is displayed.

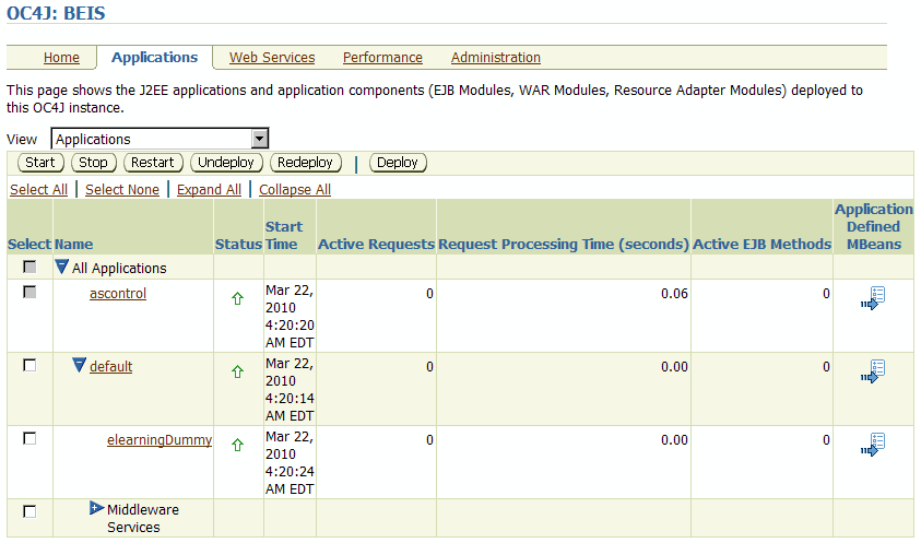
2. Click the name of the OC4J instance that will host the Identity Data Export Utilities application.

   The application can be installed in the same instance with other BEIS applications or in its own instance. In either case, BEIS applications must be installed separately from other applications so they can be independently managed.

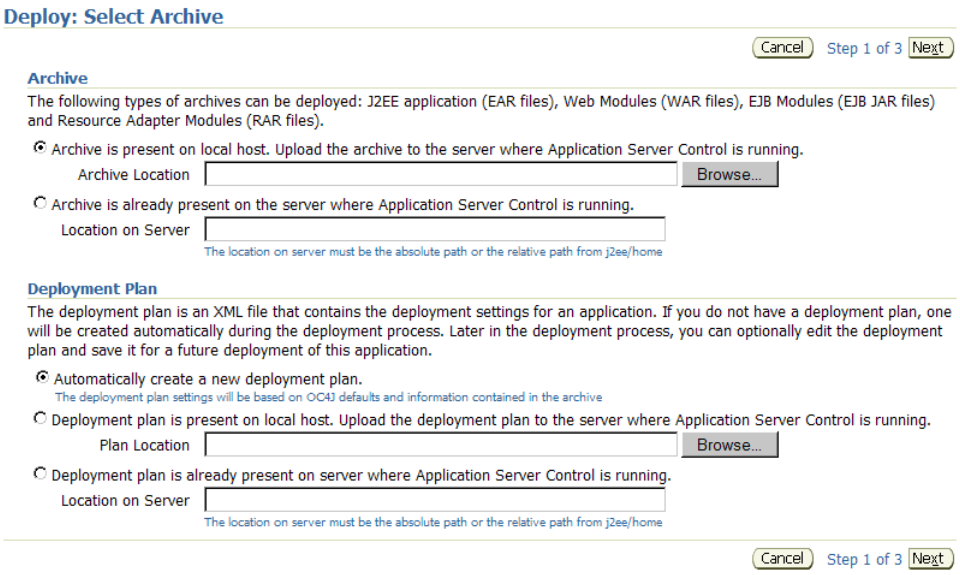
   The Home page for the selected instance is displayed.



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



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



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 navigate to the `IdentityDataExportUtilities.ear` file.

   5.3. Select the file and click **Open**.

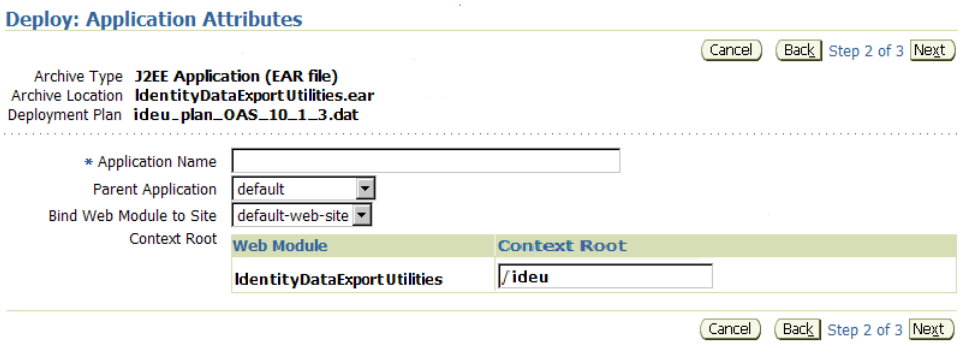
6. Select the deployment plan for the application:

   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 `ideu_plan_OAS_10_1_3.dat` file.

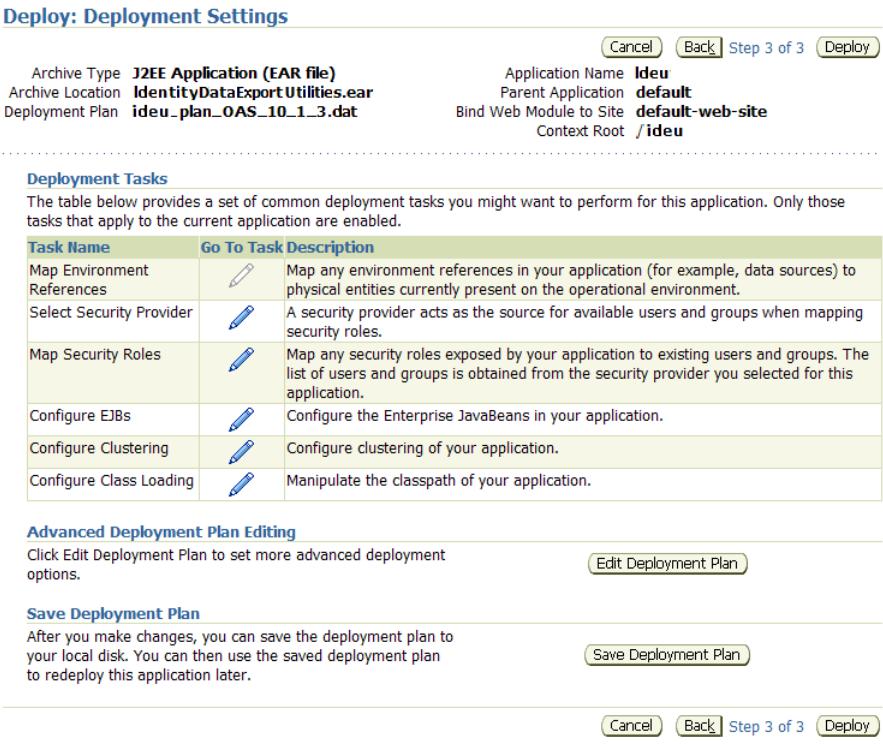
   6.3. Select the file and click **Open**.

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



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

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



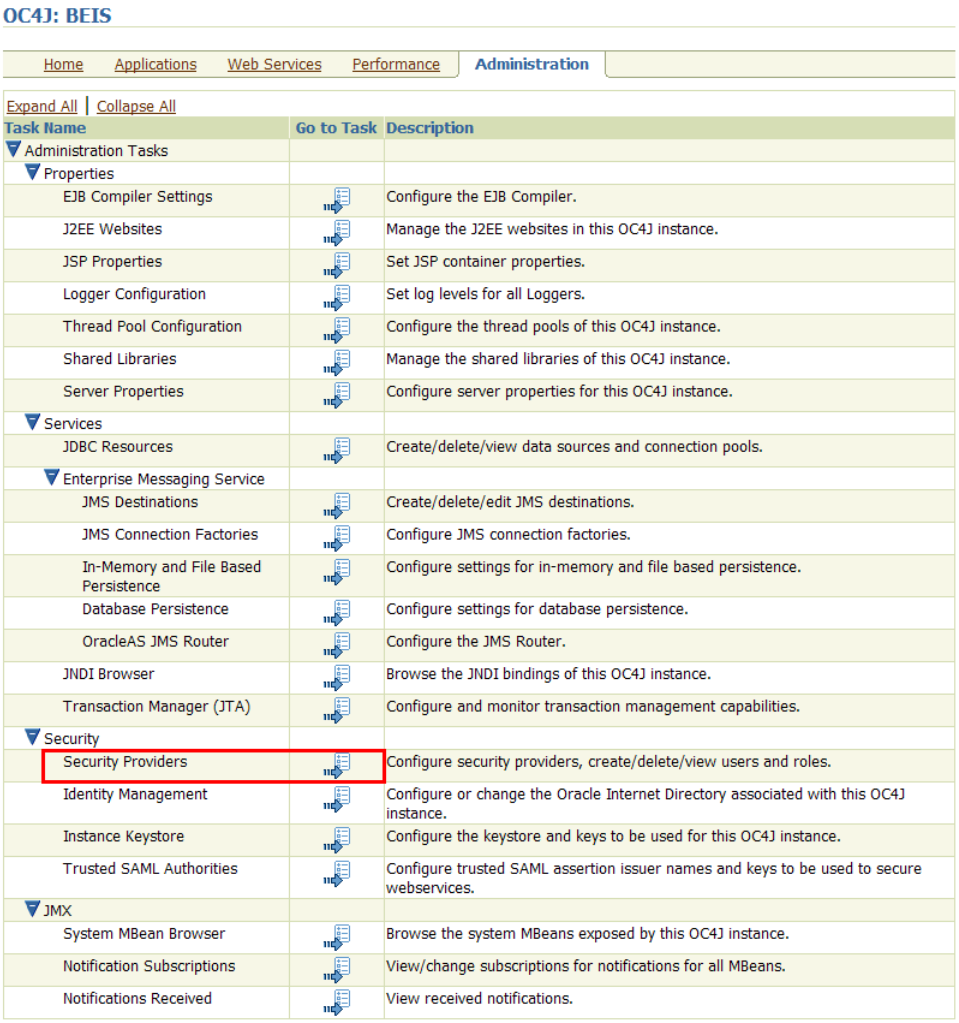
10. Click **Deploy** to accept the values and install the Identity Data Export Utilities. A deployment confirmation page is displayed.

11. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

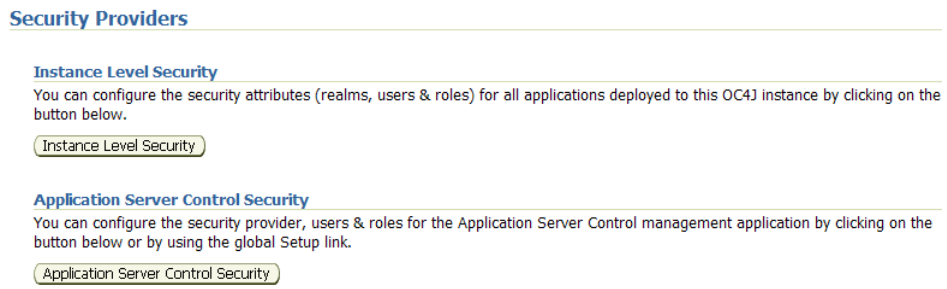
Step 4 Configure the security role and user

Use the following steps to add the `udcIdentityAdmin` role and an administrative user to the Identity Data Export Utilities application. This role and user are required for accessing the Identity Data Export Utilities administrative interface.

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

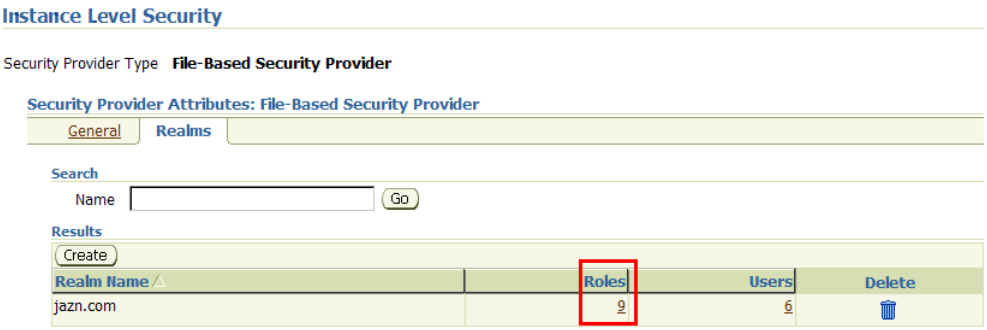


2. Select **Security Providers** from the Security section. The Security Providers page is displayed.

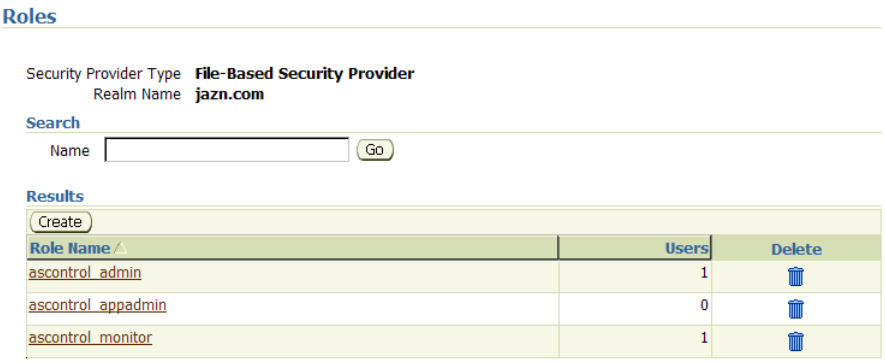


3. Click **Instance Level Security**. The Instance Level Security page is displayed.

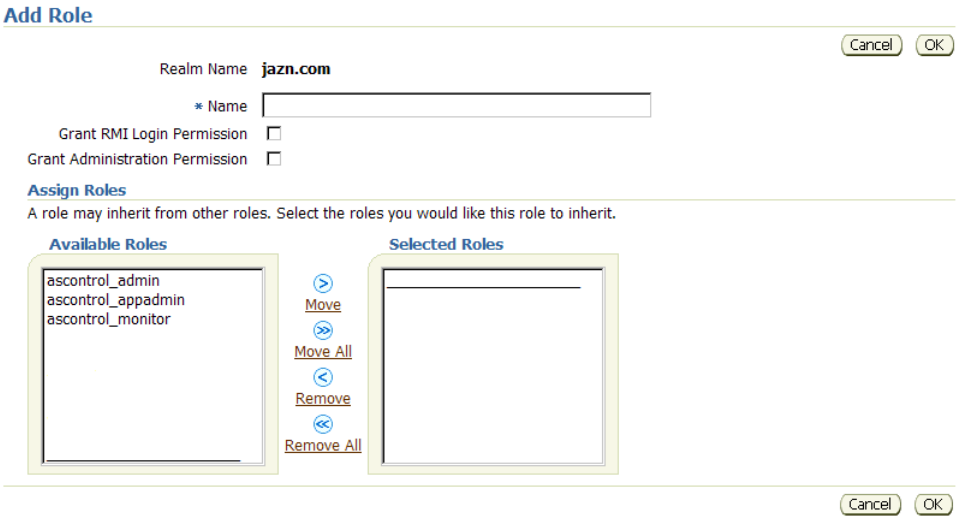
4. Select the **Realms** tab.



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



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



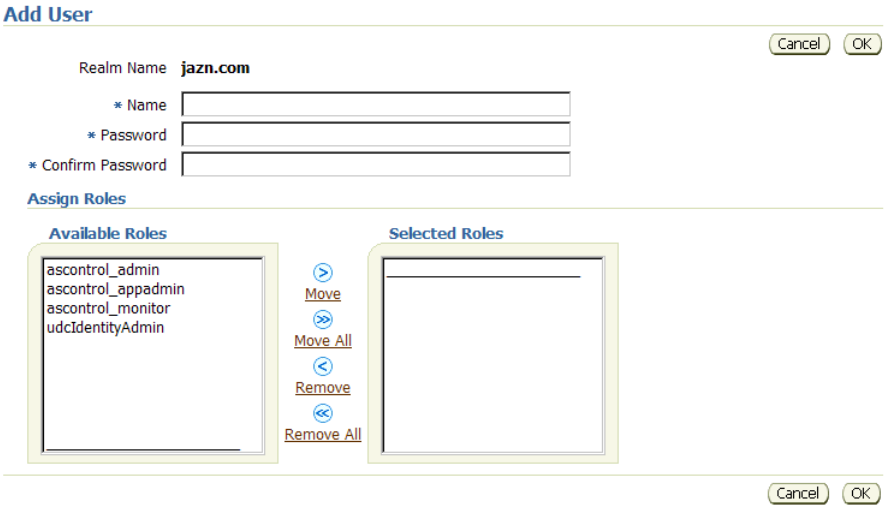
7. Enter *udcIdentityAdmin* in the **Name** field.

8. Click **OK**. The Roles page is redisplayed with the new role.

9. Return to the Instance Level Security page.

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

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



12. Enter the following information to create a user:

Name	*Admin* (This is an example. Enter the name of your choice.)
Password	Password used to log in to the Identity Data Export Utilities administrative interface
Confirm Password	Confirmation of the password

13. In the Assign Roles section, select the *udcIdentityAdmin* 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 5 Configure logging

The Identity Data Export Utilities application uses 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 file**. The default location is `<OAS_HOME>/j2ee/home/identity_data_export_utilitites.log`. This location should be changed to the OC4J instance where the Identity Data Export Utilities application is installed.
- **Location of the SPML Publisher failure log**. The default location is `<OAS_HOME>/j2ee/home/spml_publisher_failure.log`. This location should be changed to the OC4J instance where the Identity Data Export Utilities application is installed.
- **Logging level**. The default level is *INFO*, resulting in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide detailed logging for initial operations, you should change the logging level to *DEBUG*.

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

1. Navigate to `<OAS_HOME>/j2ee/<OC4J instance>/applications/Identity_Data_Export_Utilities/IdentityDataExportUtilities/WEB-INF/classes`.
2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
`log4j.appender.out.File`	`identity_data_export_utilities.log`	`../<OC4J instance>/log/ identity_data_export_utilities.log`
`log4j.appender.failure.File`	`spml_publisher_failure.log`	`../<OC4J instance>/log/ spml_publisher_failure.log`
`log4j.root Category`	`INFO,out`	`DEBUG,out`

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

Installation on Oracle WebLogic Server 11g

`Deployables\Weblogic\Identity_Data_Export_Utilities_full_release.zip` is used for installation on Oracle WebLogic Server 11g. This zip file contains an archive file named `IdentityDataExportUtilities.ear`.

Recommended configuration

BEIS components must be installed in an Oracle WebLogic Basic Domain. They must not be installed in an Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual server for BEIS and other middle-tier components. This server would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server instance should consist of the default Admin Server and at least two Managed Servers:

- One Managed Server for the Banner Identity Gateway and the Enterprise Identity Proxy Services, which must be installed together
- One Managed Server for the Identity Data Export Utilities and the SSO Manager

If a domain based on the Basic Domain template already exists for middle-tier applications, the BEIS components can be installed in two separate Managed Servers in that domain, based on the preceding recommendation.

Refer to the Oracle WebLogic Server Documentation Library for details on creating a new domain and a new Managed Server.

Installation steps

Although the Identity Data Export Utilities application runs on an application server, it requires a database for several of its features. Use the following steps to install the application on Oracle WebLogic Server 11g (version 10.3.2):

- Step 1, “Configure the database objects”
- Step 2, “Customize properties”
- Step 3, “Install the Identity Data Export Utilities”
- Step 4, “Configure the security group and user”

Step 1 Configure the database objects

Use the following steps to create the packages and synonyms required by the Identity Data Export Utilities.

1. Extract the `Identity_Data_Export_Utilities` archive file, which is in a tar or zip format.
2. Go to the directory where the file was extracted.
3. Navigate to the `db-scripts` directory and open a command prompt.
4. Run SQL*Plus and connect as `BANINST1`.
5. Execute the following command at the SQL prompt:

   ```
   @setup.sql
   ```

   Required database objects are created with the required privileges.

Step 2 Customize properties

The following properties can be customized:

- **Working directory** - The Identity Data Export Utilities application includes components that perform file operations. These components require a working directory for writing and reading XML and LDIF files. The components use the following default working directory:

  ```
  <Oracle_Middleware_HOME>/user_projects/domains/
  <domain_name>/servers/AdminServer/tmp/_WL_user/
  IdentityDataExportUtilities/b6wyjh/war/generatedFiles
  ```

  The working directory can optionally be changed. If the working directory is not changed, all files (XML and LDIF) generated by the Identity Data Export Utilities will be deleted if the application is redeployed.

- **Logging** - The Identity Data Export Utilities application uses Apache's log4j to log the activities performed by the application at runtime. The `identity_data_export_utilities.log` file is located at the following location:

  ```
  Oracle\Middleware\user_projects\domains\<domain_name>
  ```

  where `<domain_name>` is the name of the domain where the application is installed. This location cannot be changed.

  A property in the `log4j.properties` file determines the logging level. The default logging level is *INFO*, which results in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. You can modify the logging level if you want more detailed logging.

Use the following steps to customize the working directory and logging.

1. Copy `IdentityDataExportUtilities.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 IdentityDataExportUtilities.ear
   ```

   The extract contains a Web archive named `IdentityDataExportUtilities.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>/IdentityDataExportUtilities.war
   ```

5. Use the following steps to customize the working directory:

   5.1. Open `war_home\WEB-INF\classes\udcBatchApp.properties`.

   5.2. Change the `udcBatchApp.working.dir` property to the location to be used as the directory for file operations.

   5.3. Save the change.

6. Use the following steps to customize the logging level:

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

   6.2. Change the `log4j.rootCategory` property as follows:

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

   6.3. Save the change.

7. From `war_home` execute the following command to rebuild the Web archive file:

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

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

   ```
   jar cvf IdentityDataExportUtilities.ear *.war META-INF/*
   ```

   The rebuilt `IdentityDataExportUtilities.ear` is used for installation.

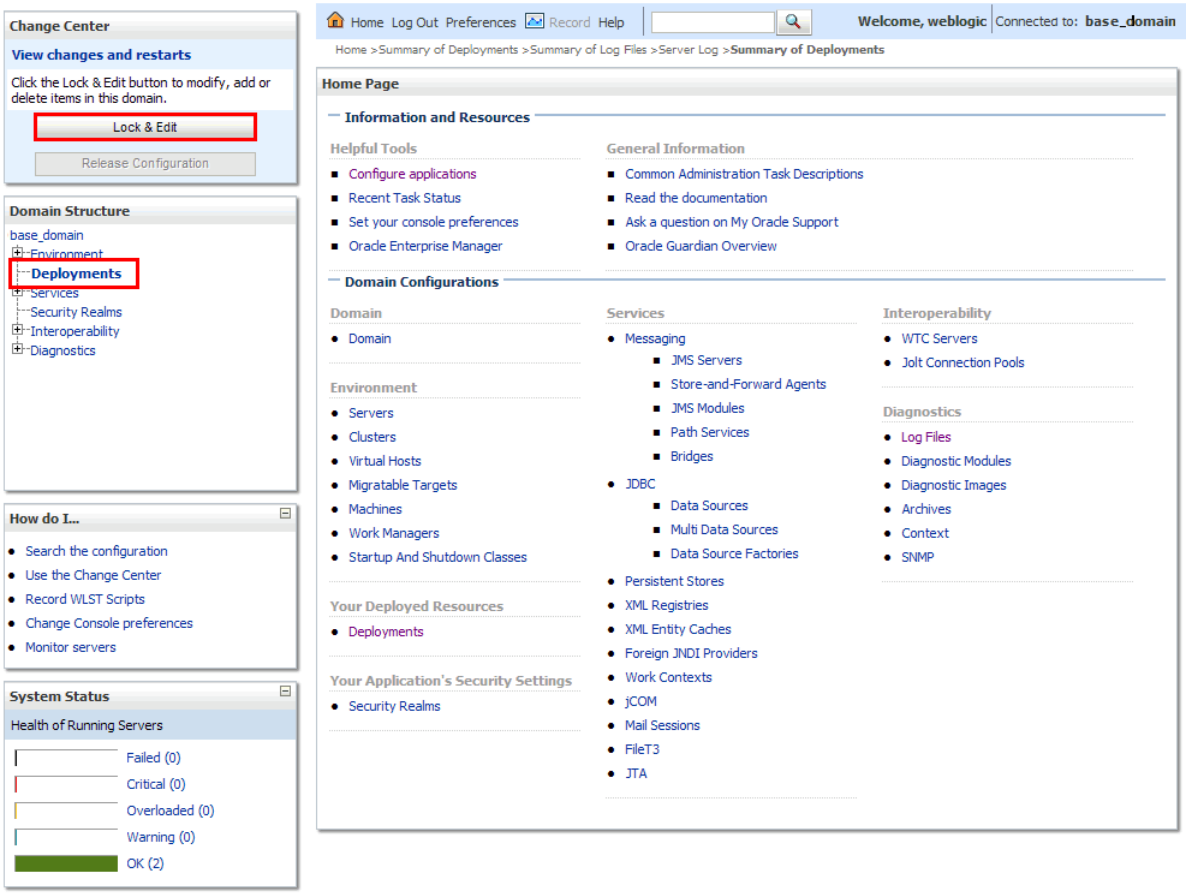
Step 3 Install the Identity Data Export Utilities

Use the following steps to install the Identity Data Export Utilities to the Oracle WebLogic Server.

1. Connect to the Oracle WebLogic Server Administration Console:

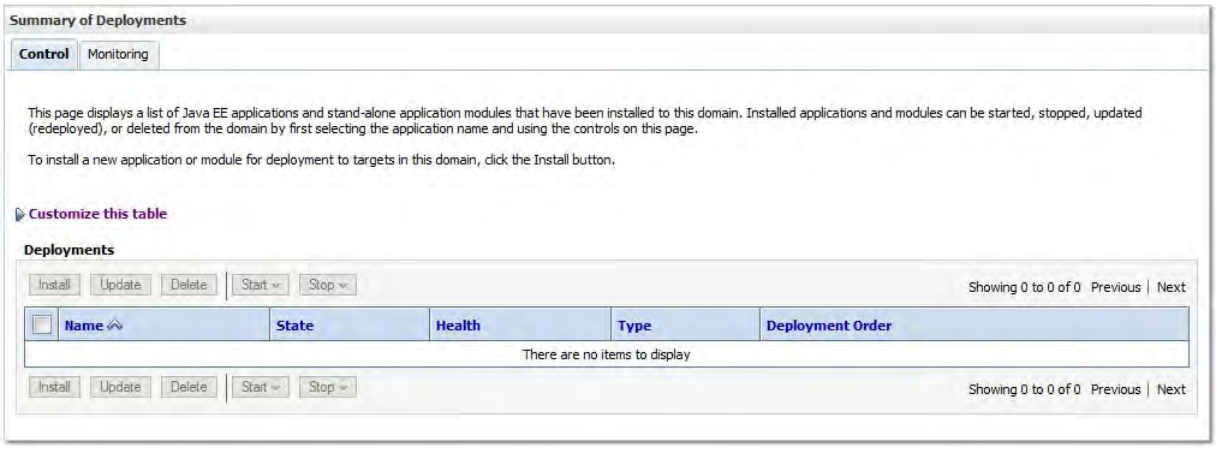
   ```
   http://<host>:<port>/console
   ```

   The Home Page is displayed.

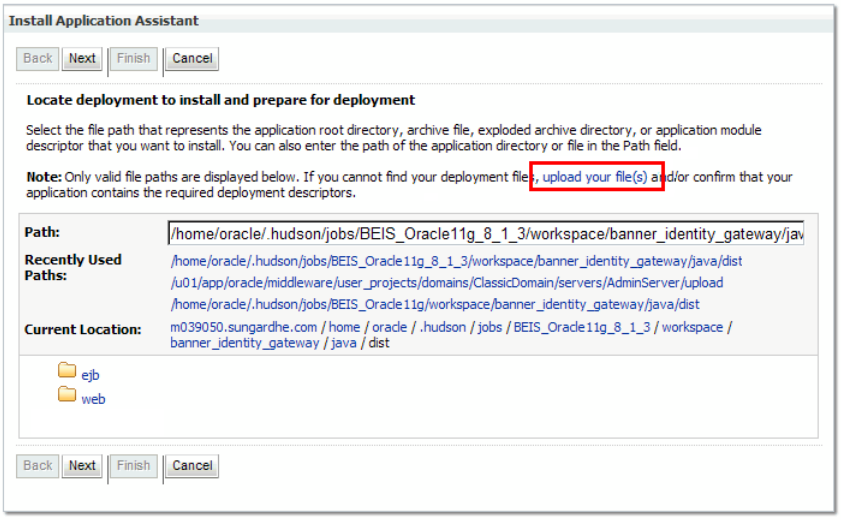


2. In the Change Center pane, click **Lock & Edit**.

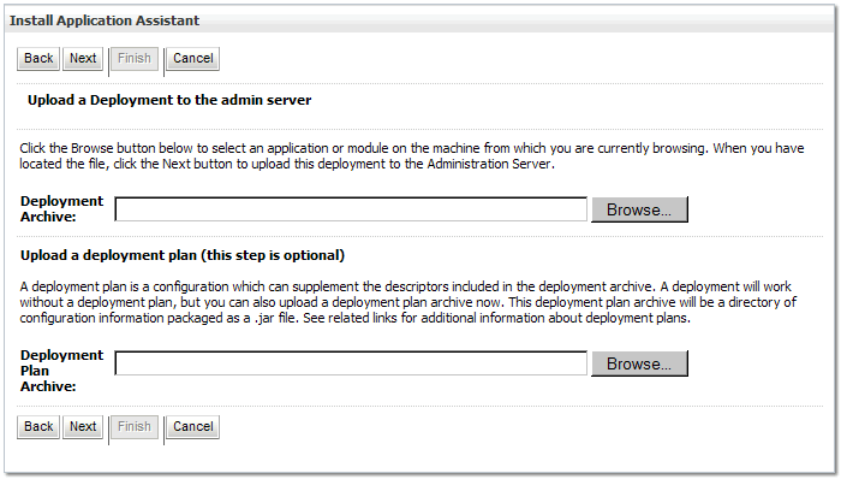
3. In the Domain Structure pane, click **Deployments**. The Summary of Deployments page is displayed.



4. Click **Install**. The Install Application Assistant page is displayed.



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

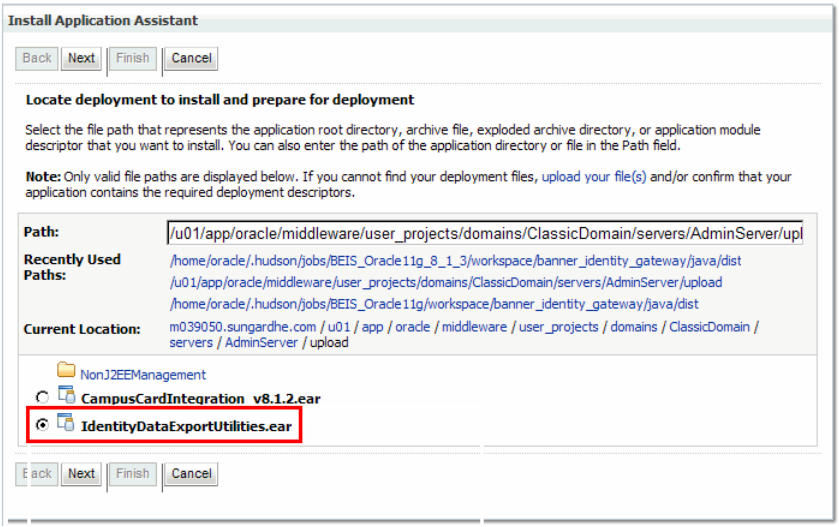


6. Select the file to be uploaded:

   6.1. In the **Deployment Archive** field, click **Browse** and navigate to the `IdentityDataExportUtilities.ear` file.

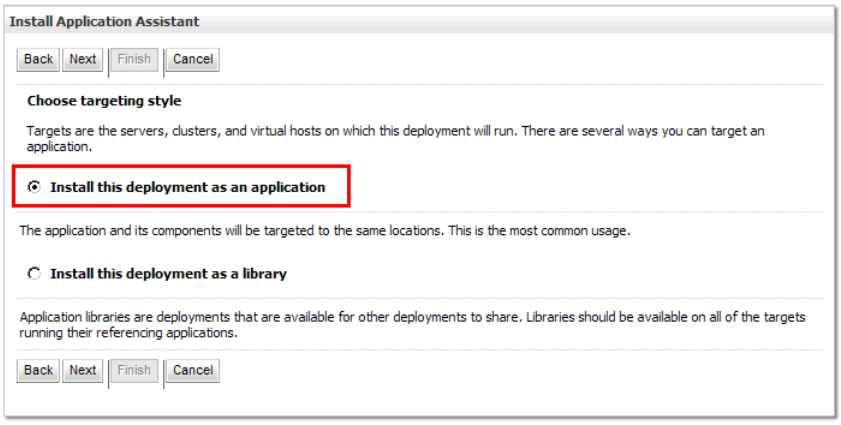
   6.2. Select the file and click **Open**.

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



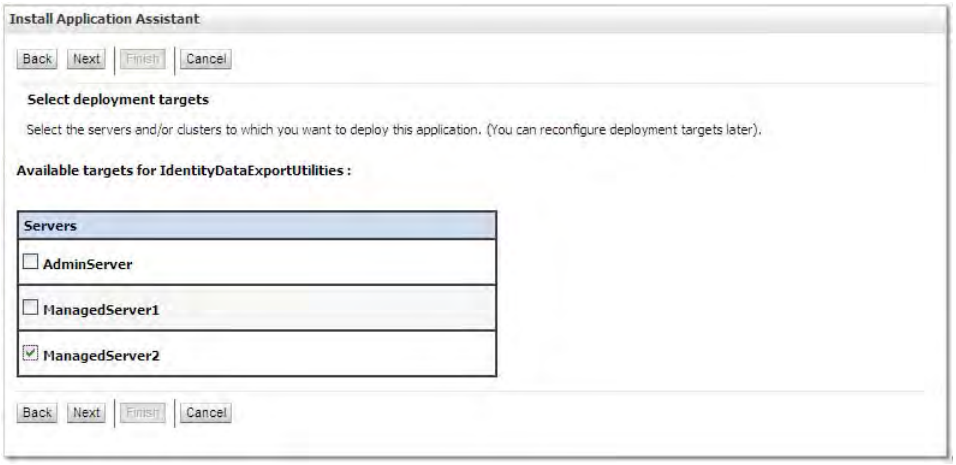
8. Select the `IdentityDataExportUtilities.ear` file from the list.

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.

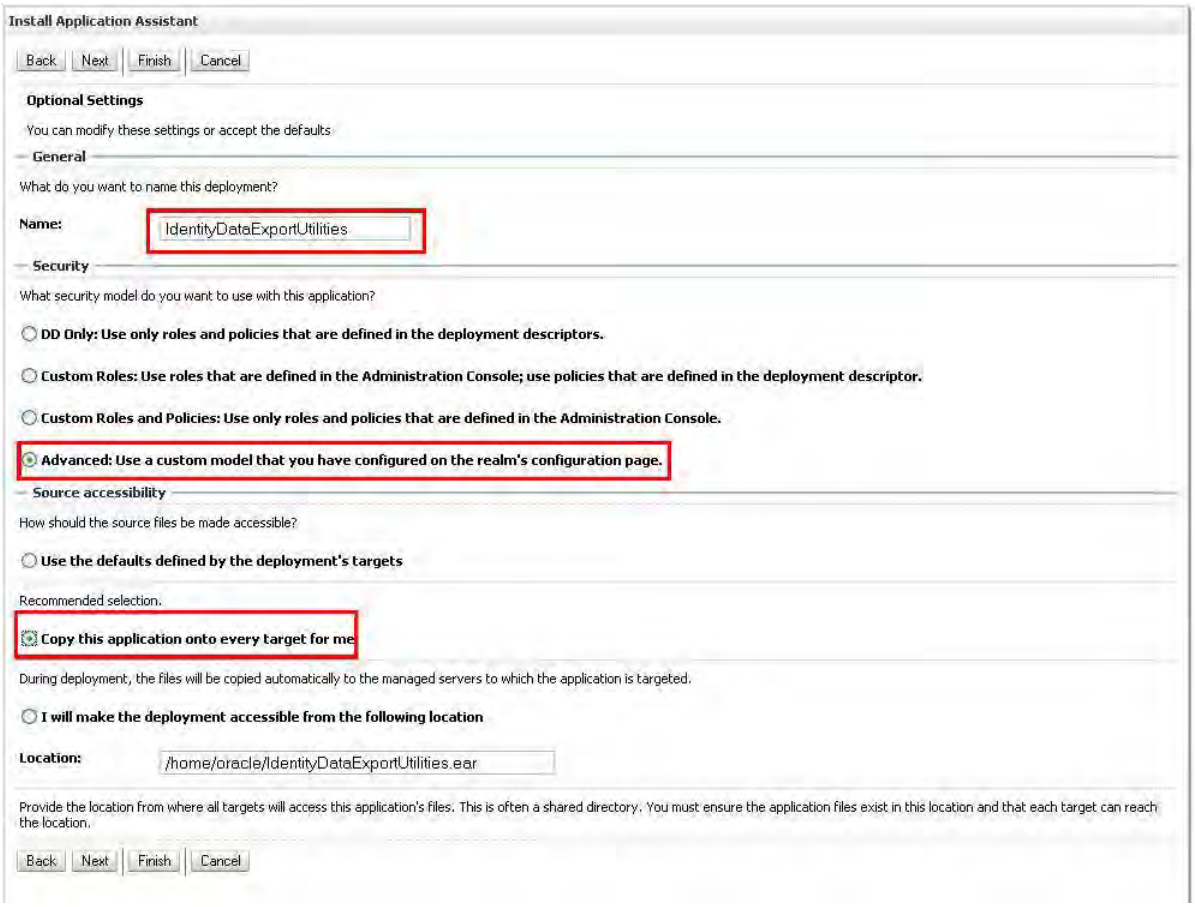


12. Select the server where the application should be deployed. (The application can be installed on an existing server.)

Note

SunGard Higher Education recommends deploying applications to a WebLogic Managed Server and not to the Administration Server. If you do not see the preceding page, you should check your WebLogic Server configuration to ensure that a Managed Server is available for deployment of applications. If a Managed Server is not available, the application will be deployed to the Administration Server, which is not a recommended configuration. For more information, consult the Oracle WebLogic Server Documentation Library. ■

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

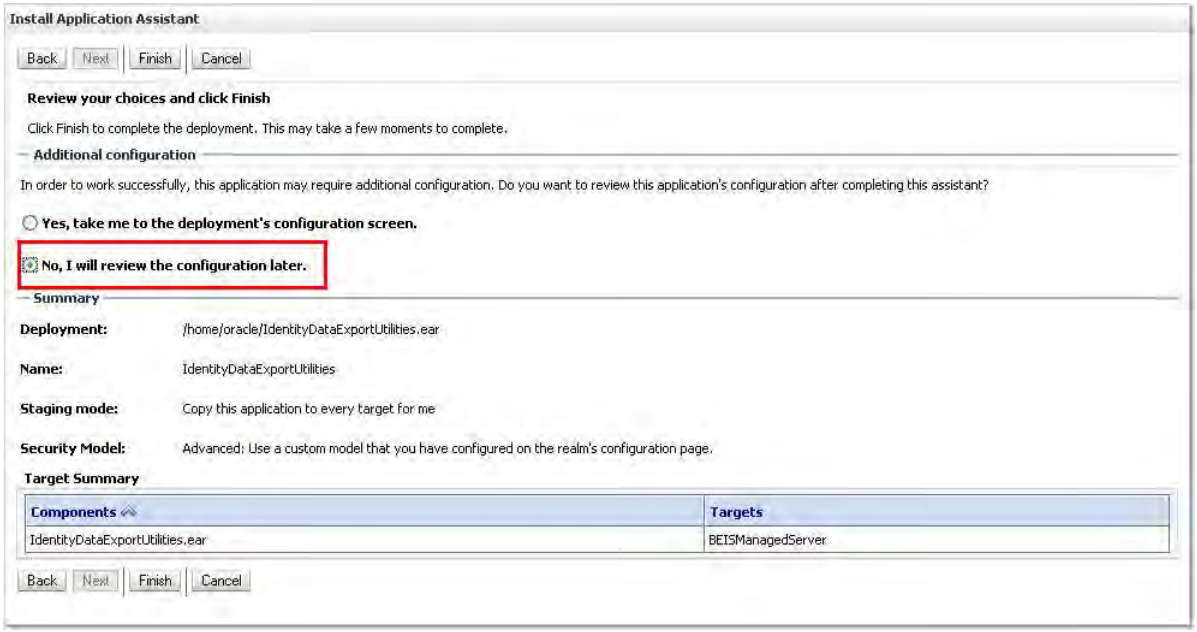


14. Enter a name for the application (for example, *IdentityDataExportUtilities*) in the **Name** field.

15. Select **Advanced: Use a custom model that you have configured on the realms configuration page.**

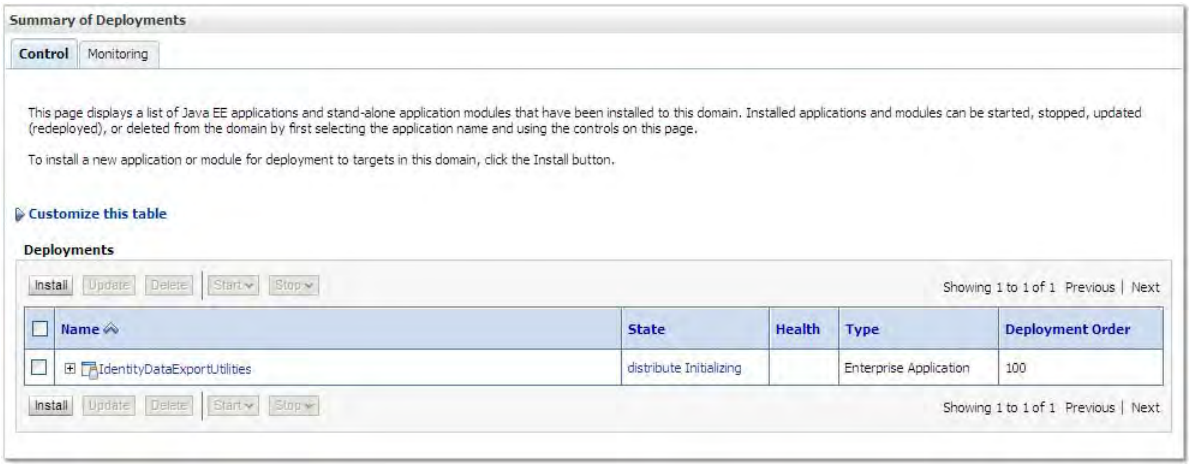
16. Select **Copy this application onto every target for me**.

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



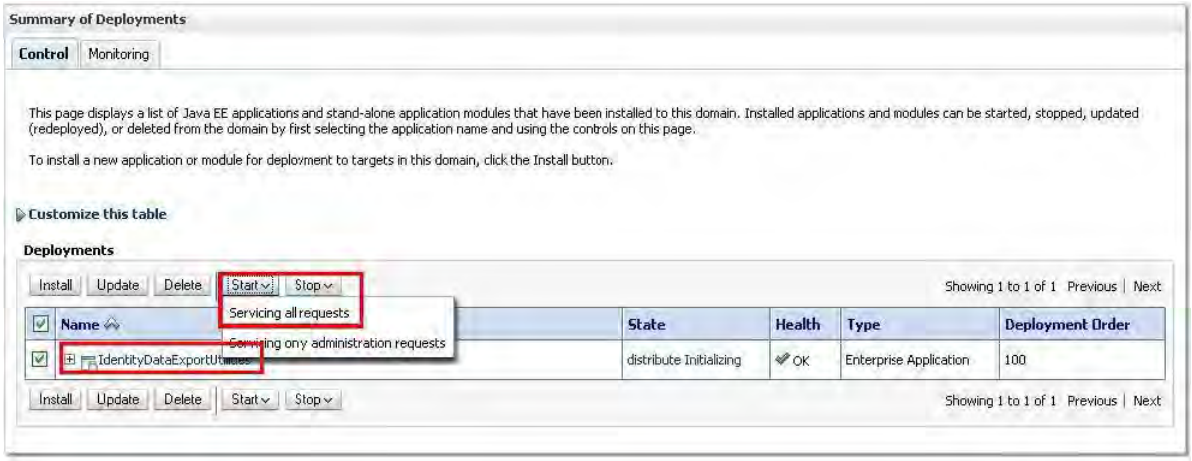
18. Select **No, I will review the configuration later**.

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



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

21. Start the newly deployed application as follows:



21.1. Select the newly deployed application.

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

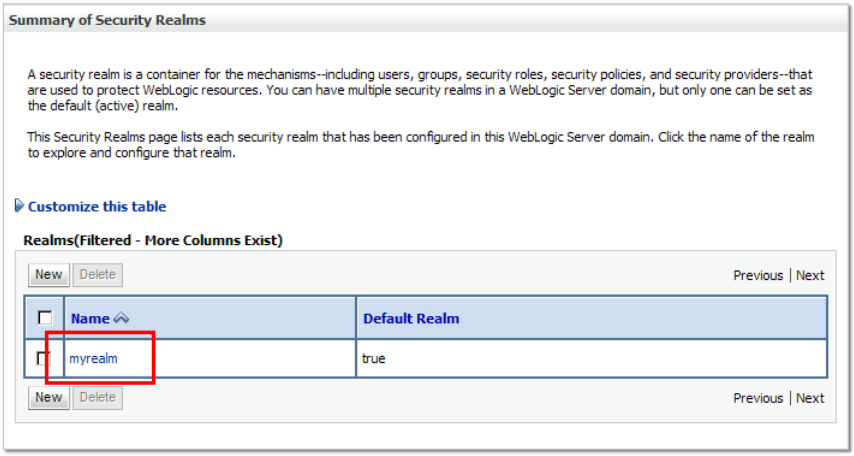
21.3. Click **Yes**.

Step 4 Configure the security group and user

Use the following steps to add the `udcAdminGroup` group and an administrative user to the Identity Data Export Utilities application. This group and user are required for accessing the Identity Data Export Utilities administrative interface.

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

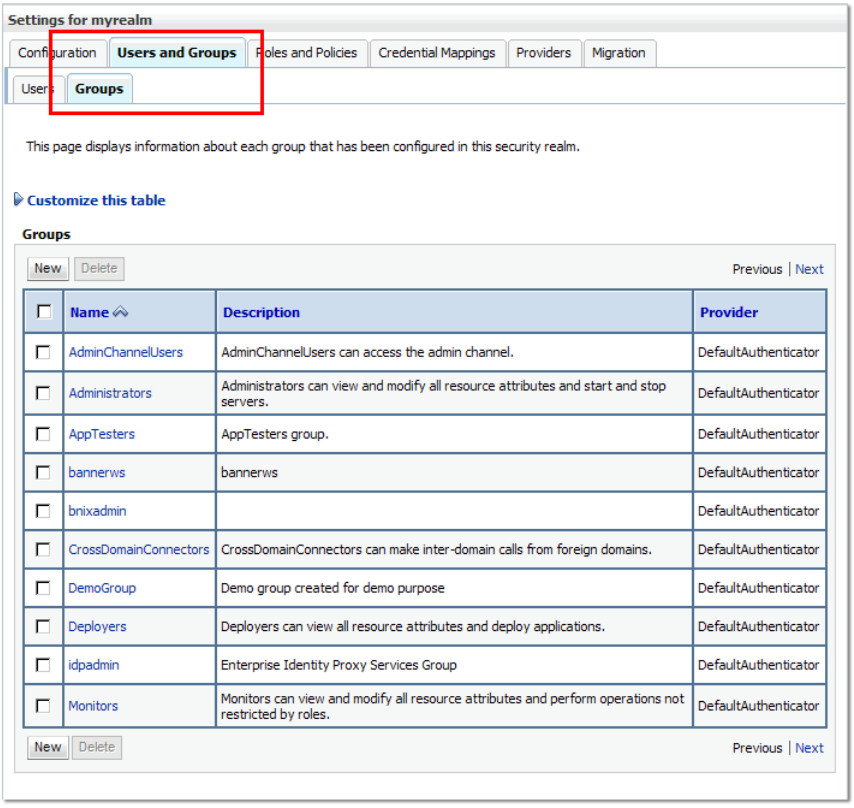
The Summary of Security Realms page is displayed.



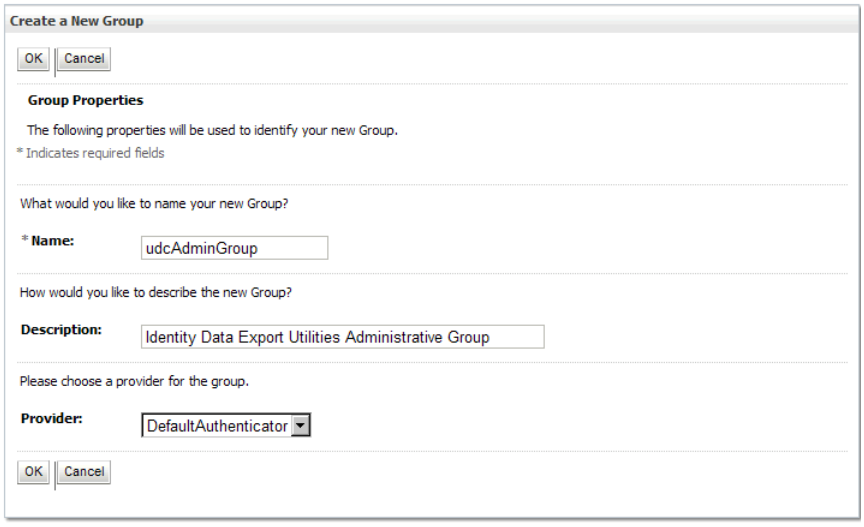
2. Click **myrealm**. The Settings page is displayed.

3. Select the **Users and Groups** tab.

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



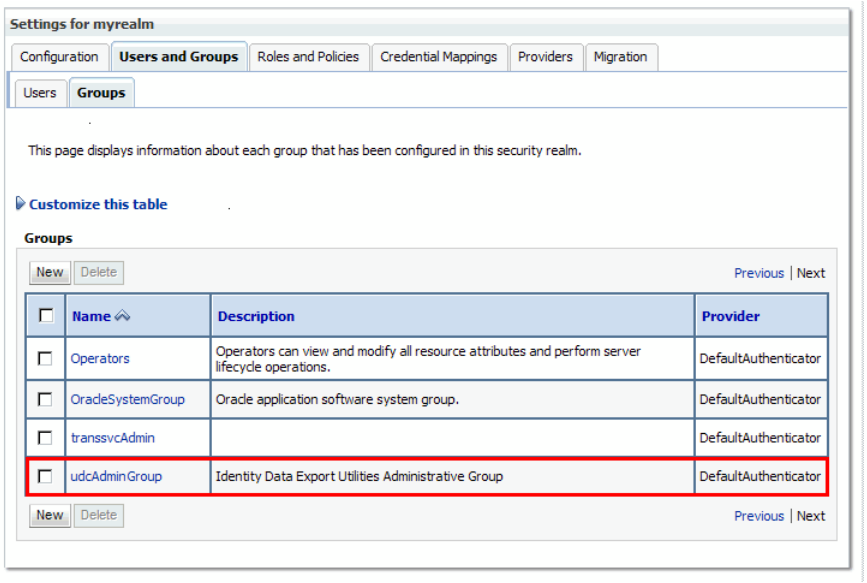
5. Click **New**. The Create a New Group page is displayed.



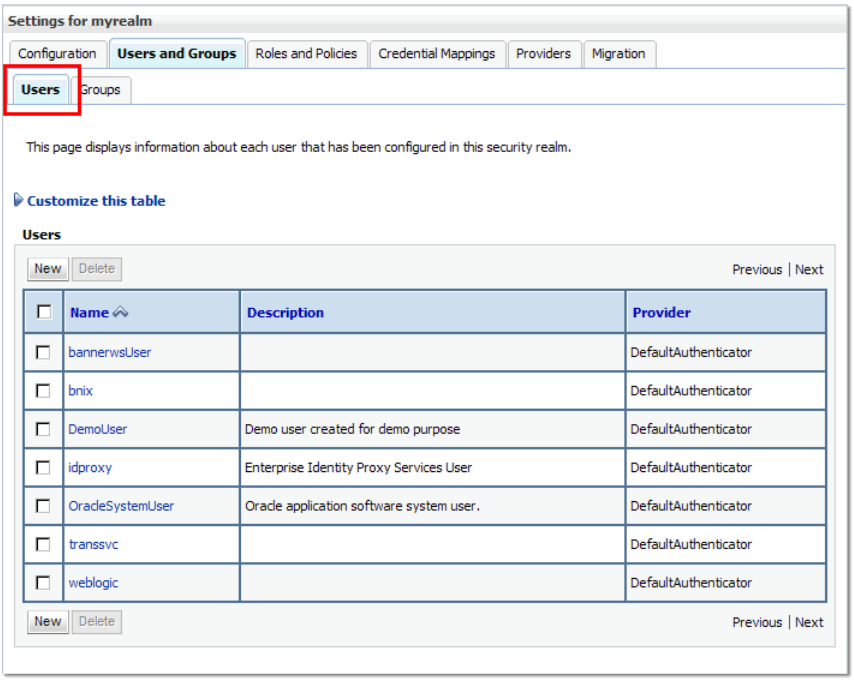
6. Enter the following information to create a group:

Name	*udcAdminGroup*
Description	*Identity Data Export Utilities Administrative Group*
Provider	*DefaultAuthenticator*

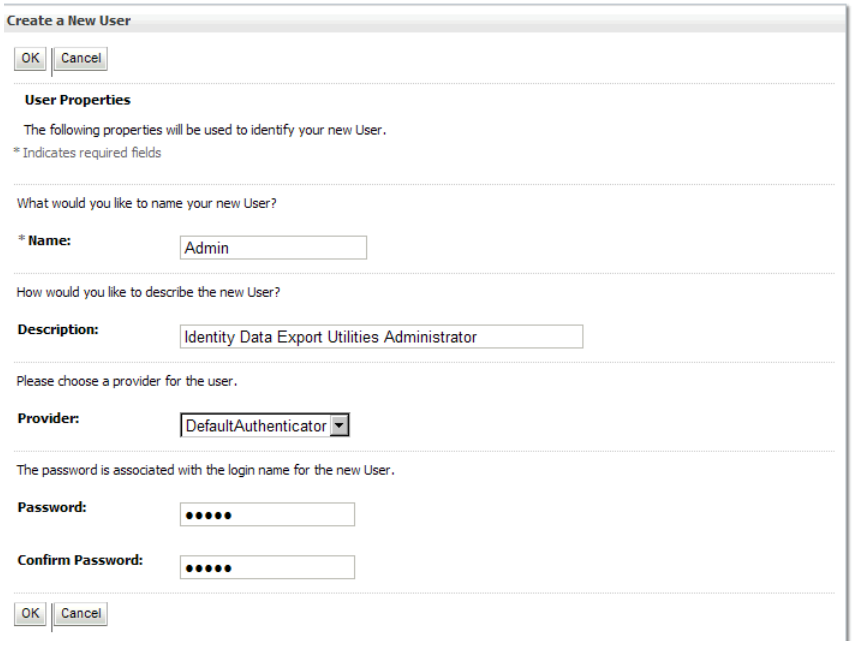
7. Click **OK**. The table of groups is redisplayed with the new group.



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



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

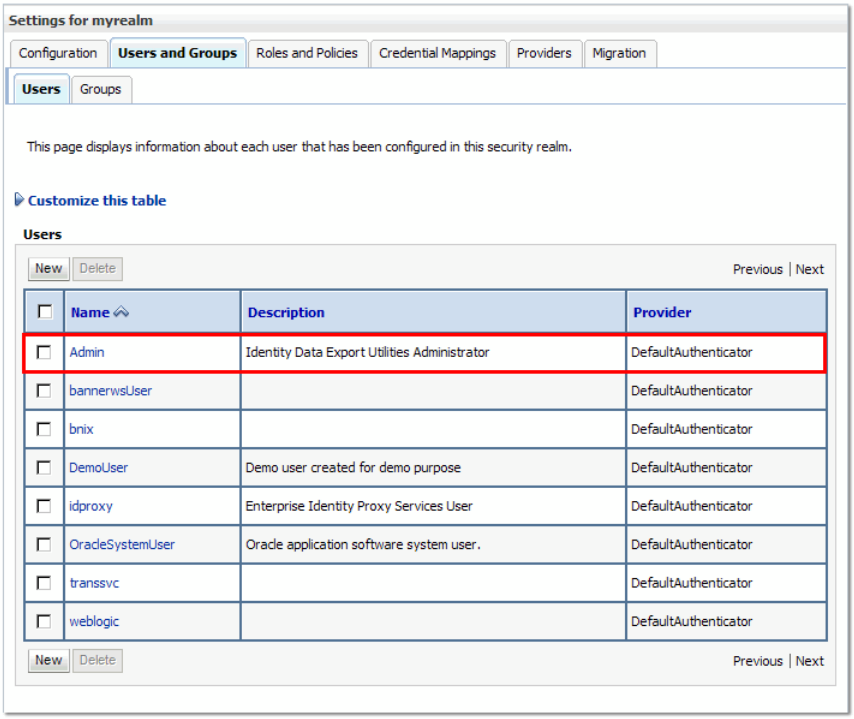


C

10. Enter the following information to create a user:

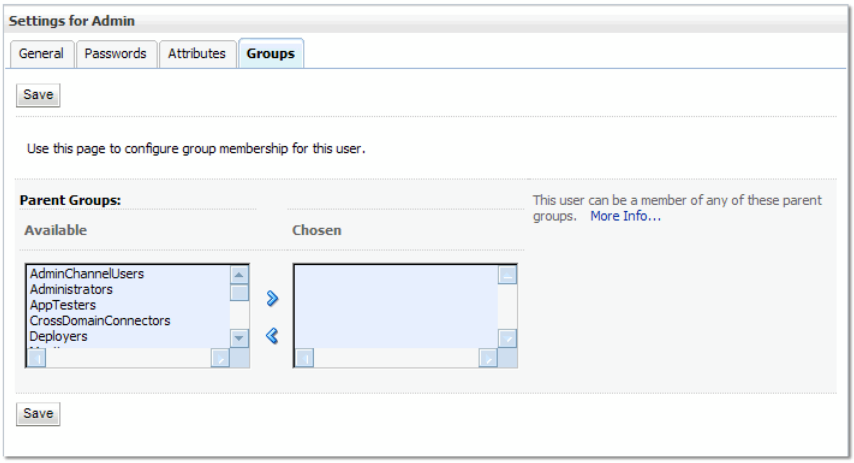
Name	*Admin* (This is an example. Enter the name of your choice.)
Description	*Identity Data Export Utilities Administrator*
Provider	*DefaultAuthenticator*
Password	Password used to log in to the Identity Data Export Utilities administrative interface
Confirm Password	Confirmation of the password

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

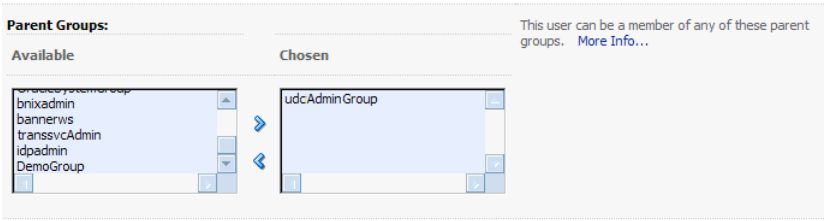


12. Click the name of the user you just created. The Settings page for the user is displayed.

13. Select the **Groups** tab.



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



15. Click **Save**.

Configuration

After the Identity Data Export Utilities application is installed, you must configure the following components:

- UDCIdentifier Assigner
- UDCIdentity Extractor
- LFIF Generator
- SPML Publisher

The following sections provide the steps for configuring each component.

Note

Do not change configurations when the components are running.

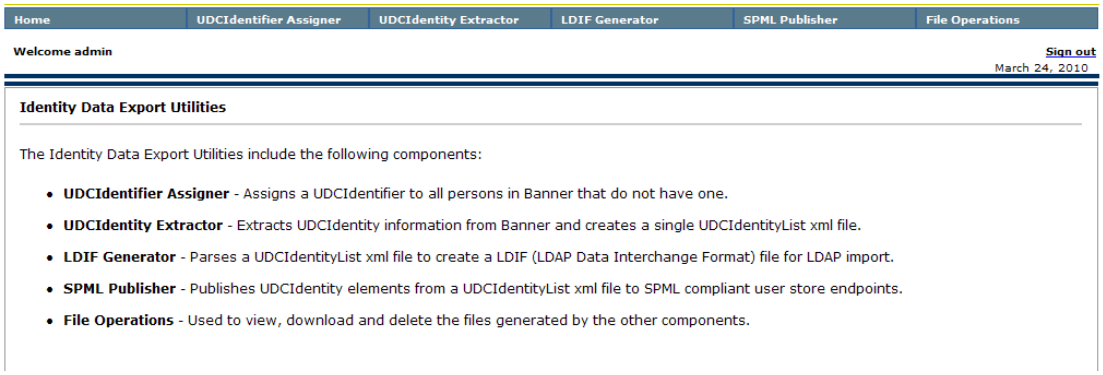
Configure the UDCIdentifier Assigner

Use the following steps to configure the UDCIdentifier Assigner.

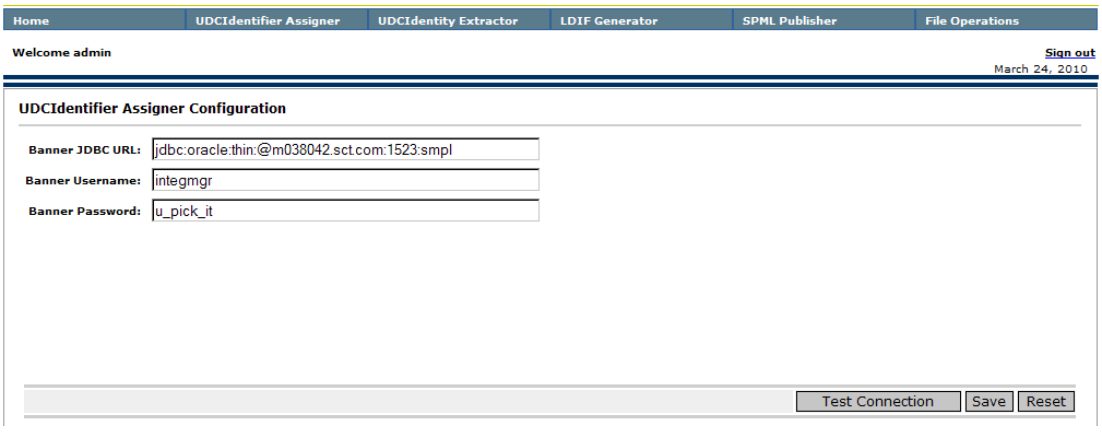
1. Connect to the Identity Data Export Utilities:

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

2. Log in with the user name and password that were mapped to the `udcIdentityAdmin` role (OAS) or the `udcAdminGroup` group (Oracle WebLogic). The following page is displayed.



3. Select UDCIdentifier Assigner > Configuration from the menu bar. The UDCIdentifier Assigner Configuration page is displayed.



4. Enter the following information to configure the UDCIdentifier Assigner:

Banner JDBC URL `jdbc:oracle:thin:@<host>:<port>:<service_name>`

This is the database URL. This format must be used.

Example

`jdbc:oracle:thin:@maldevs4:1521:s4ident`

where `maldevs4` is the `host`, `1521` is the `port`, and `s4ident` is the `service_name`. This example is for an Oracle database only.

Banner Username `integmgr`

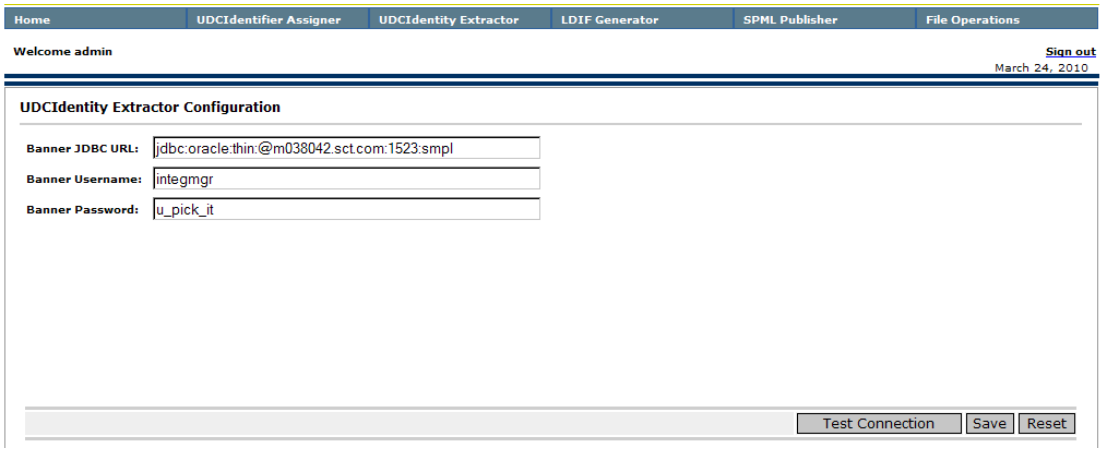
Banner Password Password for `integmgr`

5. Click **Save**.

Configure the UDCIdentity Extractor

Use the following steps to configure the UDCIdentity Extractor.

1. Select UDCIdentity Extractor > Configuration from the menu bar on the Identity Data Export Utilities administrative interface. The UDCIdentity Extractor Configuration page is displayed.



2. Enter the following information to configure the UDCIdentity Extractor:

Banner JDBC URL `jdbc:oracle:thin:@<host>:<port>:<service_name>`

This is the database URL. This format must be used.

Example

`jdbc:oracle:thin:@maldevs4:1521:s4ident`

where `maldevs4` is the `host`, `1521` is the `port`, and `s4ident` is the `service_name`. This example is for an Oracle database only.

Banner Username `integmgr`

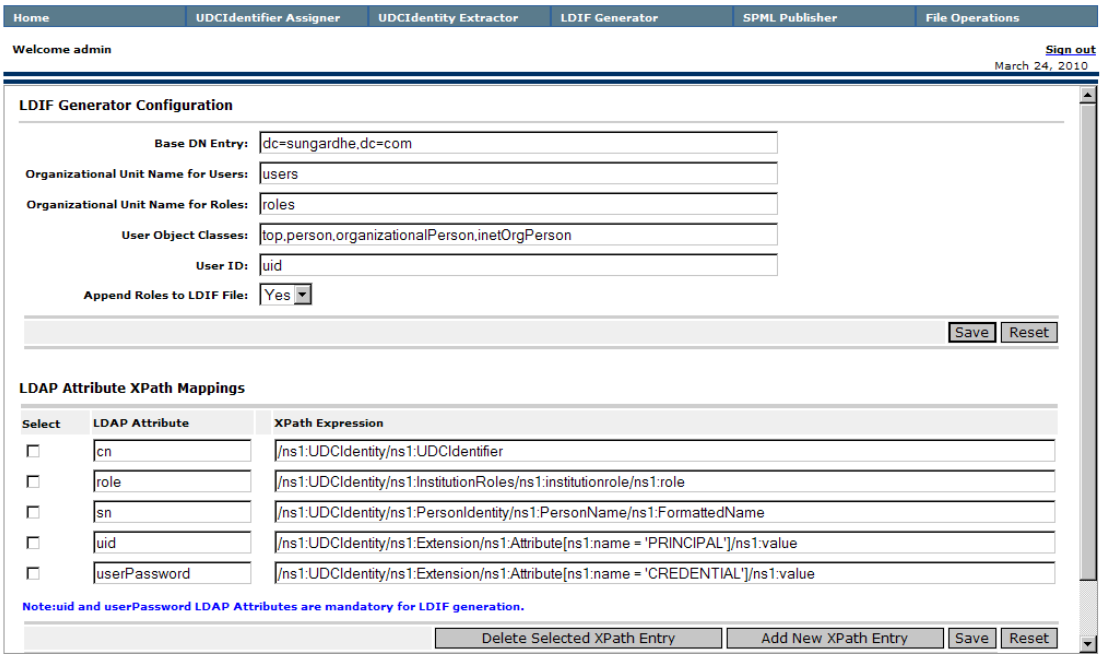
Banner Password Password for `integmgr`

3. Click **Save**.

Configure the LDIF Generator

Use the following steps to configure entries in a generated LDIF file. You must understand LDAP. Review the LDIF Generator configuration with your LDAP administrator.

1. Select LDIF Generator > Configuration from the menu bar on the Identity Data Export Utilities administrative interface. The LDIF Generator Configuration page is displayed.



2. Enter the following information to configure the entries that LDIF Generator writes in the LDIF file:

Field Name	Description	Sample Values
Base DN Entry	Base DN entries of the LDAP directory	`dc=sungardhe,dc=com`
Organizational Unit Name for Users	User's group name in the LDAP directory	`users`
Organizational Unit Name for Roles	Role's group name in the LDAP directory	`bannerRoles`

User Object Classes	Object classes added to the user group	`top`
		`person`
		`organizationalPerson`
		`inetOrgPerson`

User ID	uid attribute for user entry in the LDAP directory	`uid`
Append Roles to LDIF File	Option to append roles to main user LDIF file	`Yes` `No`

3. Click **Save**.

4. Enter the following information to map LDAP attributes to particular elements in `UDCIdentityList.xml` using XPath expressions.

Note

XPath (XML Path Language) is a language for selecting nodes from an XML document. This step captures the XPath entries to retrieve the list of required elements from the xml file generated by UDCIdentity Extractor and add those elements to the LDIF file.

LDAP Attribute	Description	XPath Expression
`cn`	Common name	`/ns1:UDCIdentity/` `ns1:UDCIdentifier`

`role`	Roles of the person	`/ns1:UDCIdentity/`
		`ns1:InstitutionRoles/`
		`ns1:institutionrole/ns1:role`

`sn`	Surname or last name of the person	`/ns1:UDCIdentity/`
		`ns1:PersonIdentity/`
		`ns1:PersonName/`
		`ns1:FormattedName`

LDAP Attribute	Description	XPath Expression
`uid`	User ID to uniquely identify the person	`/ns1:UDCIdentity/ ns1:Extension/ ns1:Attribute[ns1:name = 'PRINCIPAL']/ns1:value` (The PRINCIPAL extension attribute is configured on GORRSQL.)
`userPassword`	Password of the user	`/ns1:UDCIdentity/ ns1:Extension/ ns1:Attribute[ns1:name = 'CREDENTIAL']/ns1:value` (The CREDENTIAL extension attribute is configured on GORRSQL.)

5. Click **Save**.

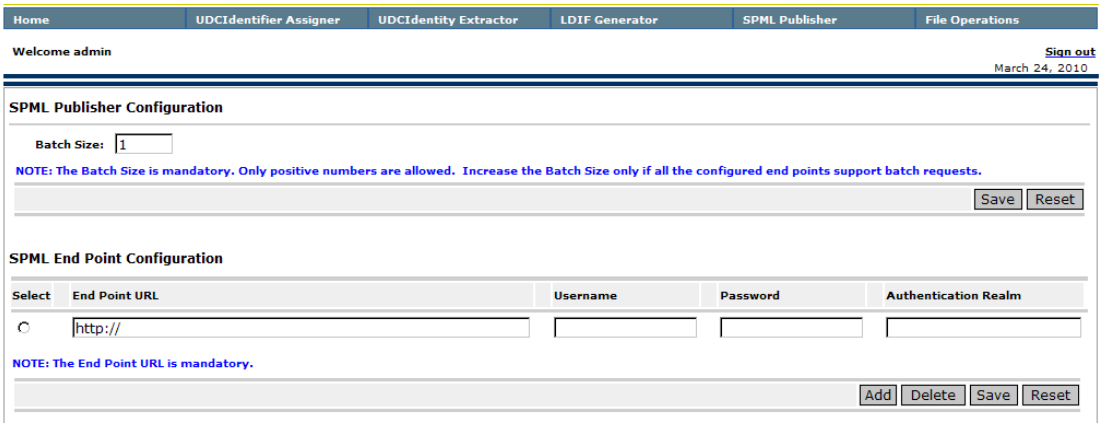
Note

To add a new XPath mapping, click **Add New XPath Entry**. To delete an XPath mapping, select the associated check box and click **Delete Selected XPath Entry**.

Configure the SPML Publisher

Use the following steps to configure the SPML Publisher.

1. Select SPML Publisher > Configuration from the menu bar on the Identity Data Export Utilities administrative interface. The SPML Publisher Configuration page is displayed.



2. Enter **Batch Size**.

   The SPML Publisher sends the UDCIdentity data to the configured SPML Web service endpoints in batches. For example, if the batch size is specified as *10*, every SOAP request sent to the configured SPML Web service endpoints contains ten UDCIdentity messages.

   The batch size is mandatory. Only positive numbers are allowed. Increase the batch size only if all configured endpoints support batch requests. If you are not sure whether the SPML Web service endpoints support batch operations, it is safe to have the batch size as *1*.

3. Click **Save**.

4. Enter the following information under **SPML End Point Configuration**:

End Point URL	URL of the external SPML Web service endpoint. Required.
Username	Username for the SPML Web service endpoint
Password	Password for the SPML Web service endpoint
Authentication Realm	Realm security for the SPML Web service endpoint

5. Click **Save**.

Note

To add a new SPML endpoint, click **Add**. To delete an SPML endpoint, click **Delete**.

Using Identity Data Export Utilities

You can use the Identity Data Export Utilities administrative interface to perform the following tasks:

- "Assign a UDCIdentifer to persons in Banner"
- "Extract UDCIdentity information from the database"
- "Generate LDIF files"
- "Publish SPML files"
- "Perform file operations"

The administrative interface is accessed at the following URL:

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

Note

If the URL was changed during installation, the new URL must be specified.

Log in with the user name and password that were mapped, during installation, to the `udcIdentityAdmin` role (OAS) or the `udcAdminGroup` group (Oracle WebLogic).

Assign a UDCIdentifer to persons in Banner

Use the following steps to assign a UDCIdentifier to all persons in Banner who do not have one.

1. Select UDCIdentifier Assigner > Assigner from the menu bar on the Identity Data Export Utilities administrative interface. The UDCIdentifier Assigner page is displayed.

2. Click **Assign**.

   The UDCIdentifier Assignment process starts. A message displays the assignment start date and time. Once the process is completed, a message displays the last run process date and time.

Extract UDCIdentity information from the database

Use the following steps to extract UDCIdentity information from the database.

1. Select UDCIdentity Extractor > Extractor from the menu bar on the Identity Data Export Utilities administrative interface. The UDCIdentity Extractor page is displayed.

2. Enter the name of the UDCIdentityList.xml file in the **Provide UDCIdentity List file name** field.

   The file name is required and must have an xml extension. Provide the UDCIdentityList.xml file name of your choice.

3. Choose one of the following:

   3.1. To extract UDCIdentity information for all persons, select **Extract all identities**.

   3.2. To extract UDCIdentity information for selected persons, select **Extract identities based on a Population Selection.** Enter the application, selection

ID, creator, and User ID that identify the population selection. All entries must be upper case.

Note

If you are using a population section to extract selected persons, you must create the population selection first. Refer to the *Banner General User Guide* for information on creating a population selection. ■

4. Click **Extract**.

   The UDCIdentity Extract process starts. A message displays the extract start date and time, output file name, number of identities processed progressively, and total processing time. Once the extract process is completed, a message displays the last run process date and time, and total number of identities processed.

Generate LDIF files

Use the following steps to generate LDIF files.

1. Select LDIF Generator > Generator from the menu bar on the Identity Data Export Utilities administrative interface. The LDIF Generator page is displayed.

2. Select *UDCIdentityList.xml* file for LDIF conversion.

3. Enter the output file name in the **LDIF output file name** field.

   You can enter an LDIF output file name of your choice. The default file name is `default_ldif_file.ldif`.

4. Click **Generate**.

   The LDIF Generator process starts. A message displays the generation date and time. Once the process is completed, a message displays the last run process date, time, and total number of identities processed.

Publish SPML files

Use the following steps to publish SPML files.

Note

If you are using the SPML Publisher to publish a SPML file to an LDAP directory, the SPML LDAP Adapter or similar component should be deployed. See Chapter 10, "SPML LDAP Adapter". ■

1. Select SPML Publisher > Publisher from the menu bar on the Identity Data Export Utilities administrative interface. The SPML Publisher page is displayed.
2. Select the file that you want to publish.
3. Click **Publish**.

   The SPML Publisher process starts. A message displays the publishing date and time. Once the process is completed, a message displays the last run process date, time, total number of batches processed, and the number of batches that have failed. The failures of SPML Publisher are logged in `spml_publisher_failure.log`.

Perform file operations

Use the following steps to view, download, and delete the files generated by the components of the Identity Data Export Utilities application.

1. Select File Operations from the menu bar on the Identity Data Export Utilities administrative interface. The File Operations page is displayed.
2. Select the file that you want to download.
3. Choose one of the following:
   - 3.1. To view the file, click **Download** and then click **Open**.
   - 3.2. To save the file, click **Download** and then click **Save**.
   - 3.3. To delete the file, click **Delete**.

7 Banner Identity Gateway

The Banner® Identity Gateway synchronizes and reconciles Banner user identities. When a preconfigured identity entity in Banner is created, updated, or deleted, Oracle Streams publishes a BannerIdentity XML message on the Oracle AQ Topic. The Banner Identity Gateway subscribes to this topic and processes the identity event. For a given PIDM, the Banner Identity Gateway looks for the UDCIdentifier (the unique ID across the enterprise), creates the UDCIdentity message, and posts it to the UDCIdentity Topic.

This chapter gives instructions for installing the Gateway manually, configuring the Gateway, and performing administrative tasks.

Note

Before installing the Gateway, you should run a script that checks to see if your environment is ready for the installation. See "BEIS_Check_script" on page 5-1.

Installation options

You can either install the Gateway with a manual or automated process.

Manual installation

You can use the application server console to manually install the Gateway. The Gateway is packaged in two zip files for manual installation:

- `Deployables\OC4J\Banner_IdentityGateway_full_release_Oracle.zip` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\Banner_IdentityGateway_full_release_Weblogic.zip` is used for installation on Oracle WebLogic Server 11g.

This chapter gives instructions for a manual installation on both servers.

Automated installation

You can use the automated Banner Enterprise Identity Services Installer to install the Banner Identity Gateway and the Enterprise Identity Proxy Services in one installation process. If you are using a windows environment, the installer provides a graphical user interface. If you are using a non-windows environment, the installer provides a command line mode.

The installer is packaged in two archive files:

- `Deployables\OC4J\beis_oc4j_installer.jar` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\beis_weblogic_installer.jar` is used for installation on Oracle WebLogic Server 11g.

The installer can be used for new installations only; it cannot be used for upgrades. Refer to Chapter 9, “Automated Installer” for instructions on using the installer.

Installation on Oracle Application Server

`Deployables\OC4J\Banner_IdentityGateway_full_release_Oracle.zip` is used for installation on Oracle Application Server 10.1.3.4/5. This zip file contains an archive file named `bnig.ear`.

The Gateway can be installed on an existing Oracle Application Server. The Gateway and Identity Proxy must be installed together in the same OC4J instance. BEIS applications should be deployed separately from other applications so they can be managed independently.

Note

The following steps are used for a manual installation of the Gateway. If you prefer to use the automated installer, refer to Chapter 9, “Automated Installer”.

Use the following steps to install the Banner Identity Gateway on OAS 10.1.3.4/5:

- Step 1, “Configure the database user and schema”
- Step 2, “Configure the integmgr user password”
- Step 3, “Define the data source for Oracle Advanced Queuing”
- Step 4, “Define the data source for the Banner Identity Gateway”
- Step 5, “Define the data source for the Oracle Streams administrator”
- Step 6, “Define the data source for the Banner security administrator”
- Step 7, “Configure the security role and user”
- Step 8, “Configure the JMS queues and topics”
- Step 9, “Deploy the Banner Identity Gateway”
- Step 10, “Configure logging”

Step 1 Configure the database user and schema

Use the following steps to create the database user, tables, and packages that are required by the Banner Identity Gateway.

1. Extract the `Banner_IdentityGateway_full_release_Oracle.zip` archive file.

2. Go to the directory where the file was extracted. This directory is referred to as `<INSTALLER_HOME>`.

   ```
   cd <INSTALLER_HOME>/db-scripts/users
   ```

3. Run SQL*Plus and connect as DBA.

4. Execute the `bnixmgr.sql` script:

   ```
   sqlplus> @bnixmgr
   ```

5. When prompted, enter the following information:

   - Schema name for the Banner Identity Gateway (*bnixmgr*)
   - Tablespace name for the Banner Identity Gateway schema (for example, *bnixmgr_tb*)
   - Name of the datafile with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0/dbs/bnig.dbf*)
   - Password for the Banner Identity Gateway schema (for example, *u_pick_it*)

6. Execute the following scripts:

   ```
   sqlplus> grant aq_administrator_role to integmgr;
   sqlplus> alter user integmgr default role
   aq_administrator_role;
   ```

7. Close SQL*Plus.

   ```
   sqlplus> exit
   ```

8. Change the directory to `<INSTALLER_HOME>/db-scripts/tables`.

   ```
   cd <INSTALLER_HOME>/db-scripts/tables
   ```

9. Run SQL*Plus and connect as the `bnixmgr` user.

10. Execute the following script to create the tables:

```
sqlplus> @setup
sqlplus> exit
```

11. Change the directory to <INSTALLER_HOME>/db-scripts/packages.

12. Run SQL*Plus and connect as the bnixmgr user.

13. Execute the following script to create the database packages:

```
sqlplus> @iokp_build
```

Step 2 Configure the integmgr user password

The Gateway manages its own database connection, refreshing it periodically to ensure optimal performance. To accomplish this refresh, the Gateway must be configured for the password of the integmgr user in your environment. Use the following steps to configure the password.

1. Copy bnig.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 bnig.ear
   ```

   The extract contains a folder named lib.

3. Navigate to the lib folder and execute the following command:

   ```
   jar xvf bnigCore.jar BnigResources.properties
   ```

4. Open BnigResources.properties, which is extracted under the lib folder.

5. Edit the last two properties as follows:

Property	Description
integmgr.pwd	Password for the integmgr schema for your environment. Required.
connection.refresh.count	Frequency for creating a new connection. A new connection is created every time this number of message sets is processed. Default is 10000.

6. Save the changes.

7. From the `lib` folder execute the following command to rebuild `bnigCore.jar`:

   ```
   jar uvf bnigCore.jar BnigResources.properties
   ```

8. After this command runs successfully, delete the `BnigResources.properties` file under the `lib` folder.

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

   ```
   jar cvf bnig.ear *.war *.jar META-INF/* lib/*
   ```

The rebuilt `bnig.ear` is used for installation.

Step 3 Define the data source for Oracle Advanced Queuing

A data source provides the connection properties to the Banner database. Use the following steps to define the data source that the Banner Identity Gateway uses to connect to Oracle Advanced Queueing to consume `banner_identity` messages.

1. Connect to the Oracle Enterprise Manager:

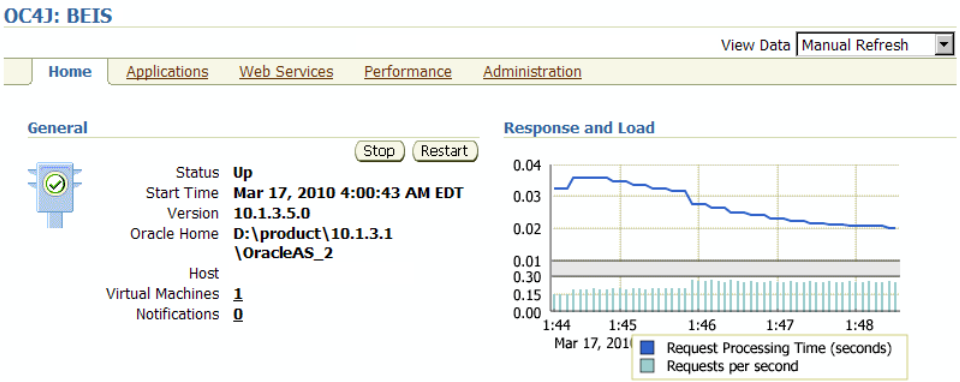
   ```
   http://<host>:<port>/em
   ```

   The console is displayed.

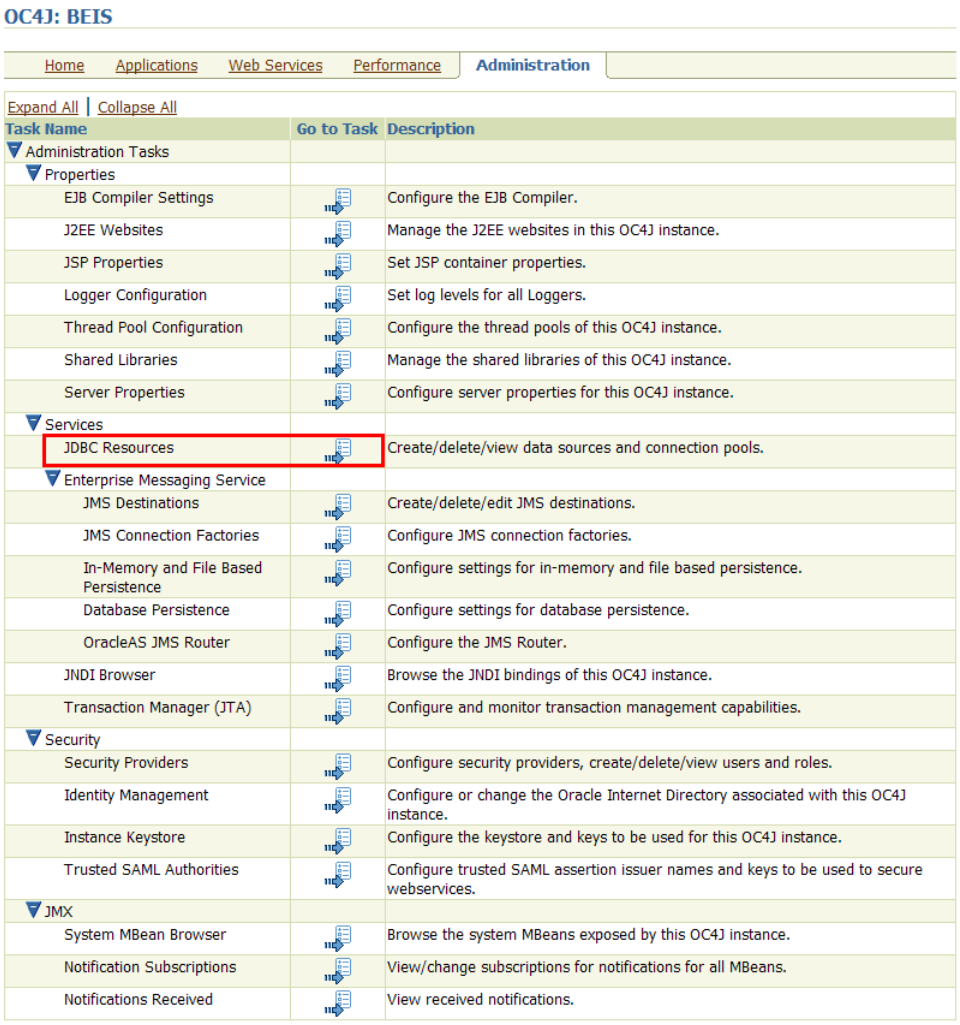
2. Click the name of the OC4J instance that will host the Banner Identity Gateway.

   The application must be installed in the same instance with the Enterprise Identity Proxy Services application. Banner Enterprise Identity Services (BEIS) applications must be installed separately from other applications so they can be independently managed.

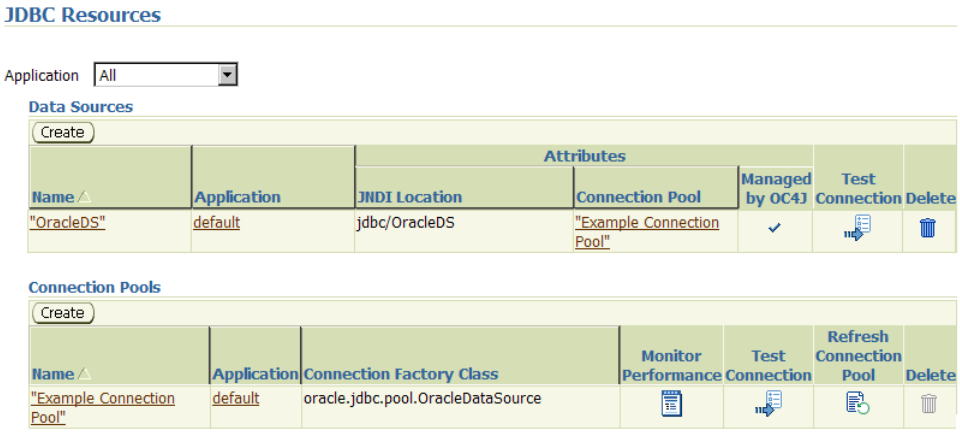
   The Home page for the selected instance is displayed.



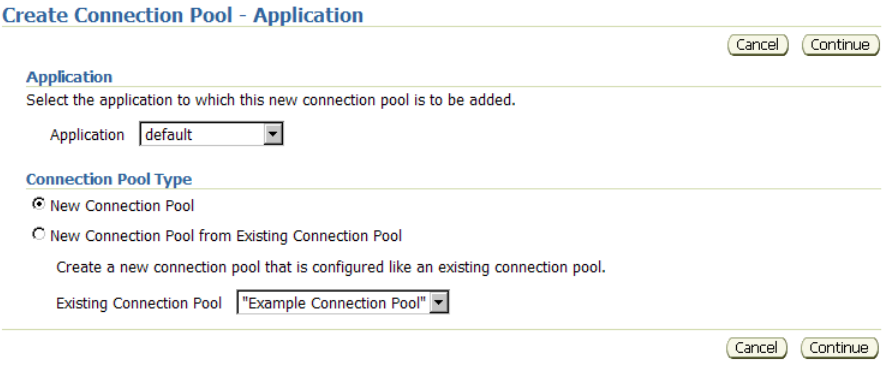
3. Select the **Administration** tab. A list of tasks is displayed.



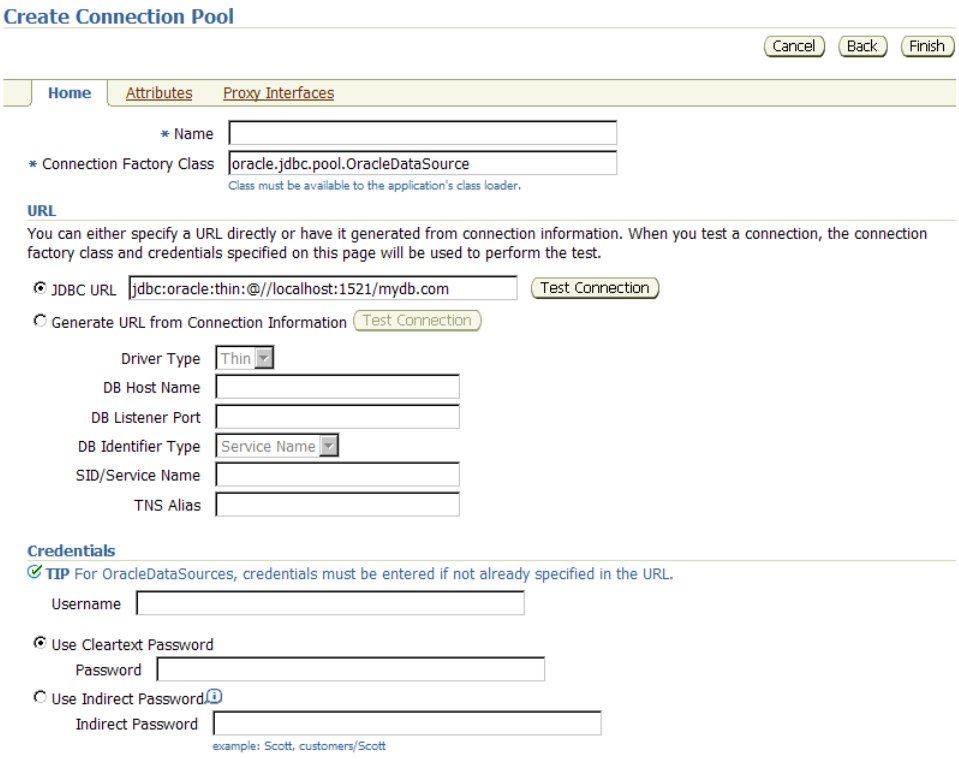
4. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.



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



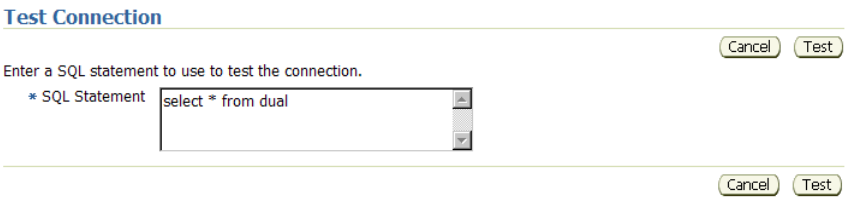
6. Click **Continue**. The Create Connection Pool page is displayed.



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

Name	*integmgr_banner_pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*integmgr*
Use Cleartext Password	Select **Use Cleartext Password** and enter a password for the `integmgr` schema.

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



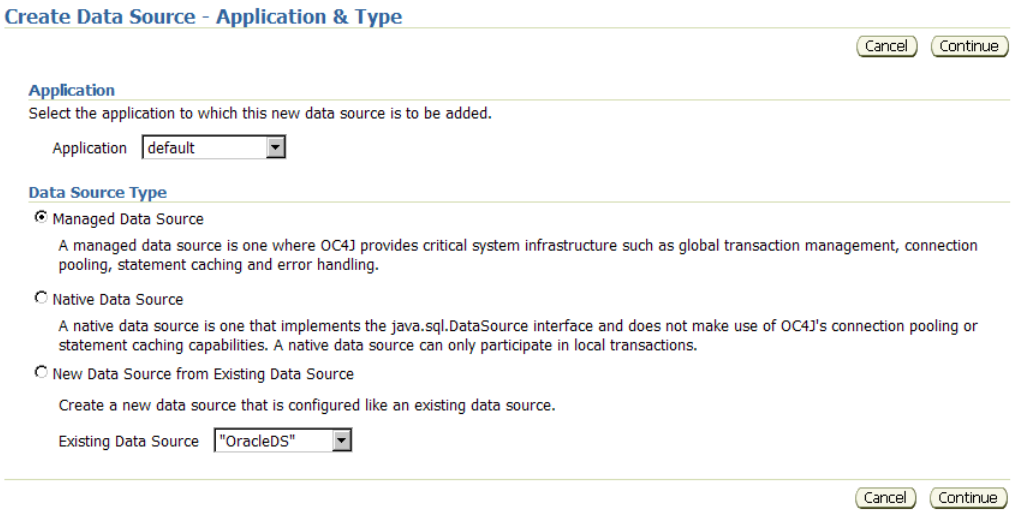
9. Click **Test** to test the connection pool for the `integmgr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

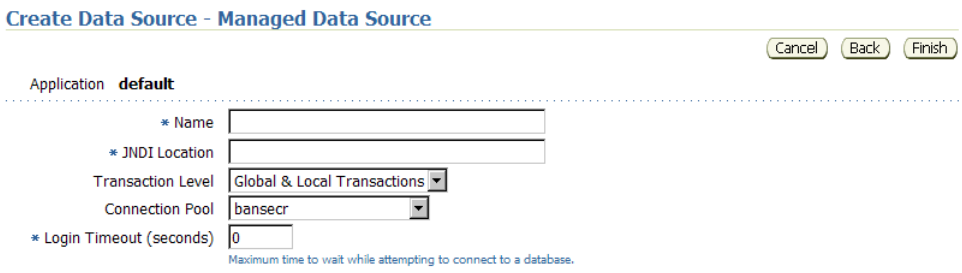
   9.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

10. Click **Finish**.

11. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.



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



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

Name	*Integmgr_banner*
JNDI Location	*jdbc/integmgr_banner*
Connection Pool	*integmgr_banner_pool*

14. Click **Finish**.

Step 4 Define the data source for the Banner Identity Gateway

Use the following steps to define the data source that is used to access the database schema that is created for the Banner Identity Gateway to store processing parameters.

1. Select the **Administration** tab. A list of tasks is displayed.
2. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.
3. Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.
4. Click **Continue**. The Create Connection Pool page is displayed.
5. Enter the following information to set up the connection pool for the `bnixmgr` schema:

Name	*bnixmgr_banner_pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*bnixmgr*
Use Cleartext Password	Select **Use Cleartext Password** and provide a password for the `bnixmgr` schema.

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

7. Click **Test** to test the connection pool for the `bnixmgr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

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

8. Click **Finish**.

9. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

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

11. Enter the following information to set up the `bnixmgr` data source:

Name	*bnixmgr*
JNDI Location	*jdbc/bnixmgr*
Connection Pool	*bnixmgr_banner_pool*

12. 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 viewing and administering the IAM Oracle Streams environment in Banner.

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

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

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

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

5. Enter the following information to set up the connection pool for the `streamsadmin` schema:

Name	*streamsadmin_banner_pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*streamsadmin*
Use Cleartext Password	Select **Use Cleartext Password** and provide a password for the `streamsadmin` schema.

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

7. Click **Test** to test the connection pool for the `streamsadmin` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

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

8. Click **Finish**.

9. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

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

11. Enter the following information to set up the `streamsadmin` data source:

Name	*streamsadmin*
JNDI Location	*jdbc/streamsadmin*
Connection Pool	*streamsadmin_banner_pool*

12. Click **Finish**.

Step 6 Define the data source for the Banner security administrator

Use the following steps to define the data source for the Banner security administrator.

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

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

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

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

5. Enter the following information to set up the connection pool for the `bansecr` schema:

Name	*inbadmin_banner_pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*bansecr*
Use Cleartext Password	Select **Use Cleartext Password** and provide a password for the `bansecr` schema.

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

7. Click **Test** to test the connection pool for the `bansecr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

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

8. Click **Finish**.

9. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

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

11. Enter the following information to set up the `inbadmin` data source:

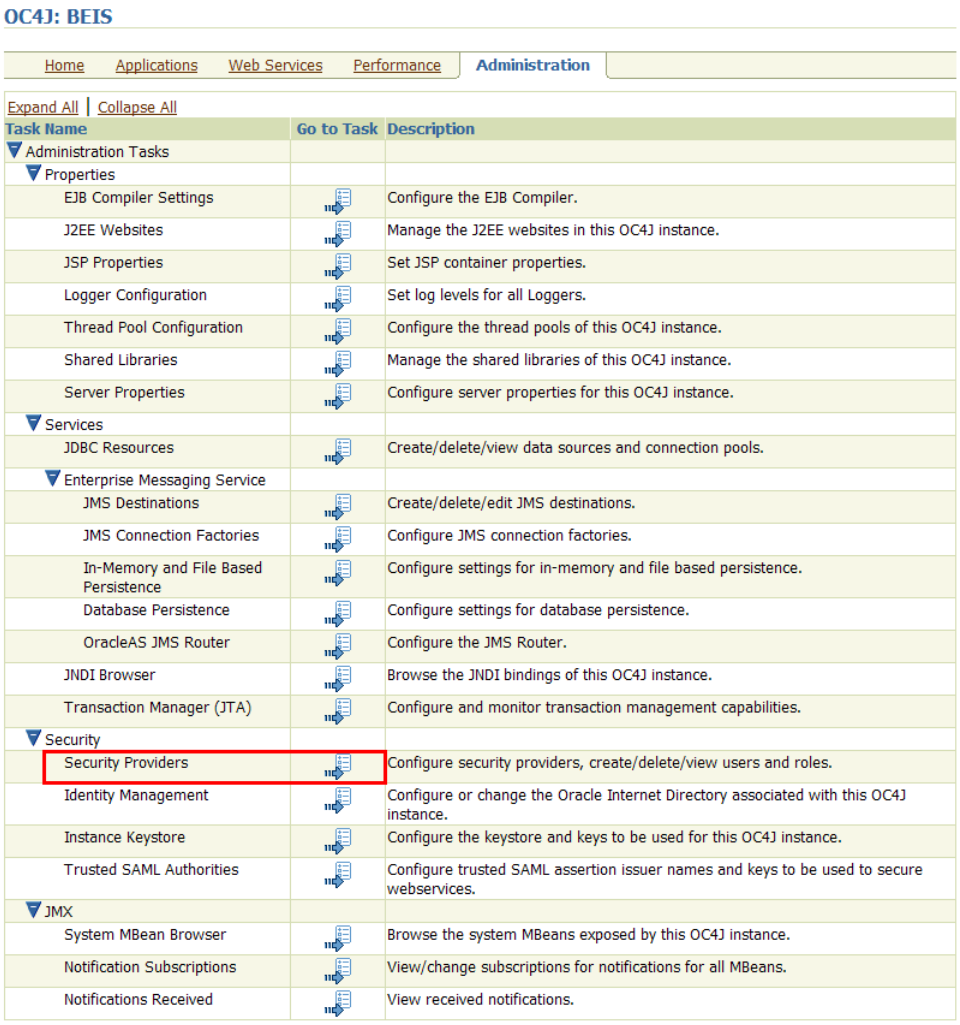
Name	*inbadmin*
JNDI Location	*jdbc/inbadmin*
Connection Pool	*inbadmin_banner_pool*

12. Click **Finish**.

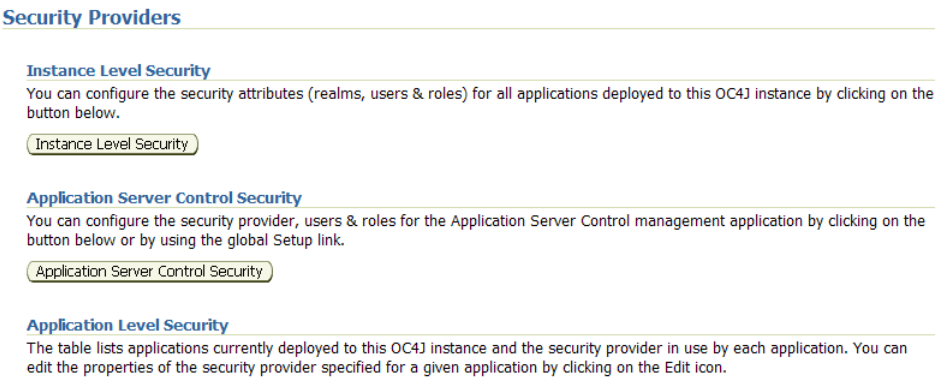
Step 7 Configure the security role and user

Use the following steps to add the `bnixadmin` role and an administrative user to the Banner Identity Gateway application. This role and user are required for accessing the Banner Identity Gateway administrative interface and for invoking the Banner PSP Web services.

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

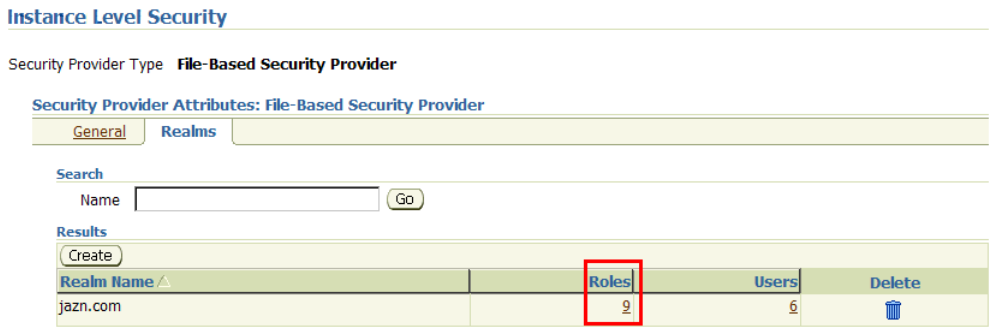


2. Select **Security Providers** in the Security section. The Security Providers page is displayed.



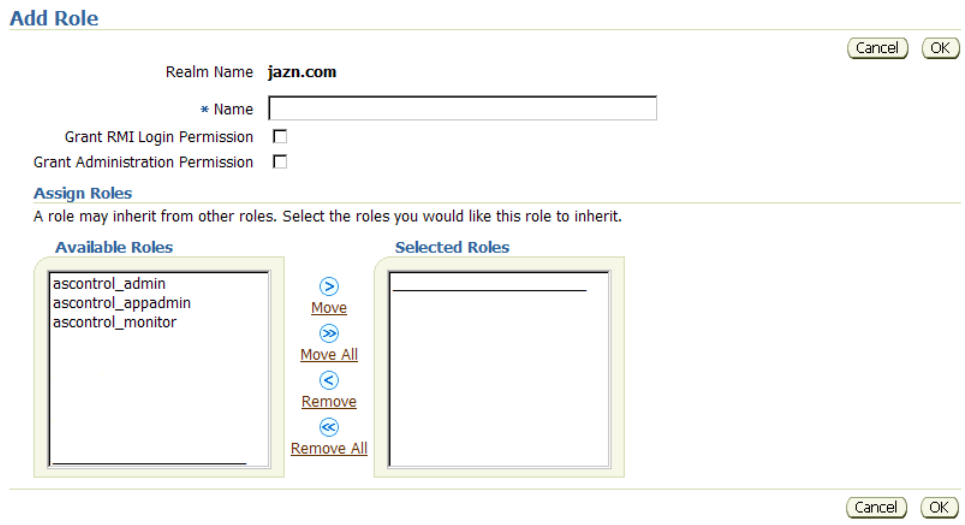
3. Click **Instance Level Security**. The Instance Level Security page is displayed.

4. Select the **Realms** tab.



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

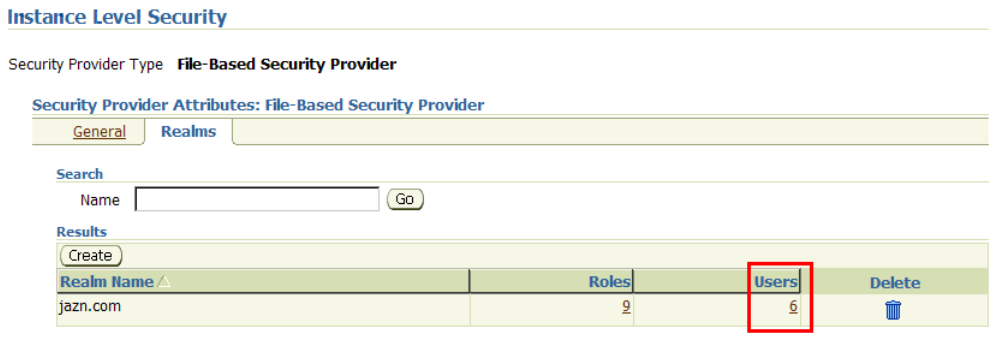
6. Click **Create**. The Add Role page is displayed.



7. Enter *bnixadmin* in the **Name** field.

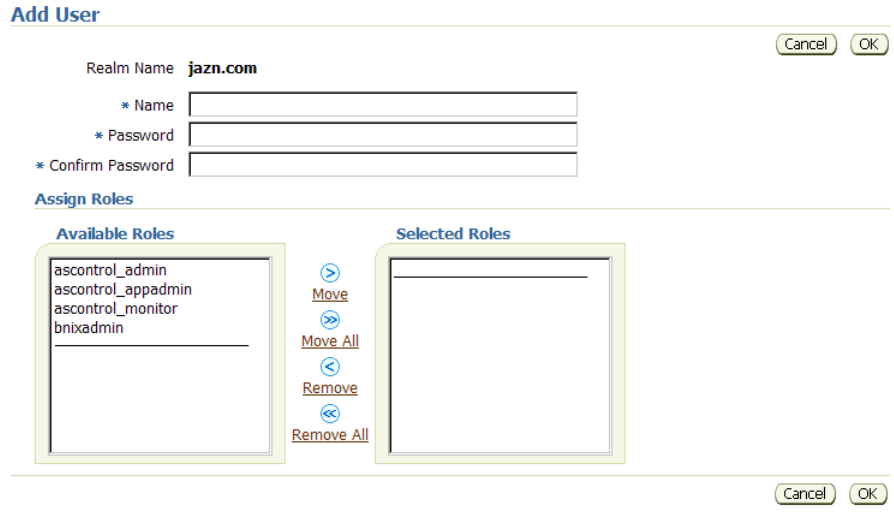
8. Click **OK**. The Roles page is redisplayed with the new role.

9. Return to the Instance Level Security page.



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

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



12. Enter the following information to create a user:

Name	*bnix* (This is an example. Enter the name of your choice.)
Password	Password used by the user to log in to the Banner Identity Gateway administrative interface
Confirm Password	Confirmation of the password

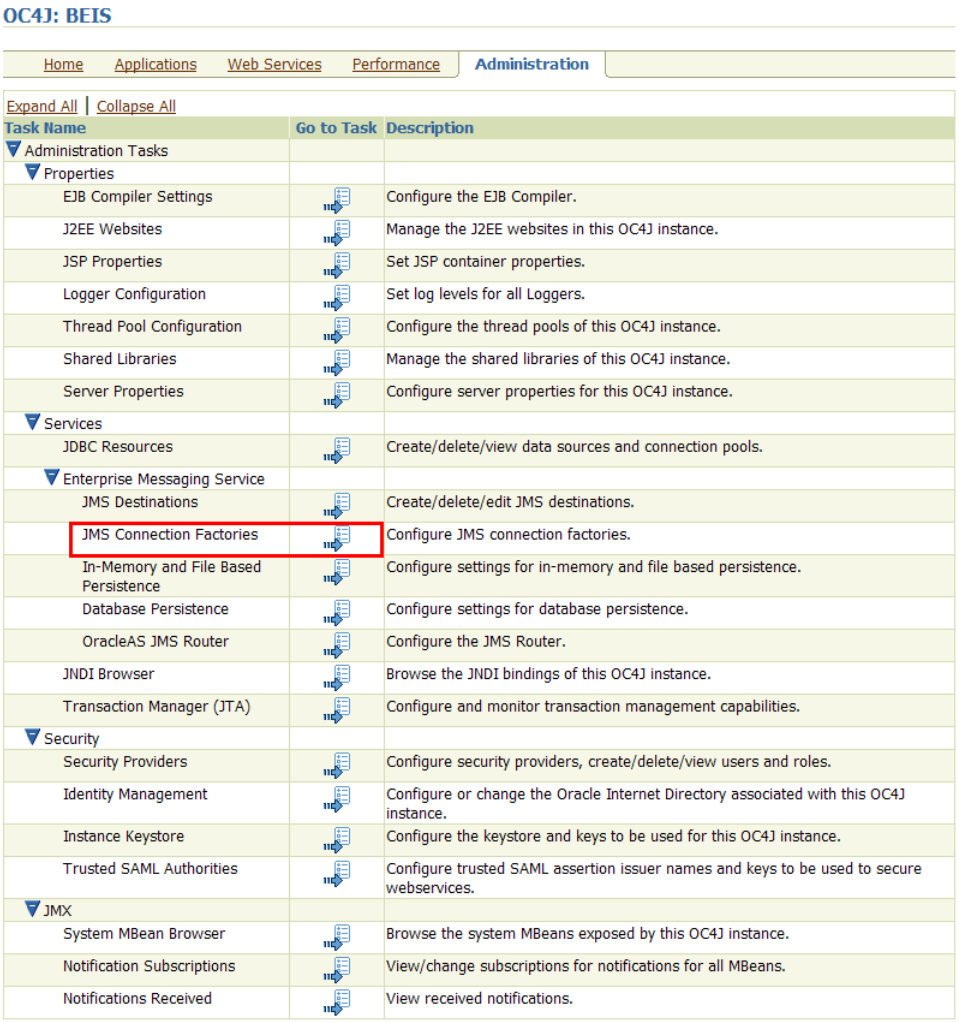
13. In the Assign Roles section, select the *bnixadmin* role in the **Available Roles** list and move it to the **Selected Roles** list.

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

Step 8 Configure the JMS queues and topics

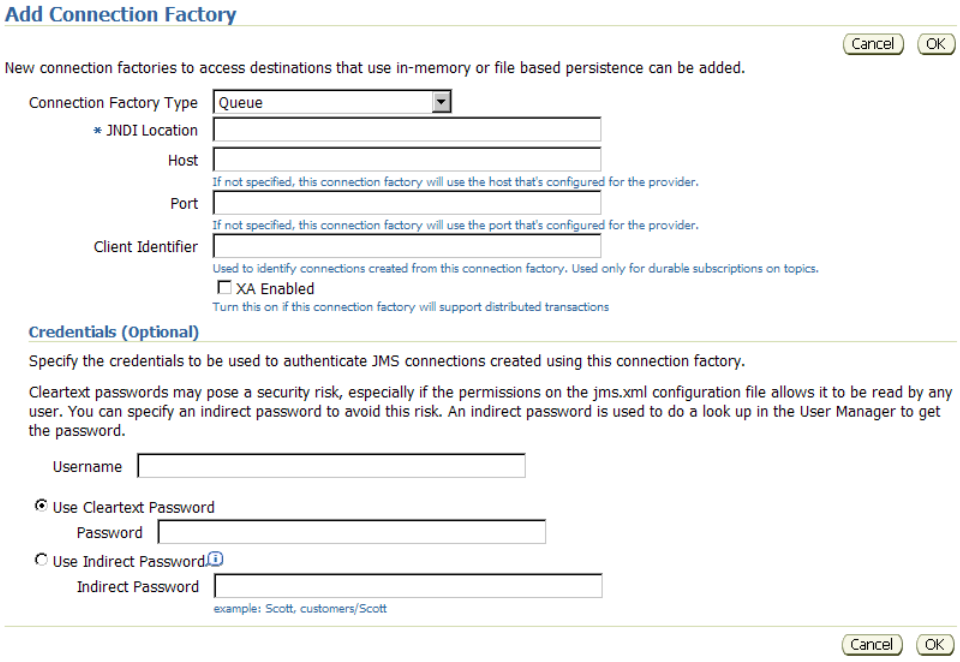
The BEIS application uses Java messaging to move UDCIdentity messages among its various services. Use the following steps to add JMS queues and topics to the Oracle Application Server for the Banner Identity Gateway.

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



2. Select **JMS Connection Factories** in the Enterprise Messaging Service section. The JMS Connection Factories page is displayed.

3. Click **Create New**. The Add Connection Factory page is displayed.

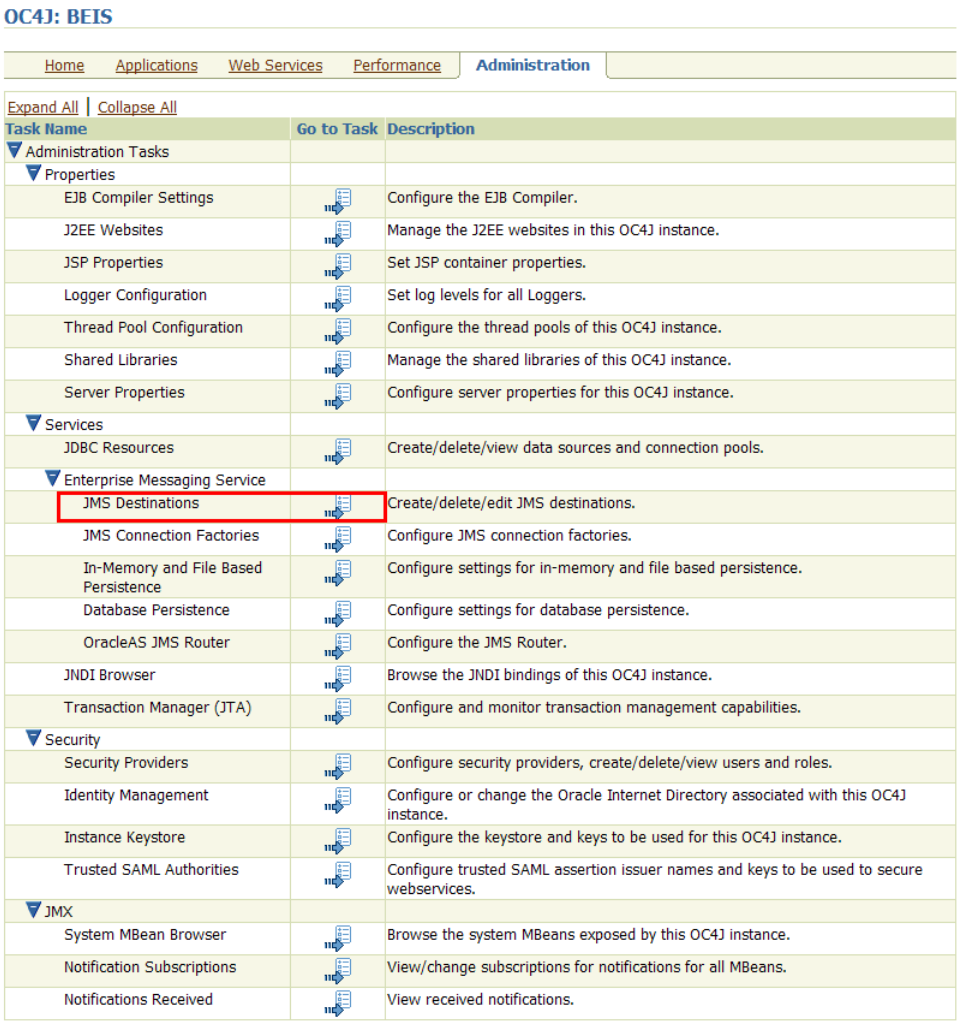


4. Enter the following information to create a connection factory:

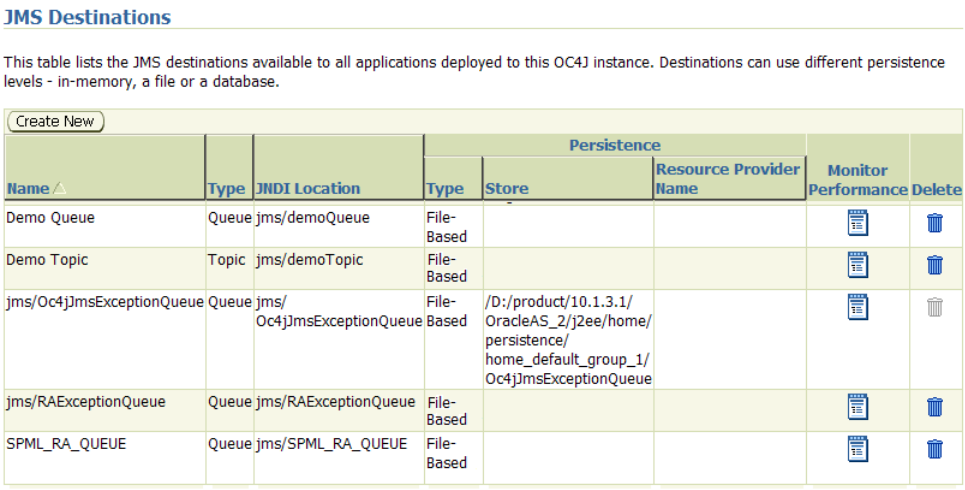
Connection Factory Type	*Topic*
JNDI Location	*jms/UDC_IDENTITY_TCF*
Host	*[ALL]*
Client Identifier	Leave blank
XA Enabled	checked (yes)

5. Click **OK**. The JMS Connection Factories page is redisplayed.

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



7. Select **JMS Destinations** in the Enterprise Messaging Service section. The JMS Destinations page is displayed.



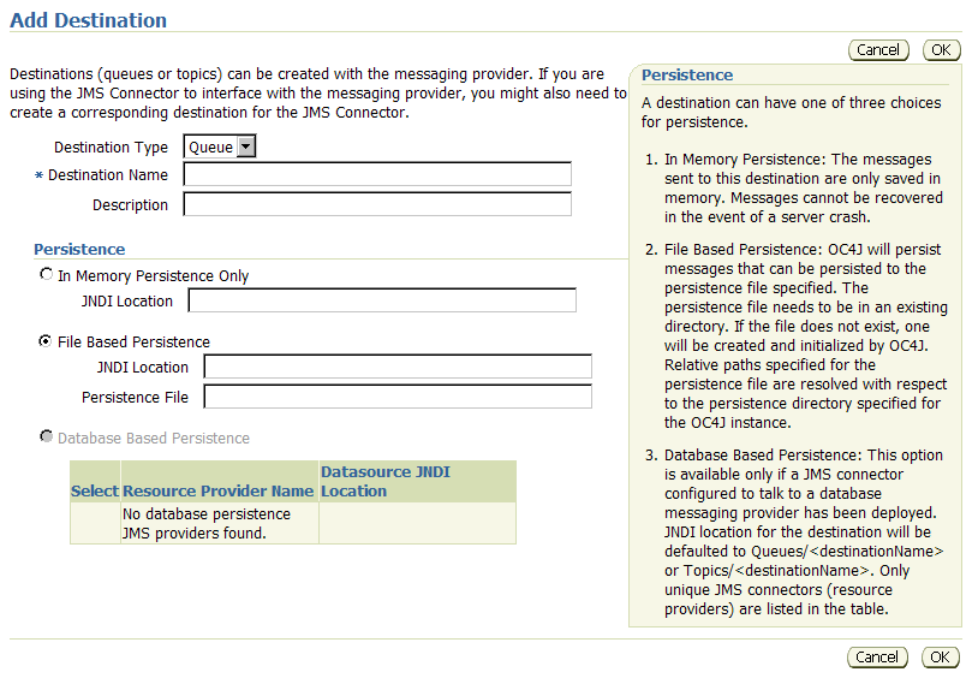
JMS Destinations

This table lists the JMS destinations available to all applications deployed to this OC4J instance. Destinations can use different persistence levels - in-memory, a file or a database.

Create New

Name	Type	JNDI Location	Persistence			Monitor Performance	Delete
			Type	Store	Resource Provider Name		
Demo Queue	Queue	jms/demoQueue	File-Based				
Demo Topic	Topic	jms/demoTopic	File-Based				
jms/Oc4jJmsExceptionQueue	Queue	jms/Oc4jJmsExceptionQueue	File-Based	/D:/product/10.1.3.1/OracleAS_2/j2ee/home/persistence/home_default_group_1/Oc4jJmsExceptionQueue			
jms/RAExceptionQueue	Queue	jms/RAExceptionQueue	File-Based				
SPML_RA_QUEUE	Queue	jms/SPML_RA_QUEUE	File-Based				

8. Click **Create New**. The Add Destination page is displayed.



Add Destination

Cancel OK

Destinations (queues or topics) can be created with the messaging provider. If you are using the JMS Connector to interface with the messaging provider, you might also need to create a corresponding destination for the JMS Connector.

Destination Type: Queue

* Destination Name:

Description:

Persistence

In Memory Persistence Only

JNDI Location:

File Based Persistence

JNDI Location:

Persistence File:

Database Based Persistence

Select	Resource Provider Name	Datasource JNDI Location
	No database persistence JMS providers found.	

Persistence

A destination can have one of three choices for persistence.

1. In Memory Persistence: The messages sent to this destination are only saved in memory. Messages cannot be recovered in the event of a server crash.
2. File Based Persistence: OC4J will persist messages that can be persisted to the persistence file specified. The persistence file needs to be in an existing directory. If the file does not exist, one will be created and initialized by OC4J. Relative paths specified for the persistence file are resolved with respect to the persistence directory specified for the OC4J instance.
3. Database Based Persistence: This option is available only if a JMS connector configured to talk to a database messaging provider has been deployed. JNDI location for the destination will be defaulted to Queues/<destinationName> or Topics/<destinationName>. Only unique JMS connectors (resource providers) are listed in the table.

Cancel OK

9. Enter the following information to add a JMS destination:

Destination Type *Topic*

Destination Name *UDC_IDENTITY_TOPIC*

Description *UDC Topic*

10. Select **File Based Persistence** and enter the following information:

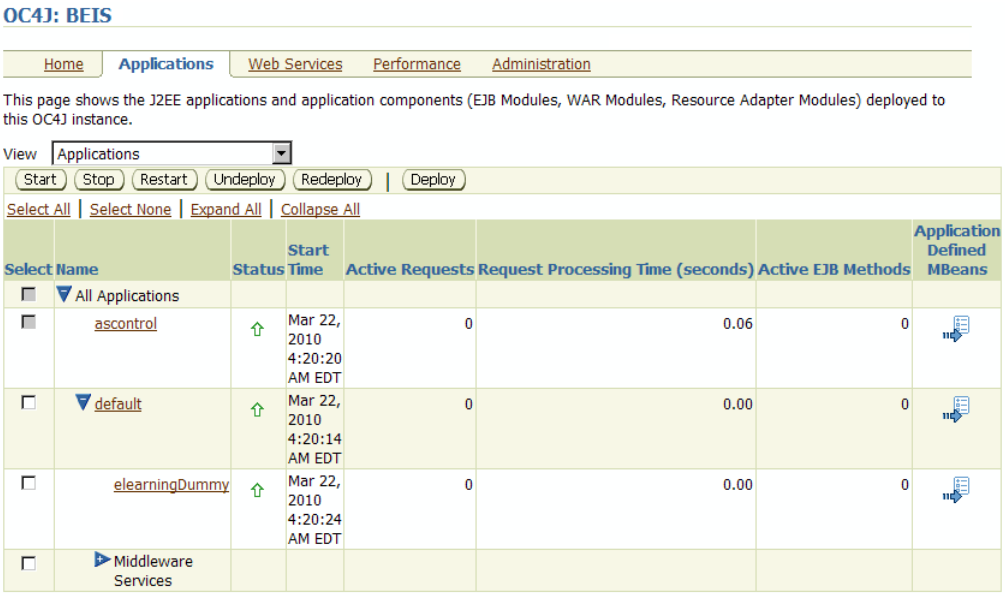
JNDI Location	*jms/UDC_IDENTITY_TOPIC*
Persistence File	*UDC_IDENTITY_TOPIC*

11. Click **OK**. The JMS Destinations page is redisplayed.

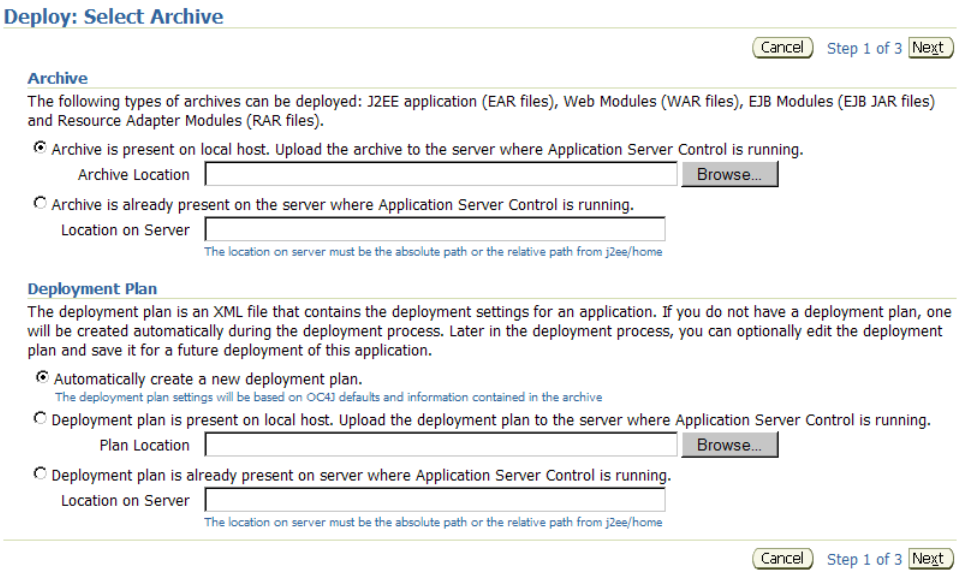
Step 9 Deploy the Banner Identity Gateway

Use the following steps to deploy the Banner Identity Gateway to the Oracle Application Server.

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



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



3. Select the file to be uploaded:

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

   3.2. In the **Archive Location** field, click **Browse** and navigate to the `bnig.ear` file.

   3.3. Select the file and click **Open**.

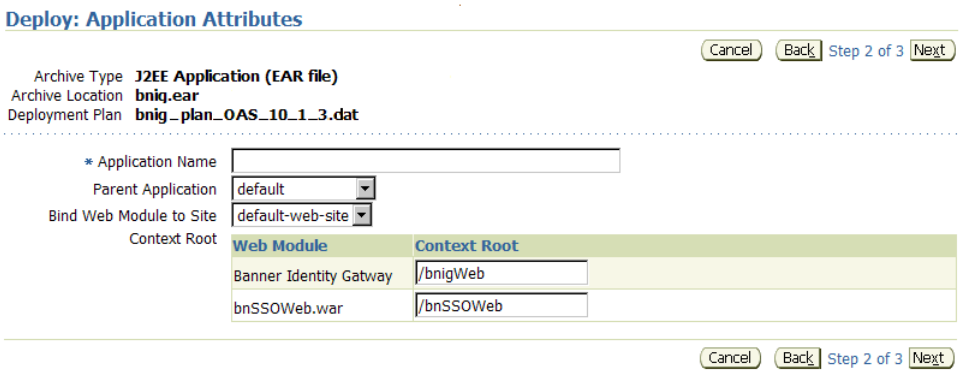
4. Select the deployment plan for the application:

   4.1. 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.**

   4.2. In the **Plan Location** field, click **Browse** and navigate to the `bnig_plan_OAS_10_1_3.dat` file.

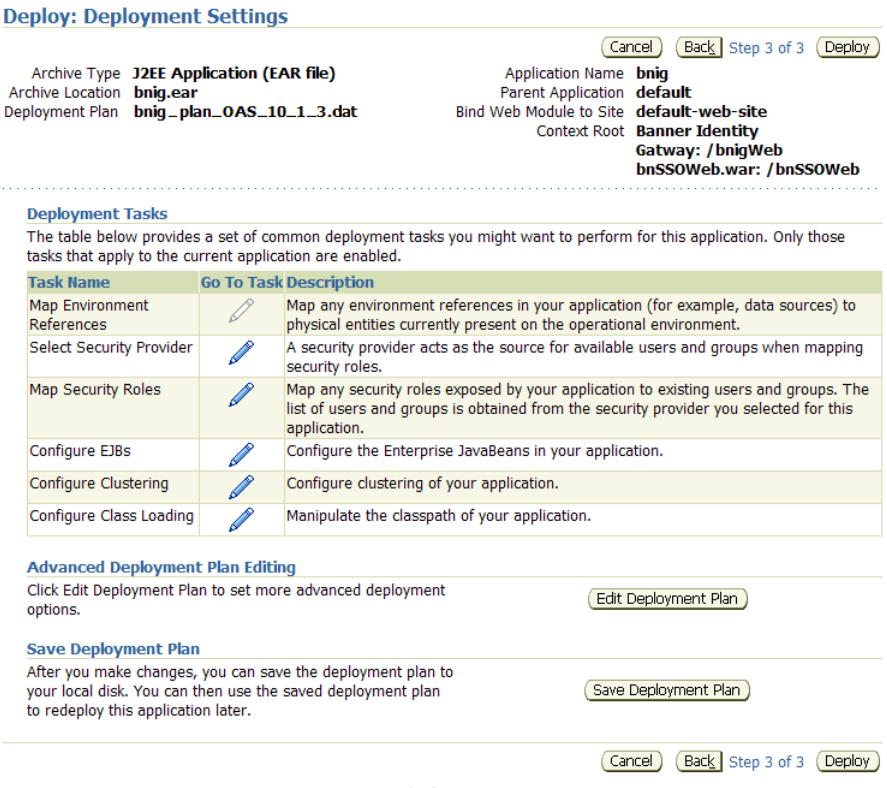
   4.3. Select the file and click **Open**.

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



6. Enter a name for the application (for example, *bnig*) in the **Application Name** field.

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



8. Click **Deploy** to accept the values and install the Banner Identity Gateway. A deployment confirmation page is displayed.

9. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

Step 10 Configure logging

The Banner Identity Gateway application uses 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 file**. The default location is `<OAS_HOME>/j2ee/home/bnig_application.log`. This location should be changed to the OC4J instance where the Banner Identity Gateway is installed.
- **Logging level**. The default level is *INFO*, resulting in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide detailed logging, you should modify the log4j configurations.

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

1. Navigate to `<OAS_HOME>/j2ee/<OC4J instance>/applications/bnig/bnigWeb/WEB-INF/classes`.
2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
`log4j.appender.out.File`	`bnig_application.log`	`../<OC4J instance>/log/ bnig_application.log`
`log4j.rootCategory`	`INFO,out`	`DEBUG,out`

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

Installation on Oracle WebLogic Server 11g

`Deployables\Weblogic\Banner_IdentityGateway_full_release_Weblogic.zip` is used for installation on Oracle WebLogic Server 11g. This zip file contains an archive file named `bnig.ear`.

Recommended configuration

The Gateway must be installed in an Oracle WebLogic Basic Domain Managed Server with the Enterprise Identity Proxy Services component. These components must be

installed in the same Managed Server. They must not be installed in an Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual server for BEIS and other middle-tier components. This server would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server instance should consist of the default Admin Server and at least two Managed Servers:

- One Managed Server for the Gateway and the Identity Proxy, which must be installed together
- One Managed Server for the Identity Data Export Utilities and the SSO Manager

If a domain based on the Basic Domain template already exists for middle-tier applications, the BEIS components can be installed in two separate Managed Servers in that domain, based on the preceding recommendation.

Refer to the Oracle WebLogic Server Documentation Library for details on creating a new domain and a new Managed Server.

Installation steps

Note

The following steps are used for a manual installation of the Gateway. If you prefer to use the automated installer, refer to Chapter 9, “Automated Installer”.

Use the following steps to install the Gateway on Oracle WebLogic Server 11g (version 10.3.2).

- Step 1, “Configure the database user and schema”
- Step 2, “Customize properties”
- Step 3, “Configure the authentication provider”
- Step 4, “Define the data source for Oracle Advanced Queuing”
- Step 5, “Define the data source for the Banner Identity Gateway”
- Step 6, “Define the data source for the Oracle Streams administrator”
- Step 7, “Define the data source for the Banner security administrator”
- Step 8, “Configure the security group and user”
- Step 9, “Create a JMS server”
- Step 10, “Create a JMS module”

- Step 11, “Configure a JMS topic and connection factory”
- Step 12, “Deploy the Banner Identity Gateway”

Step 1 Configure the database user and schema

Use the following steps to create the database user, tables, and packages required by the Banner Identity Gateway.

1. Extract the `Banner_IdentityGateway_full_release_WebLogic.zip` archive file.

2. Go to the directory where the file was extracted. This directory is referred to as `<INSTALLER_HOME>`.

   ```
   cd <INSTALLER_HOME>/db-scripts/users
   ```

3. Run SQL*Plus and connect as DBA.

4. Execute the `bnixmgr.sql` script:

   ```
   sqlplus> @bnixmgr
   ```

5. When prompted, enter the following information:

   - Schema name for the Banner Identity Gateway (*bnixmgr*)
   - Tablespace name for the Banner Identity Gateway schema (for example, *bnixmgr_tb*)
   - Name of the datafile with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0/dbs/bnig.dbf*)
   - Password for the Banner Identity Gateway schema (for example, *u_pick_it*)

6. Execute the following scripts:

   ```
   sqlplus> grant aq_administrator_role to integmgr;
   sqlplus> alter user integmgr default role
   aq_administrator_role;
   ```

7. Close SQL*Plus.

   ```
   sqlplus> exit
   ```

8. Change the directory to `<INSTALLER_HOME>/db-scripts/tables`.

   ```
   cd <INSTALLER_HOME>/db-scripts/tables
   ```

9. Run SQL*Plus and connect as the `bnixmgr` user.

10. Execute the following scripts to create the tables:

```
sqlplus> @setup
sqlplus> exit
```

11. Change the directory to <`INSTALLER_HOME`>`/db-scripts/packages`.

12. Run SQL*Plus and connect as the `bnixmgr` user.

13. Execute the following script to create the database packages:

```
sqlplus> @iokp_build
```

Step 2 Customize properties

The following properties can be customized:

- **Logging** - The Gateway uses Apache's log4j to log the activities performed by the application at runtime. The `bnig_application.log` file is located at the following location:

  `Oracle\Middleware\user_projects\domains\<domain_name>`

  where <`domain_name`> is the name of the domain where the application is installed. This location cannot be changed.

  A property in the `log4j.properties` file determines the logging level. The default logging level is *INFO*, which results in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. You can modify the logging level if you want more detailed logging.

- **integmgr user password** - The Gateway manages its own database connection, refreshing it periodically to ensure optimal performance. To accomplish this refresh, the Gateway must be configured for the password of the `integmgr` user in your environment.

Use the following steps to customize logging and the INTEGMGR user password.

1. Copy `bnig.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 bnig.ear
   ```

   The extract contains a Web archive named `bnigWeb.war` and a folder named `lib`.

3. Use the following steps to customize logging:

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

   3.2. Navigate to `war_home` and execute the following command:

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

   3.3. Open `war_home/WEB-INF/classes/log4j.properties`.

   3.4. Change the `log4j.rootCategory` property as follows:

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

   3.5. Save the change.

   3.6. From `war_home` execute the following command to rebuild the Web archive file:

   ```
   jar cvf <EAR_HOME>/bnigWeb.war *
   ```

4. Use the following steps to customize the `integmgr` user password:

   4.1. Navigate to the `lib` folder and execute the following command:

   ```
   jar xvf bnigCore.jar BnigResources.properties
   ```

   4.2. Open `BnigResources.properties`, which is extracted under the `lib` folder.

   4.3. Edit the last two properties as follows:

Property	Description
`integmgr.pwd`	Password for the `integmgr` schema for your environment. Required.
`connection.refresh.count`	Frequency for creating a new connection. A new connection is created every time this number of message sets is processed. Default is 10000.

   4.4. Save the changes.

   4.5. From the `lib` folder execute the following command to rebuild `bnigCore.jar`:

   ```
   jar uvf bnigCore.jar BnigResources.properties
   ```

4.6. After this command runs successfully, delete the BnigResources.properties file under the lib folder.

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

```
jar cvf bnig.ear *.war *.jar META-INF/* lib/*
```

The rebuilt bnig.ear is used for installation.

Step 3 Configure the authentication provider

An authentication provider must be configured in Oracle WebLogic Server 11g to allow for basic authentication against the Web services that the Gateway exposes. The authentication provider is set via a JAAS configuration file. The Oracle WebLogic Managed Server where the Gateway will be deployed must be configured to load the configuration file on startup. Use the following steps to configure the authentication provider.

1. If a JAAS configuration file already exists for the Oracle WebLogic domain where the Gateway will be deployed, skip to step 3.

   If a JAAS configuration file does not exist for the Oracle WebLogic domain where the Gateway will be deployed, use a text editor to create the jaas.config file with the following content:

```
myrealm {
   weblogic.security.auth.login.UsernamePasswordLoginModule
   REQUIRED;
};
```

2. Save jaas.config in the following location:

```
<WebLogic Home>/user_projects/domains/<your domain dir>/config
/security
```

   where <WebLogic Home> is the base directory for the Oracle WebLogic software packages and configuration files, and <your domain dir> is the domain where the Gateway will be deployed.

3. Configure the Managed Server to use the authentication provider.

   There are two ways to configure the Managed Server, depending on how you want to start the Managed Server. Use option 1 (page 7-32) if the Managed Server will be started by using the Oracle WebLogic Administration Console. Use option 2 (page 7-36) if the Managed Server will be started by running a script.

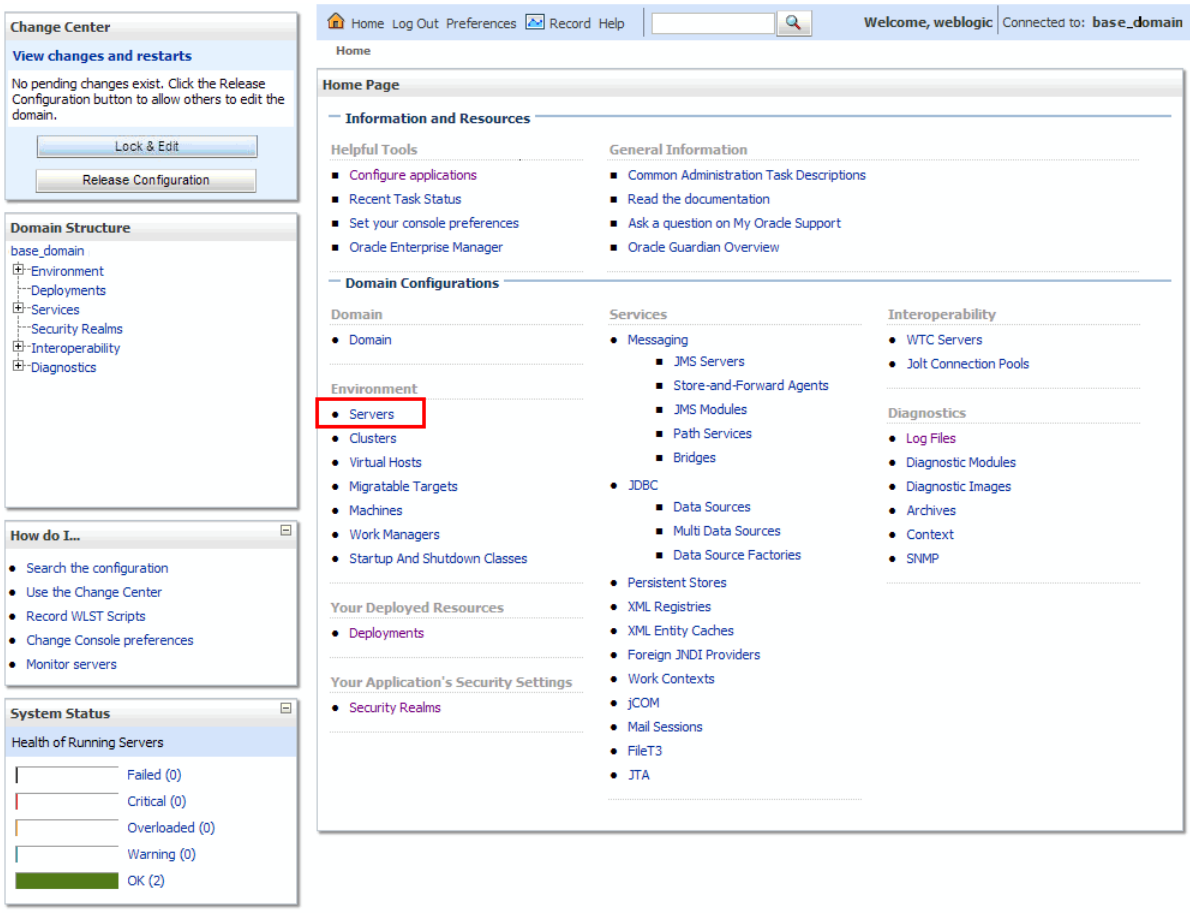
Option 1 - If you are using the administration console to start the Managed Server

With this option, the location of the JAAS configuration file is set as an argument on the Server Start tab of the specific Managed Server. The location of the JAAS configuration file applies only to that specific Managed Server.

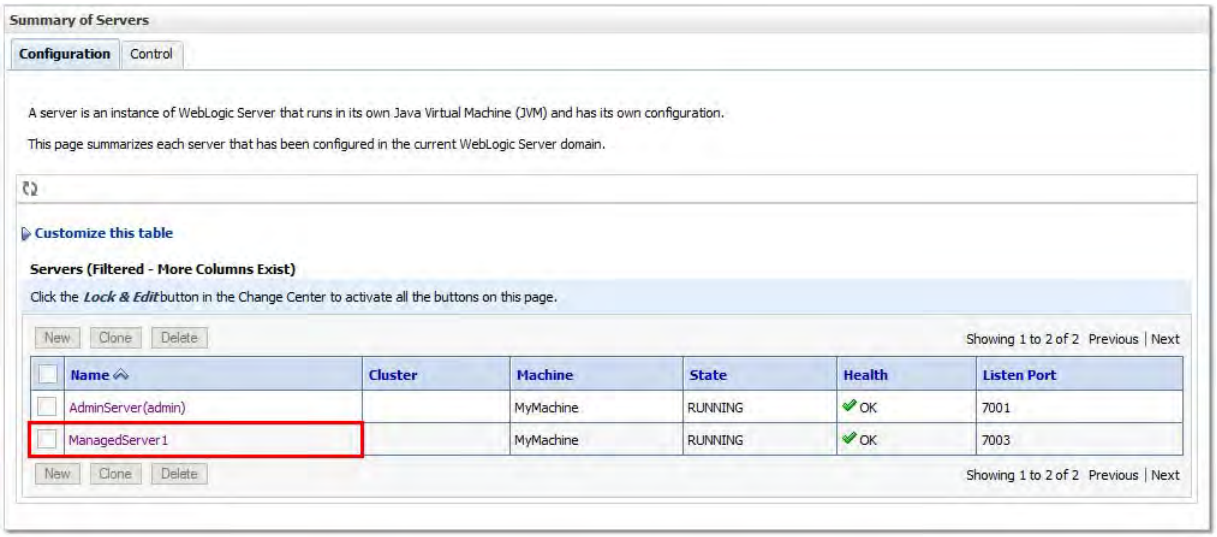
3.1. Connect to the Oracle WebLogic Server Administration Console for the domain where the Gateway will be deployed:

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

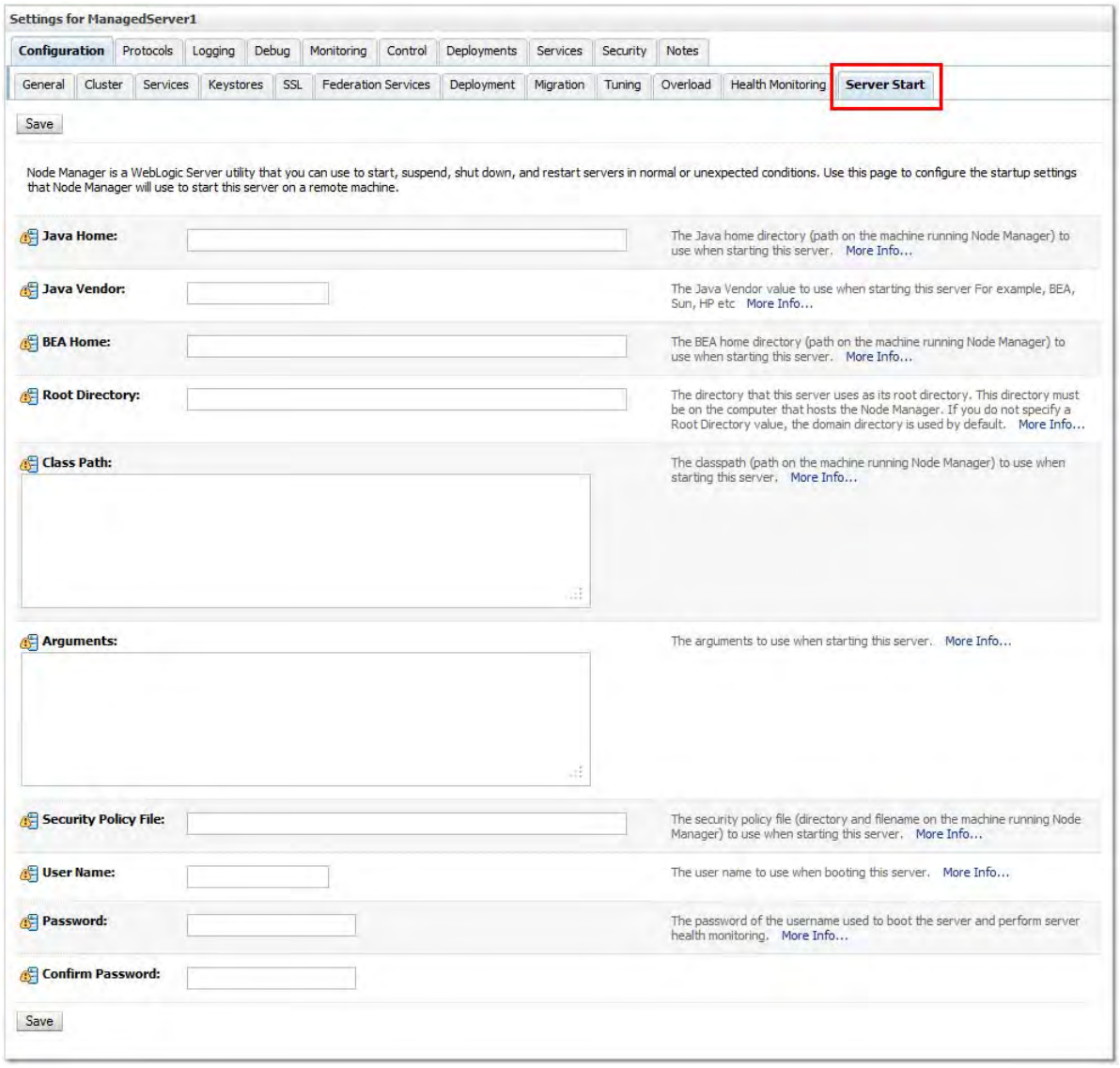
The Home Page is displayed.



3.2. Click **Servers**. The Summary of Servers page is displayed.



3.3. Click the name of the server where the Gateway will be deployed. The Settings page is displayed.



3.4. Click the **Server Start** tab.

3.5. Click **Lock & Edit** in the Change Center pane.

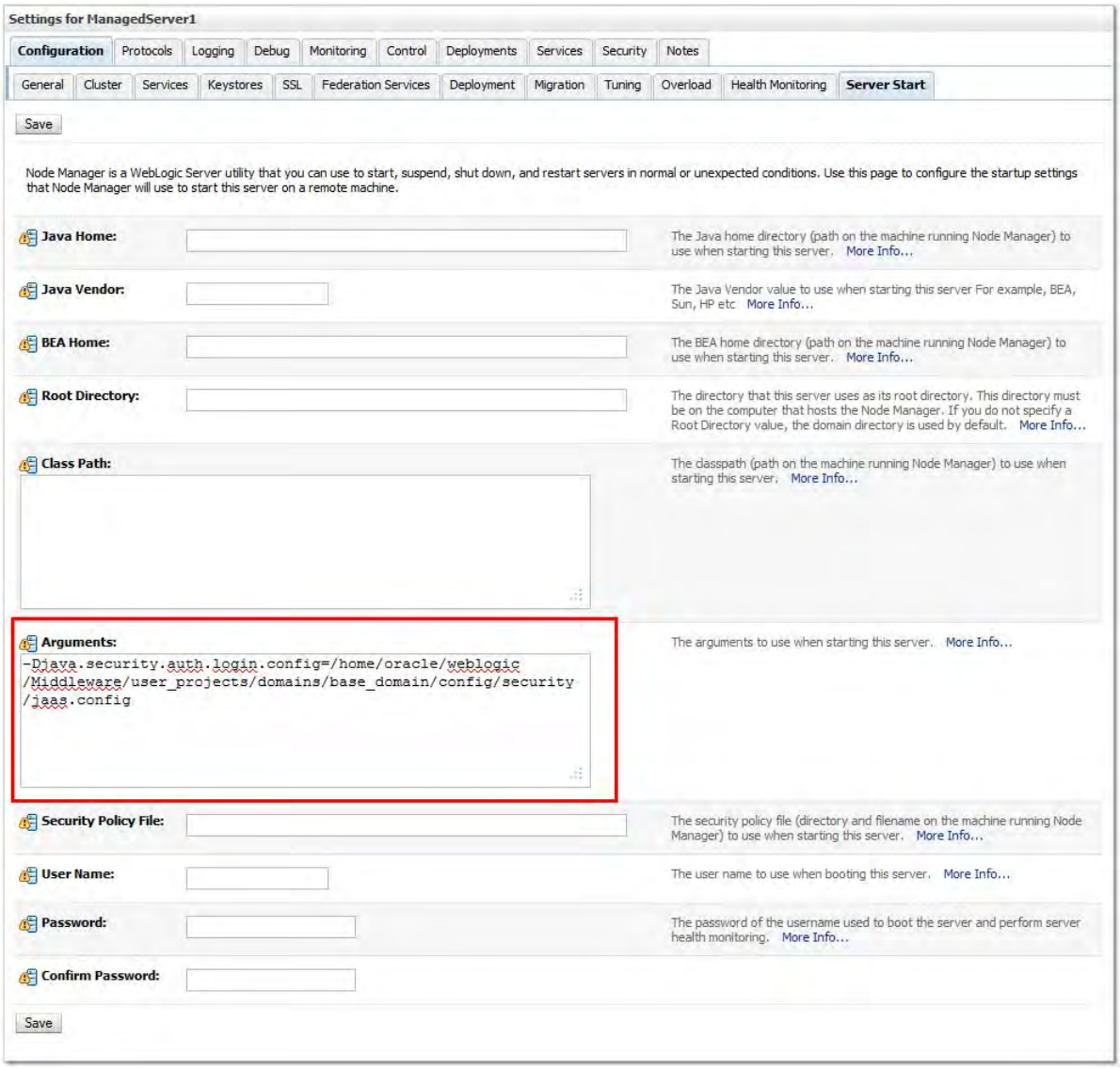
3.6. Enter the following in the **Arguments** field:

```
-Djava.security.auth.login.config=<WebLogic Home>/
user_projects/domains/<your domain dir>/config/security/
jaas.config
```

where `<WebLogic Home>` is the base directory for all Oracle WebLogic software packages and configuration files, and `<your domain dir>` is the domain where the Gateway will be deployed.

Note

The argument should reflect the full path to the `jaas.config` file and the name of the file itself. ■



3.7. Click **Save.**

3.8. Click **Activate Changes** in the Change Center pane.

Option 2 - If you are using a script to start the Managed Server

Use this option if the Managed Server will be started by running the `startManagedWebLogic.sh` (or `.cmd`) script. A `JAVA_OPTIONS` statement must be added to the `setDomainEnv.sh` (or `.cmd`) script. The location of the JAAS configuration file applies to the entire domain, including the Admin Server and all Managed Servers.

Use the following steps to update the script for Windows.

3.1. Open the `setDomainEnv.cmd` file located under `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

3.2. Search for the last occurrence of the following text:

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
```

3.3. Add the following in the line preceding the line identified in step 3.2.

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Djava.security.auth.login.config=
<domain home>\config\security\jaas.config
```

Use the following steps to update the script for Linux/Unix.

3.1. Open the `setDomainEnv.sh` file located under `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

3.2. Search for the last occurrence of the following text:

```
JAVA_OPTIONS="${JAVA_OPTIONS}"
```

3.3. Add the following in the line preceding the line identified in step 3.2.

```
JAVA_OPTIONS="${JAVA_OPTIONS}
-Djava.security.auth.login.config=
<domain home>/config/security/jaas.config"
```

Note

There is a space between the closing brace and the dash (that is, {JAVA_OPTIONS}**[space]**-Djava). ■

4. Restart the appropriate server(s).

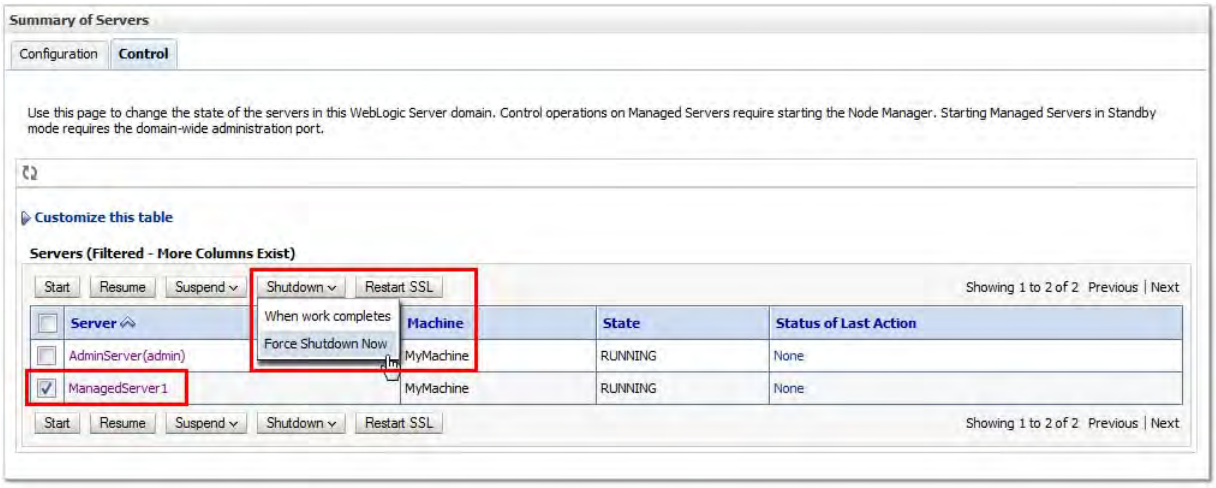
There are two ways to restart the server(s). Use option 1 (page 7-37) if a single Managed Server was configured. Use option 2 (page 7-38) if all servers in the domain were configured.

Option 1 - If a single Managed Server was configured

Use this option if a single Managed Server was configured. Only that server needs to be restarted.

4.1. Navigate to the Summary of Servers page.

4.2. Click the **Control** tab.

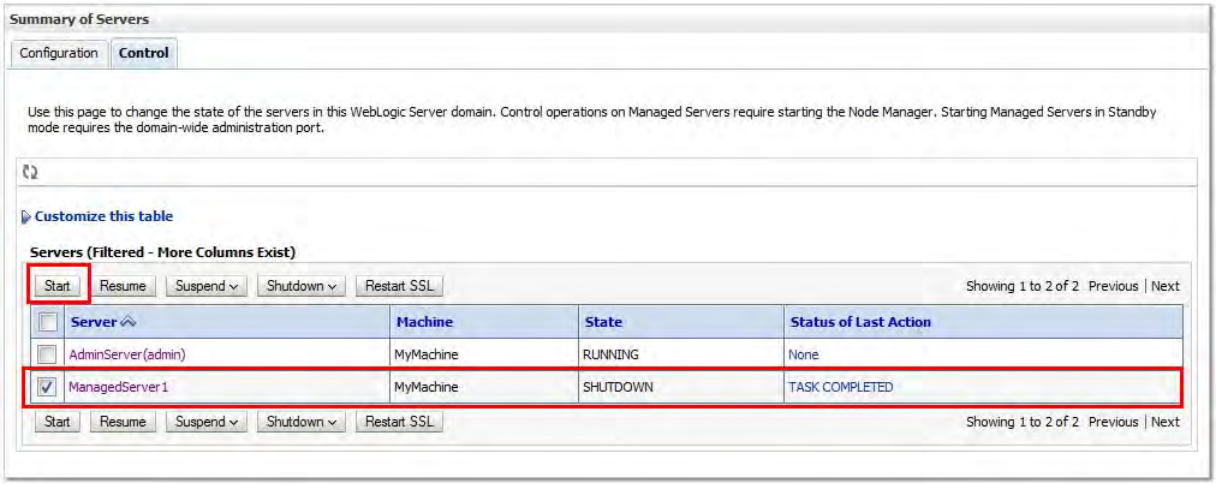


4.3. Select the Managed Server where the configuration changes were made.

4.4. Click **Shutdown** -> **Force Shutdown Now**.

4.5. Confirm the selection.

4.6. Wait for the server to enter a *SHUTDOWN* state.

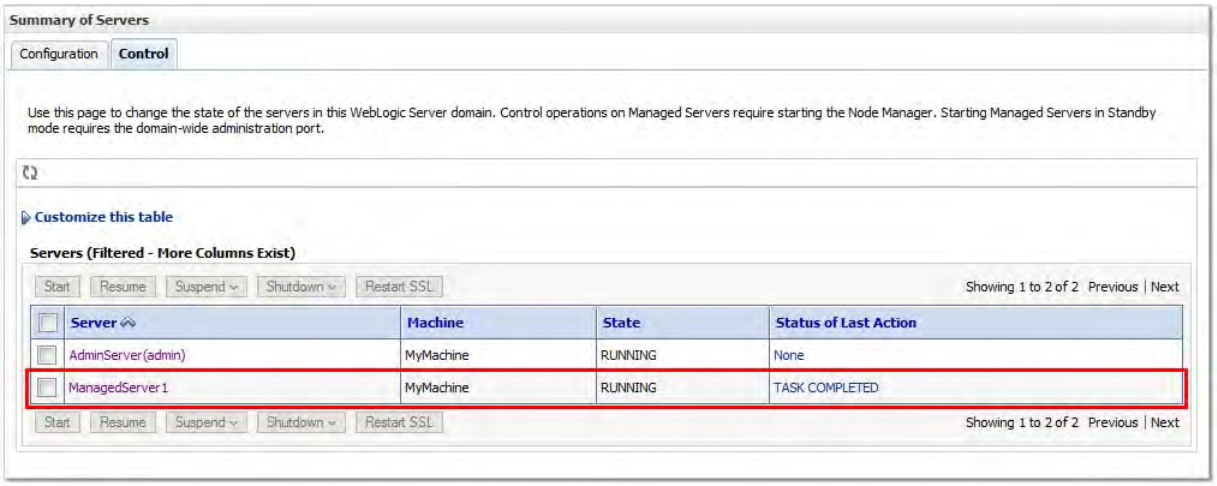


4.7. Select the same Managed Server.

4.8. Click **Start**.

4.9. Confirm the selection.

4.10. Wait for the server to enter a *RUNNING* state.



Option 2 - If all servers in the domain were configured

Use this option if all servers in the domain were configured. Only the Managed Server needs to be restarted.

Use the following steps to restart the server for Windows.

4.1. Navigate to `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

4.2. Stop the server by running the following script:

```
stopManagedWebLogic.cmd <ServerName>
```

Note

There is a space between the command and <ServerName>; that is, stopManagedWeblogic.cmd**[space]**<ServerName>. ■

Example:

```
stopManagedWebLogic.cmd ManagedServer1
```

4.3. Start the server by running the following script:

```
startManagedWebLogic.cmd <ServerName>
```

Use the following steps to restart the server for Linux/Unix.

4.1. Navigate to `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

4.2. Stop the server by running the following script:

```
./stopManagedWebLogic.sh <ServerName>
```

Note

There is a space between the command and <ServerName>; that is, ./stopManagedWeblogic.sh**[space]**<ServerName>.

Example:

```
./stopManagedWebLogic.sh ManagedServer1
```

4.3. Start the server by running the following script:

```
./startManagedWebLogic.sh <ServerName>
```

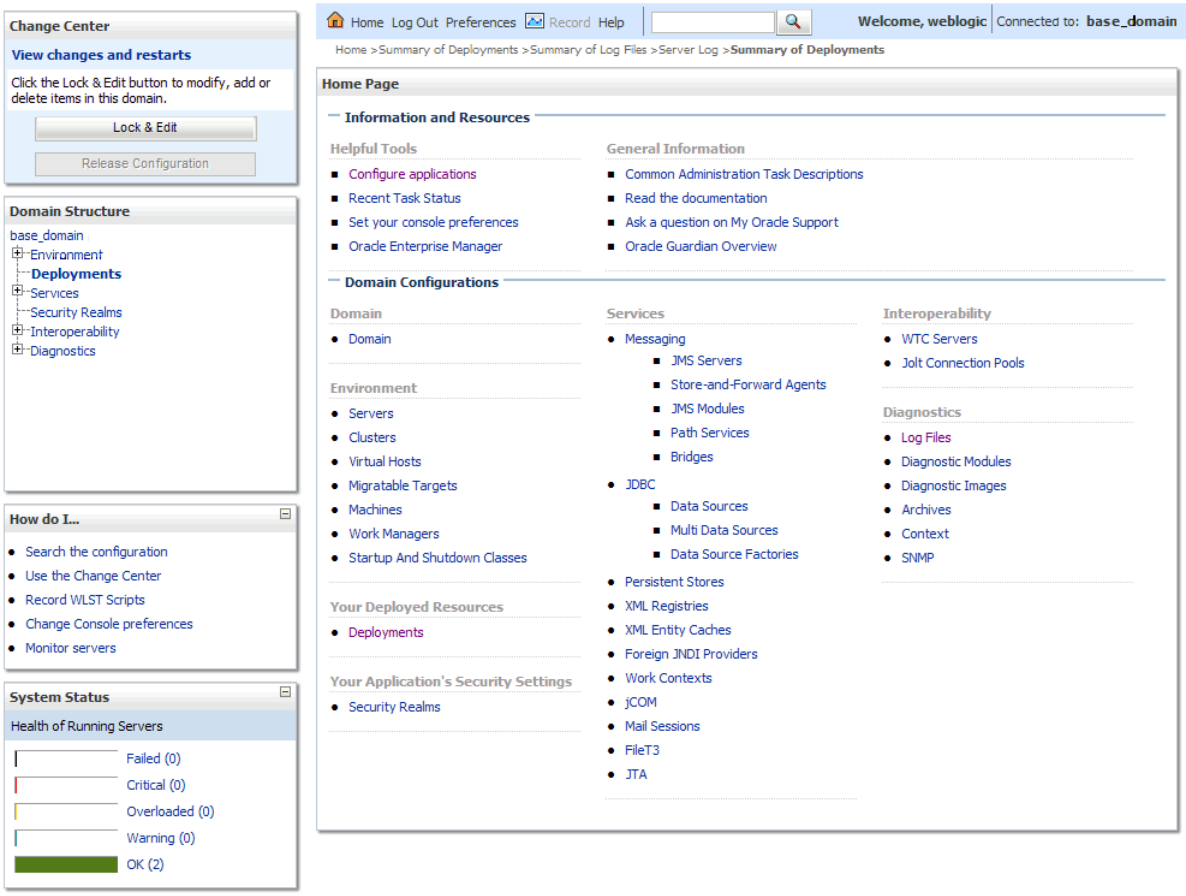
Step 4 Define the data source for Oracle Advanced Queuing

A data source provides the connection properties to the Banner database. Use the following steps to define the data source that the Banner Identity Gateway uses to connect to Oracle Advanced Queueing to consume Banner identity messages.

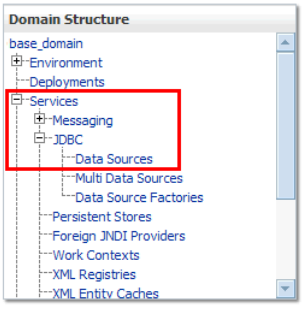
1. Connect to the Oracle WebLogic Server Administration Console:

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

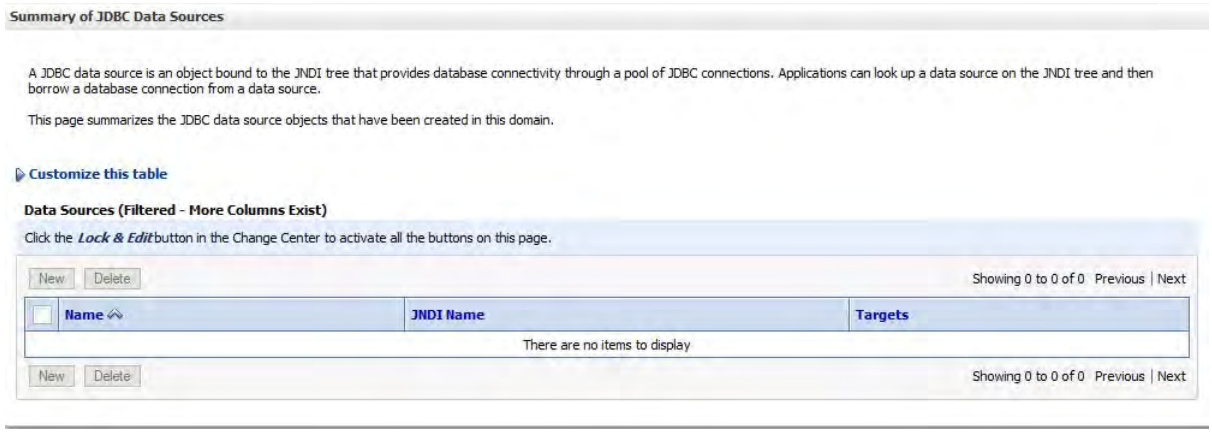
The Home Page is displayed.



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

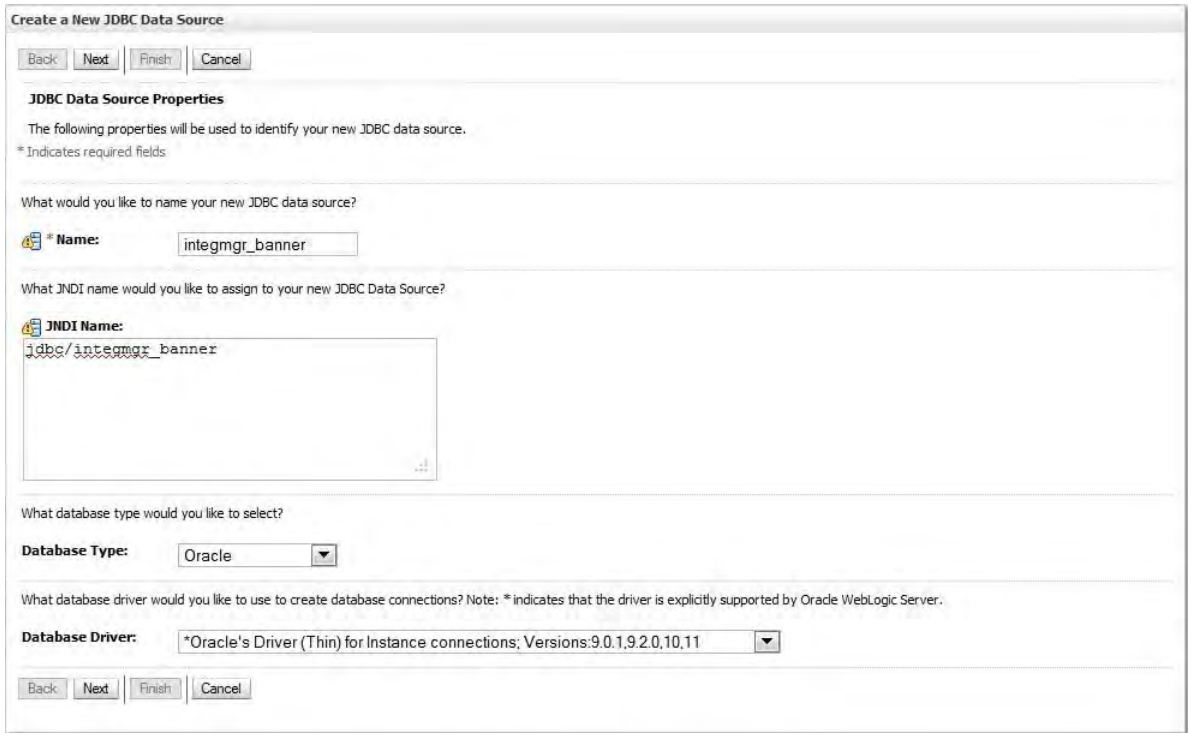


The Summary of JDBC Data Sources page is displayed.



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

4. On the Summary of JDBC Data Sources page, click **New**. The Create a New JDBC Data Source page is displayed.

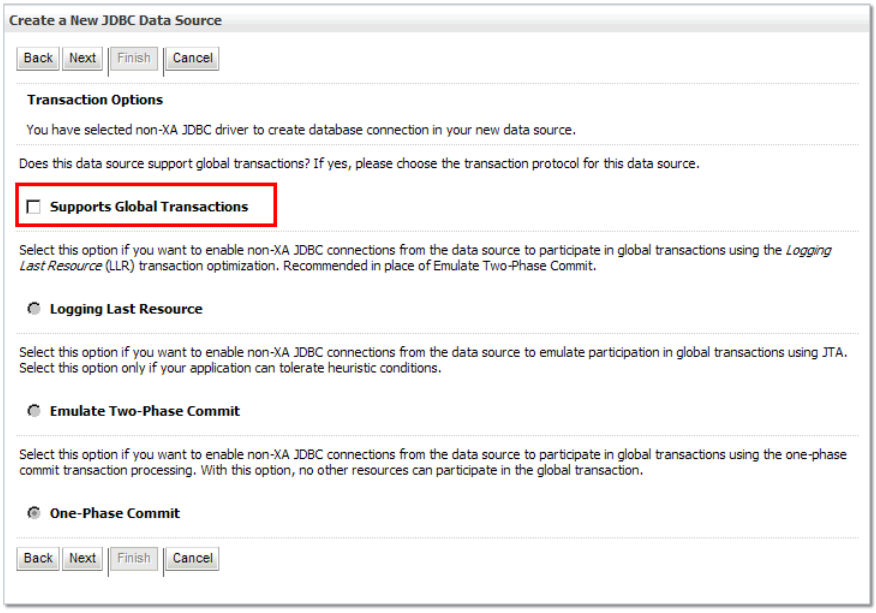


5. Enter the following data source properties:

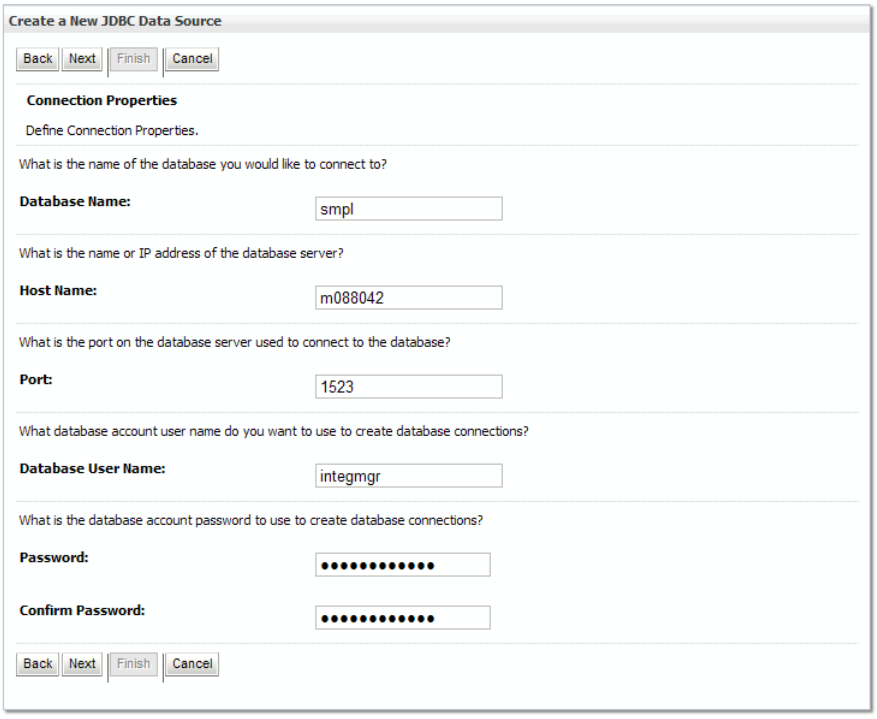
Name	*integmgr_banner*
JNDI Name	*jdbc/integmgr_banner*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

6. 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 7. If the following page is not displayed skip to step 8.



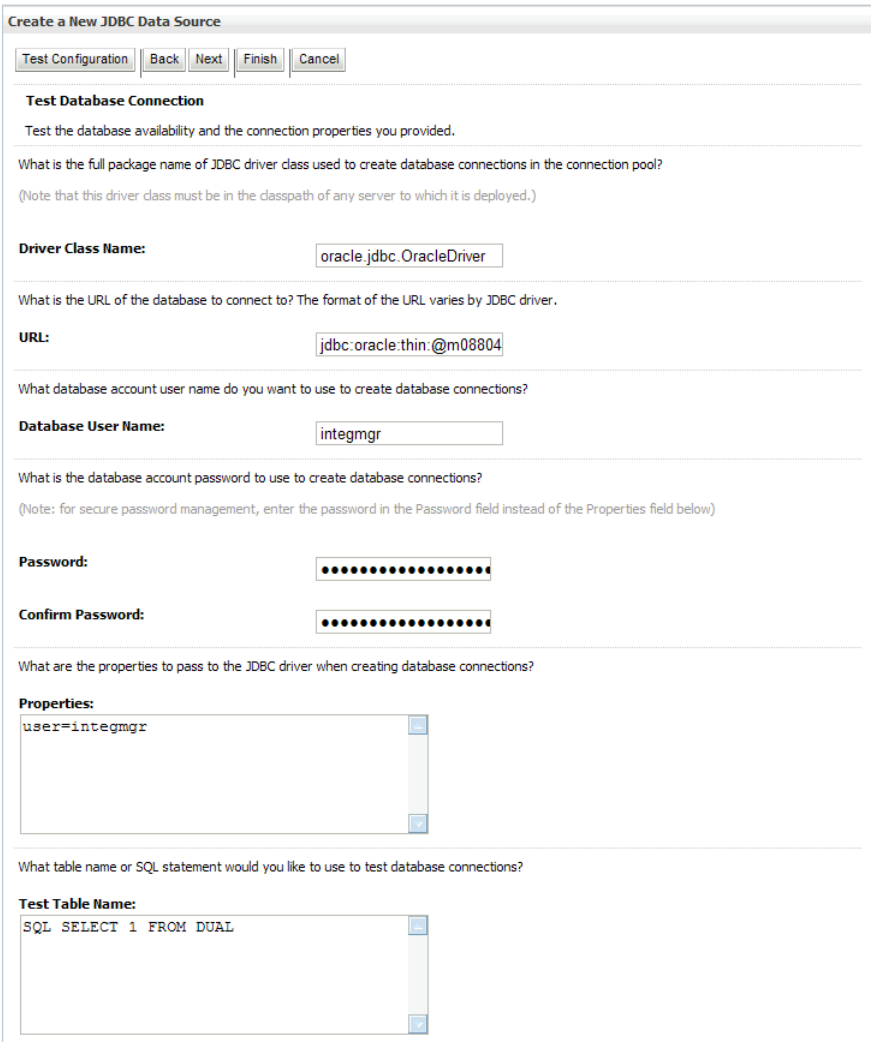
7. Click **Next**. The next page is displayed.



8. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*integmgr*
Password	Password for the `integmgr` user
Confirm Password	Confirmation of the password

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



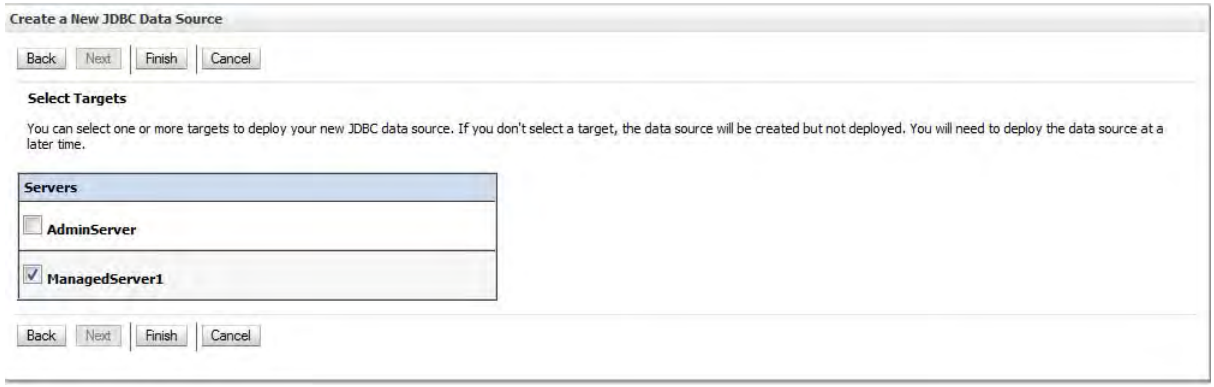
10. Verify the property values.

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.

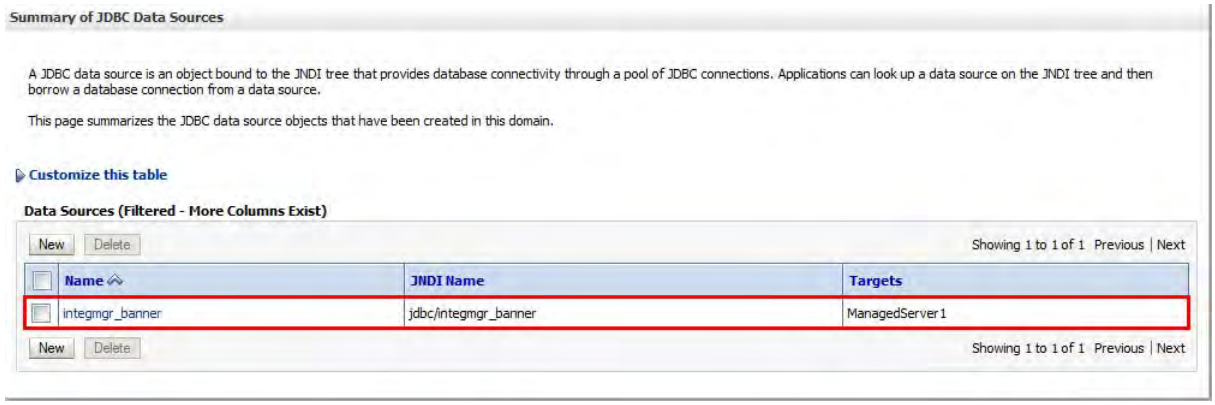
    11.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

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



13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the Gateway will be deployed.

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



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

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

Step 5 Define the data source for the Banner Identity Gateway

Use the following steps to define the data source that is used to access the database schema that is created for the Banner Identity Gateway.

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	*bnixmgr*
JNDI Name	*jdbc/bnixmgr*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

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 or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*bnixmgr*
Password	Password for the `bnixmgr` 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 **Next**. The next page is displayed.

13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the Gateway will be deployed.

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

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

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

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 viewing and administering the IAM Oracle Streams environment in Banner.

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	*streamsadmin*
JNDI Name	*jdbc/streamsadmin*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

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 or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*streamsadmin*
Password	Password for the `streamsadmin` 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 **Next**. The next page is displayed.

13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the Gateway will be deployed.

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

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

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

Step 7 Define the data source for the Banner security administrator

Use the following steps to define the data source for the Banner security administrator.

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	*inbadmin*
JNDI Name	*jdbc/inbadmin*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

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 or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*bansecr*

Password	Password for the `bansecr` user
Confirm Password	Confirmation of the password

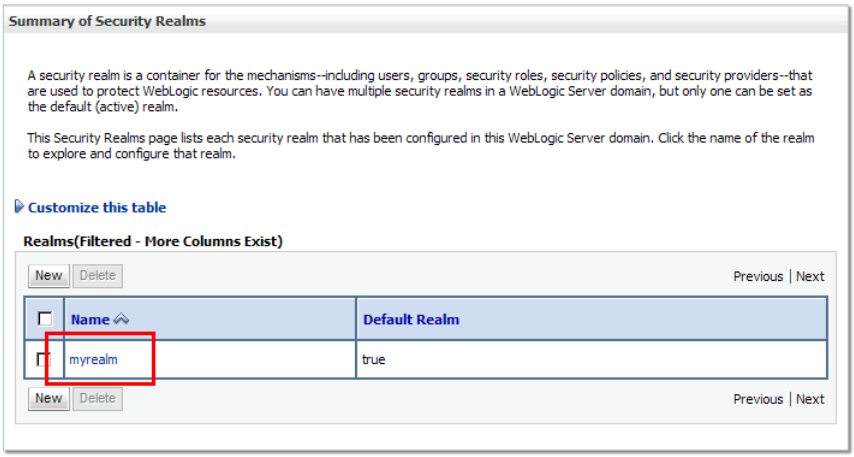
9. Click **Next**. The next page is displayed with the properties that you entered.
10. Verify the property values.
11. Click **Test Configuration**. The page is redisplayed with a success or failure message.
    11.1. If the test succeeds, continue with the next step.
    11.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
12. Click **Next**. The next page is displayed.
13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the Gateway will be deployed.
14. Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.
15. Verify that the new data source is associated with the server.
16. In the Change Center pane, click **Activate Changes.**

Step 8 Configure the security group and user

Use the following steps to add the `bnigAdminGroup` group and an administrative user to the Banner Identity Gateway application. This group and user are required for accessing the Banner Identity Gateway administrative interface and for invoking the Banner PSP Web services.

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

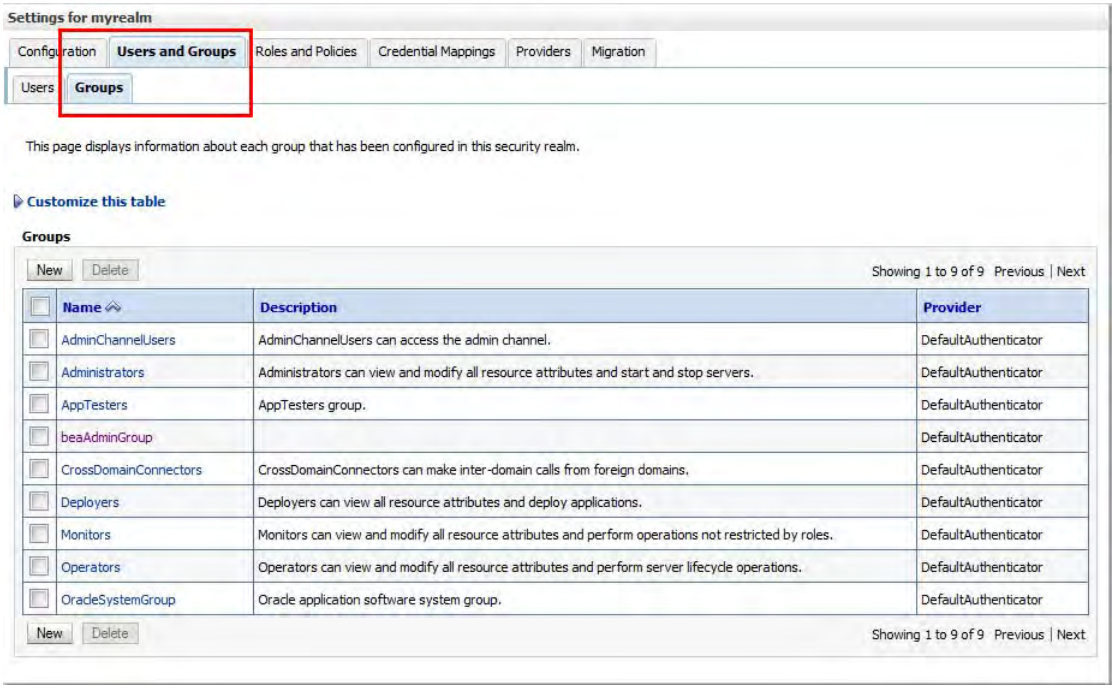
The Summary of Security Realms page is displayed.



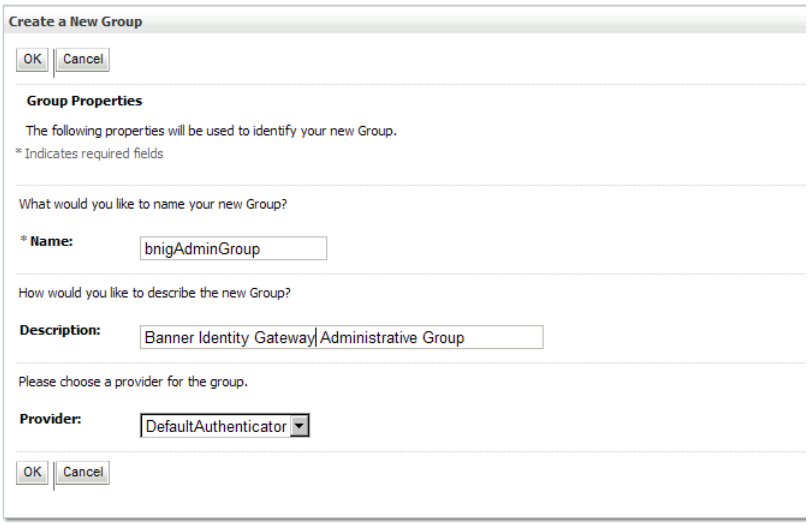
2. Click **myrealm**. The Settings page is displayed.

3. Select the **Users and Groups** tab.

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



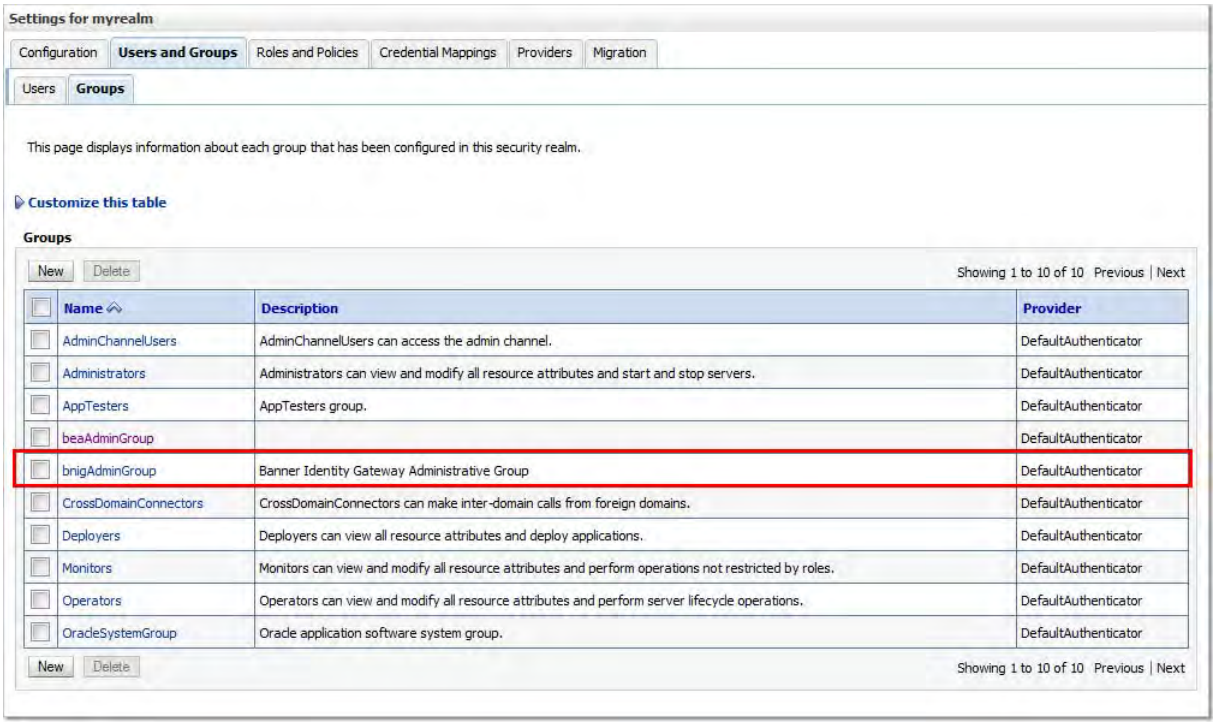
5. Click **New**. The Create a New Group page is displayed.



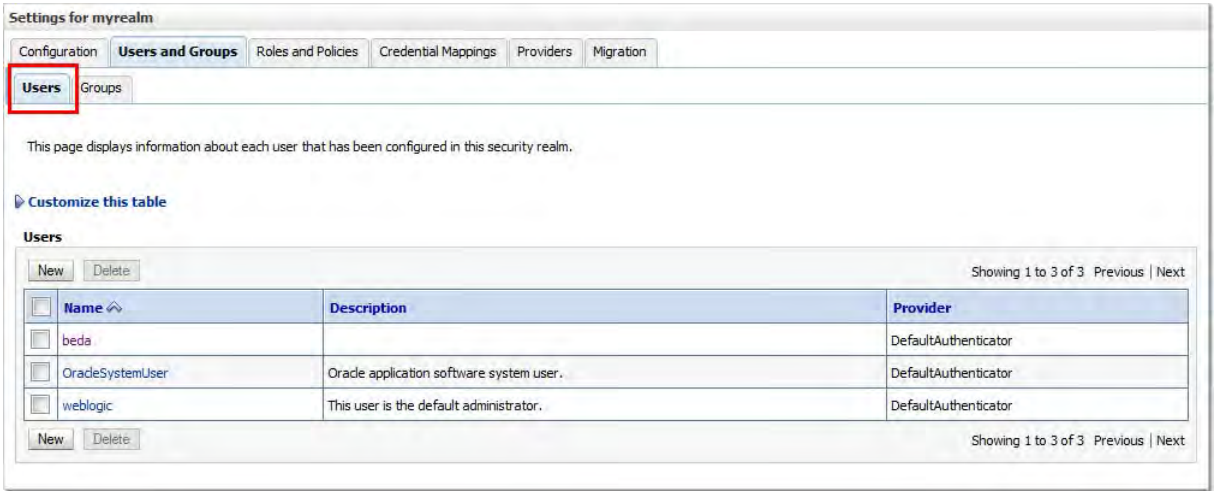
6. Enter the following information to create a group:

Name	*bnigAdminGroup*
Description	*Banner Identity Gateway Administrative Group*
Provider	*DefaultAuthenticator*

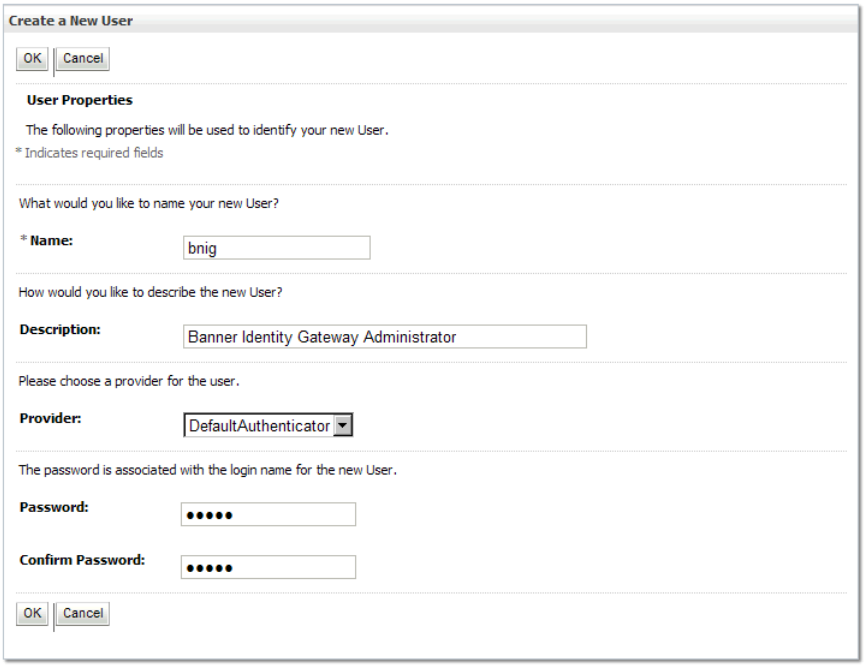
7. Click **OK**. The table of groups is redisplayed with the new group.



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



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

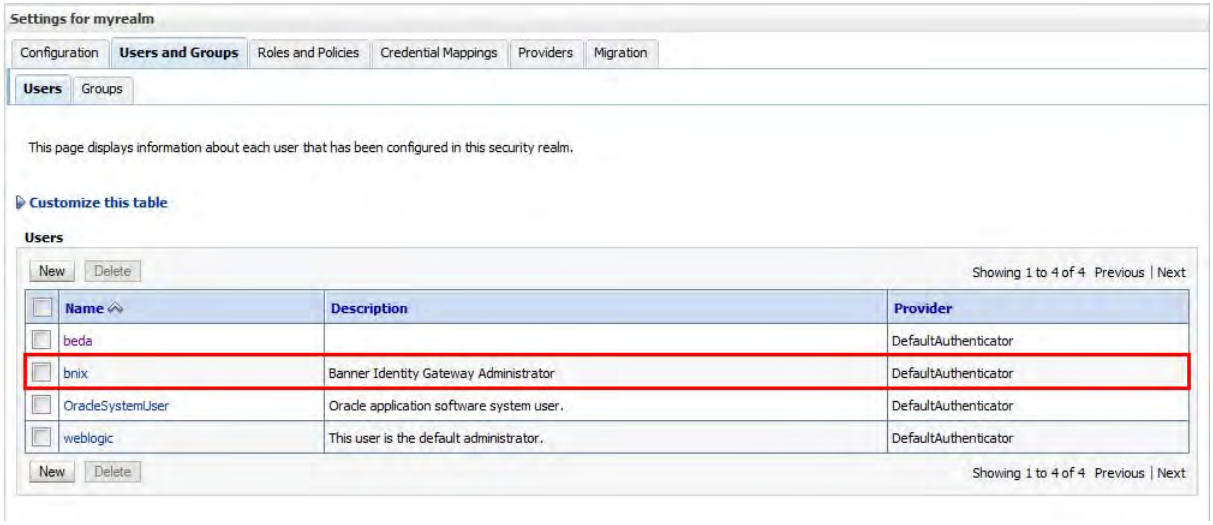


10. Enter the following information to create a user:

Name	*bnig* (This is an example. Enter the name of your choice.*)*
Description	*Banner Identity Gateway Administrator*

Provider	*DefaultAuthenticator*
Password	Password used to log in to the Banner Identity Gateway administrative interface
Confirm Password	Confirmation of the password

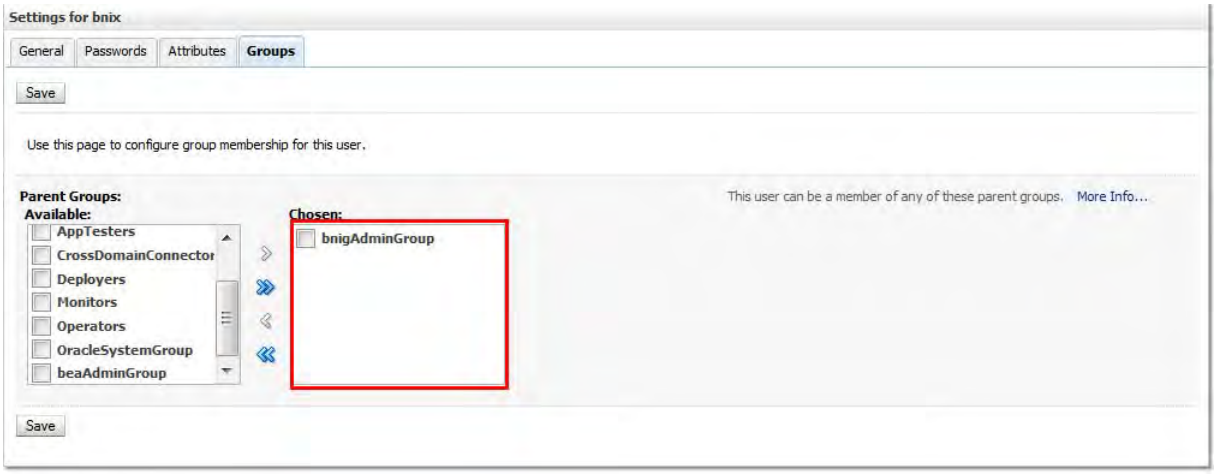
11. Click **OK**. The table of users is redisplayed with the new user.



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 *bnigAdminGroup* in the **Available** list and move it to the **Chosen** list.



15. Click **Save**.

Step 9 Create a JMS server

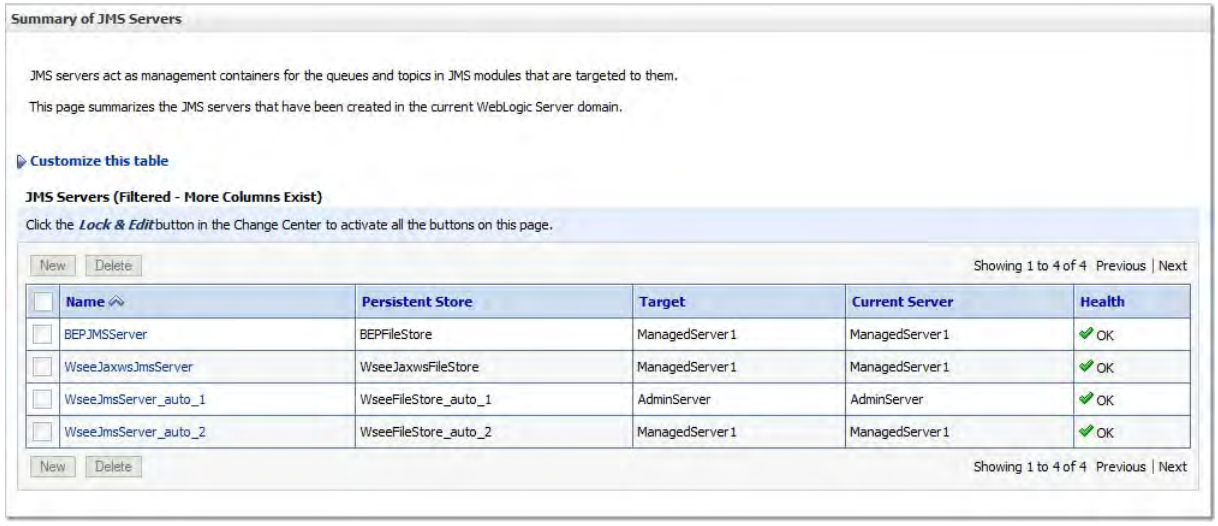
The BEIS application uses Java messaging to move UDCIdentity messages among its various services. A JMS server acts as a management container for JMS resources such as connection factories, queues, and topics. Use the following steps to create a JMS server for BEIS.

Note

All BEIS components share the JMS server. If the server was previously created, you can skip this step. ■

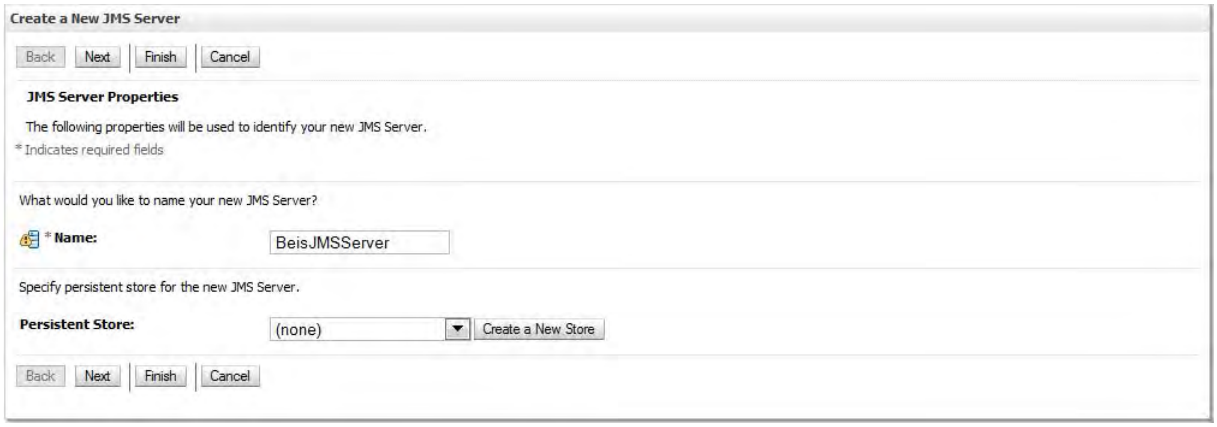
1. In the Domain Structure pane, expand and click **Services -> Messaging -> JMS Servers**.

The Summary of JMS Servers page is displayed.



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

3. On the Summary of JMS Servers page, click **New**. The Create a New JMS Server page is displayed.

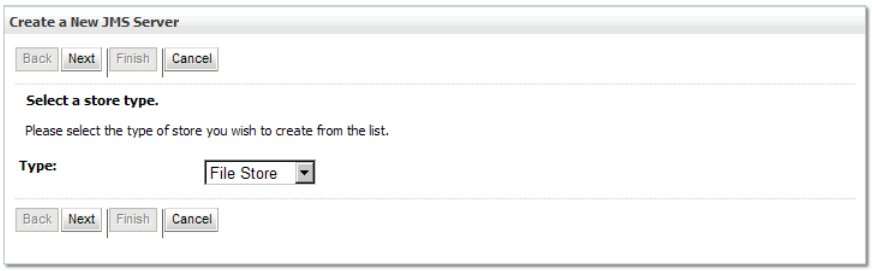


4. Enter the name of the new JMS server in the **Name** field.

5. If the persistent store is listed in the **Persistent Store** drop-down list, select the persistent store and go to step 6.

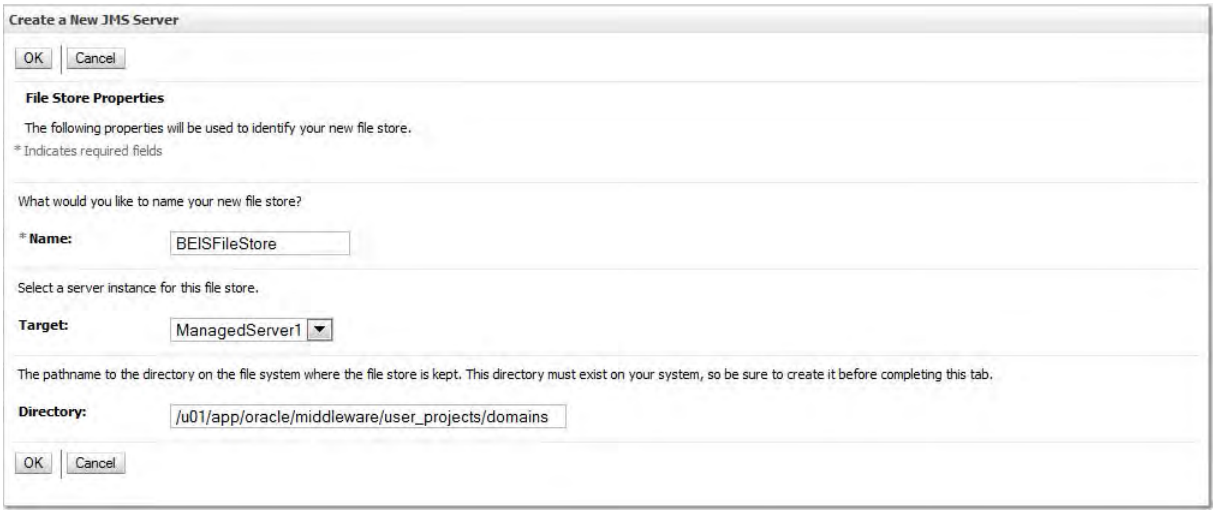
   If the persistent store is not listed in the **Persistent Store** drop-down list, use the following steps to create the persistent store:

   5.1. Click **Create a New Store**. The Create a New JMS Server page is displayed.



   5.2. Select *File Store* from the **Type** drop-down list.

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



   5.4. Enter the following file store properties:

Name	Name of the file store
Target	Server where the file store is to be created
Directory	Directory path where the file store is to be created. It is recommended that you create a directory under the domain where you are deploying the BEIS components (for example, `/u01/app/oracle/middleware/user_projects/domains/BEIS_8_1_5/beisfileStore`).

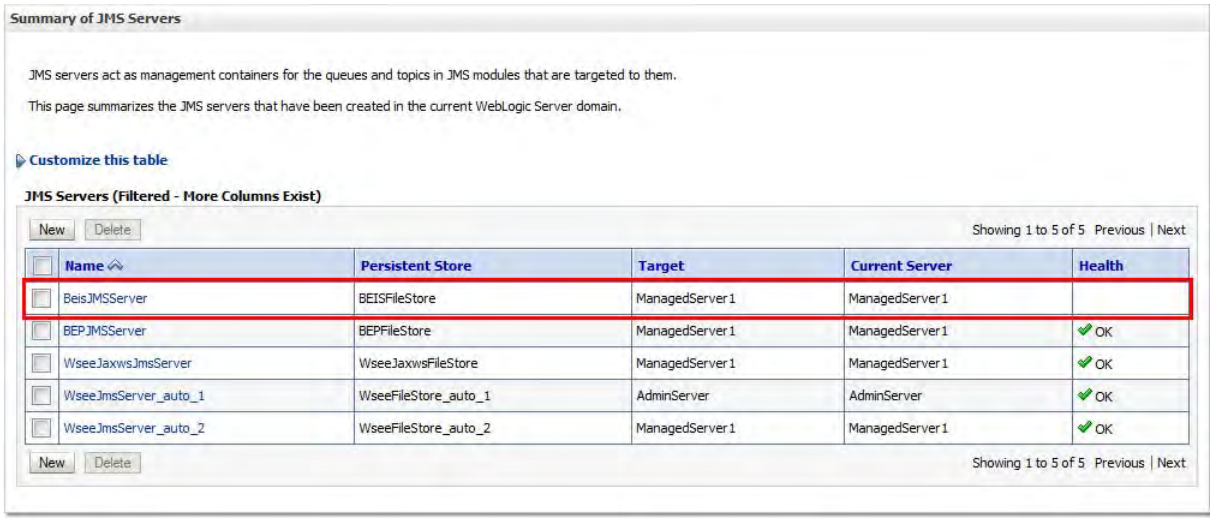
5.5. Click **OK**. The first page of the Create a New JMS Server page is redisplayed.

5.6. Select the new file store from the **Persistent Store** drop-down list and go to step 6.

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

7. Select the target server from the **Target** drop-down list. This should be the Managed Server where the Gateway will be deployed.

8. Click **Finish**. The Summary of JMS Servers page is redisplayed with the new server.



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

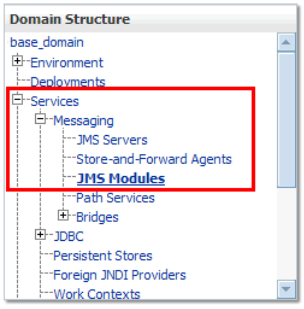
Step 10 Create a JMS module

A JMS module is used to store JMS resources such as connection factories, queues, and topics. Use the following steps to create a JMS module for BEIS.

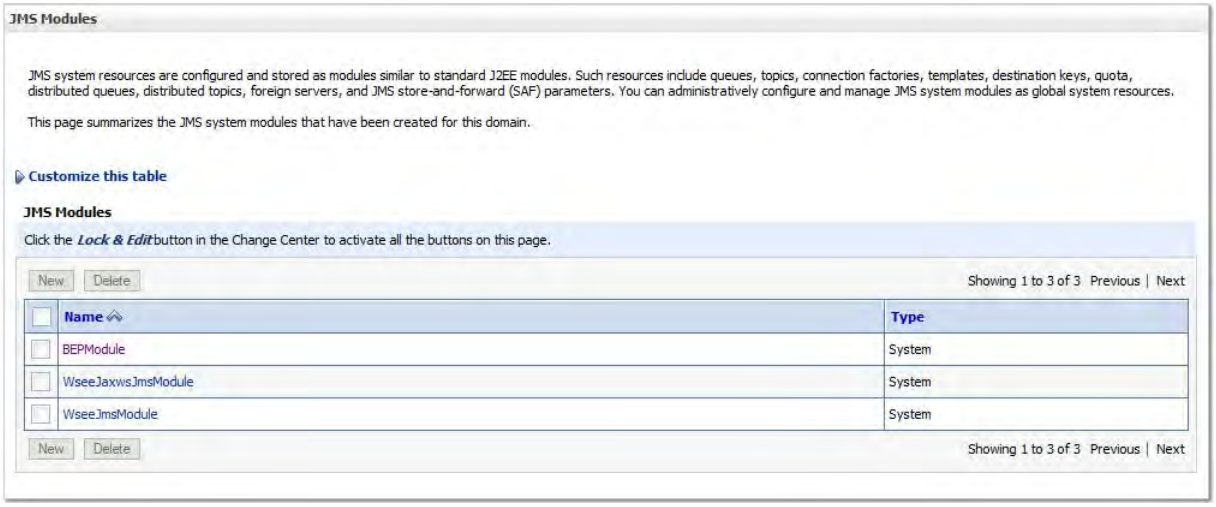
Note

All components of BEIS share the JMS module. If the module was previously created, you can skip this step. ■

1. In the Domain Structure pane, expand and click **Services -> Messaging -> JMS Modules**.

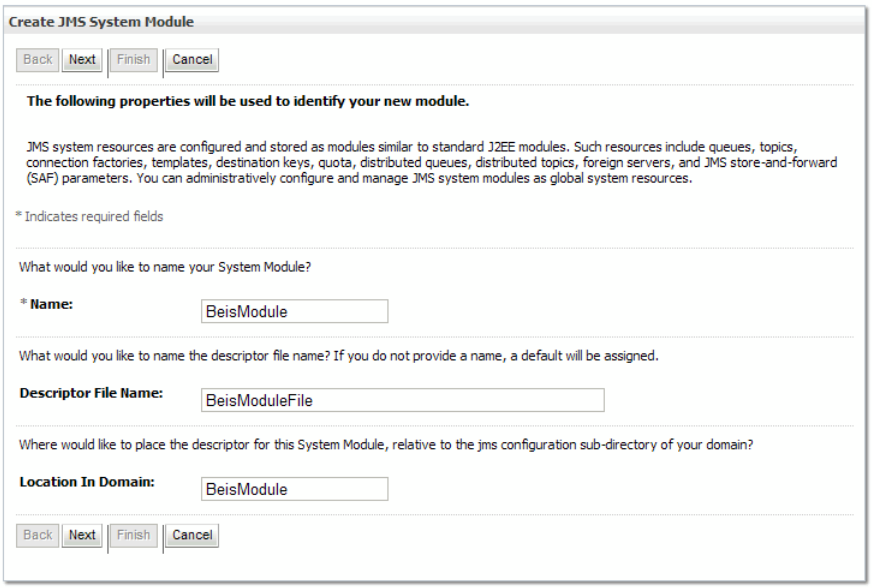


The JMS Modules page is displayed.



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

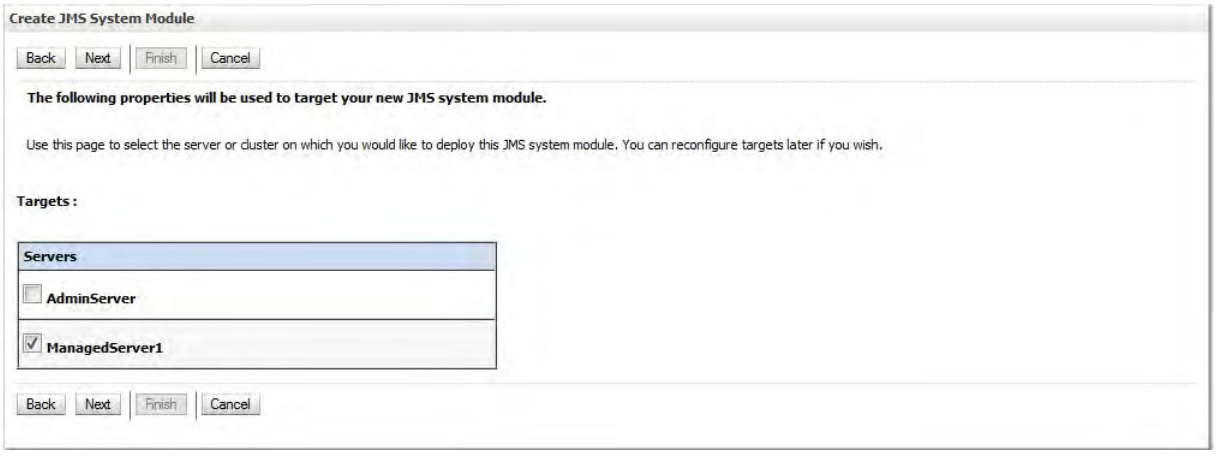
3. Click **New**. The Create JMS System Module page is displayed.



4. Enter the following information to create a JMS module:

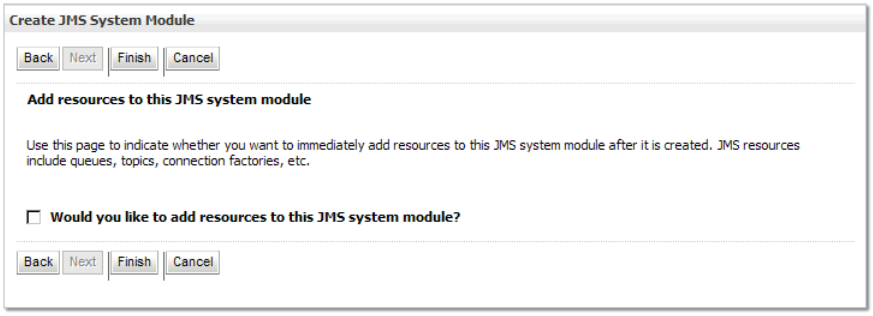
Name	Name of the new JMS module
Descriptor File Name	Name of the descriptor file
Location in Domain	Location where the descriptor file should be placed

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

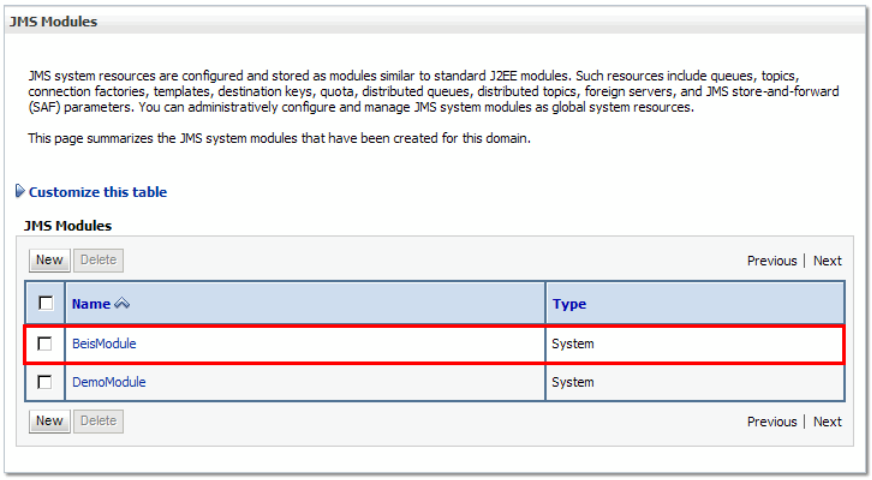


6. Select the server where the JMS module should be deployed.

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



8. Click **Finish**. The JMS Modules page is redisplayed with the new module.



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

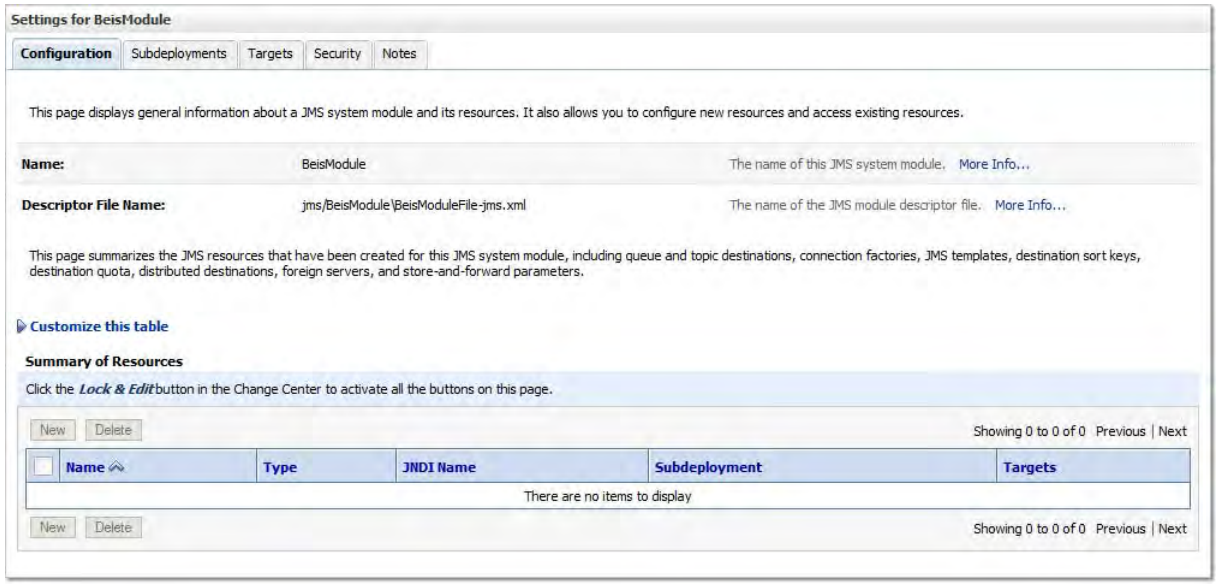
Step 11 Configure a JMS topic and connection factory

Use the following steps to add a JMS topic and connection factory for the Banner Identity Gateway.

1. In the Domain Structure pane, expand and click **Services -> Messaging -> JMS Modules**.

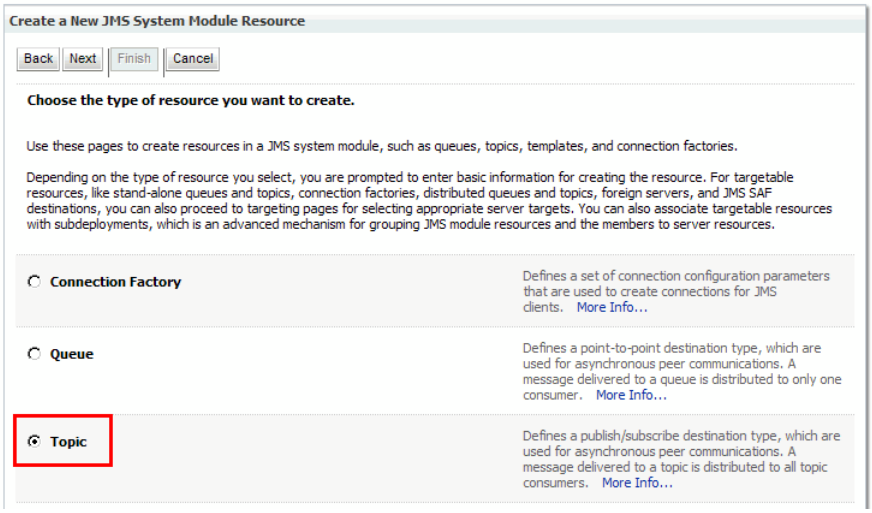
The JMS Modules page is displayed.

2. Click the name of the module that was created for BEIS. The Settings page is displayed.



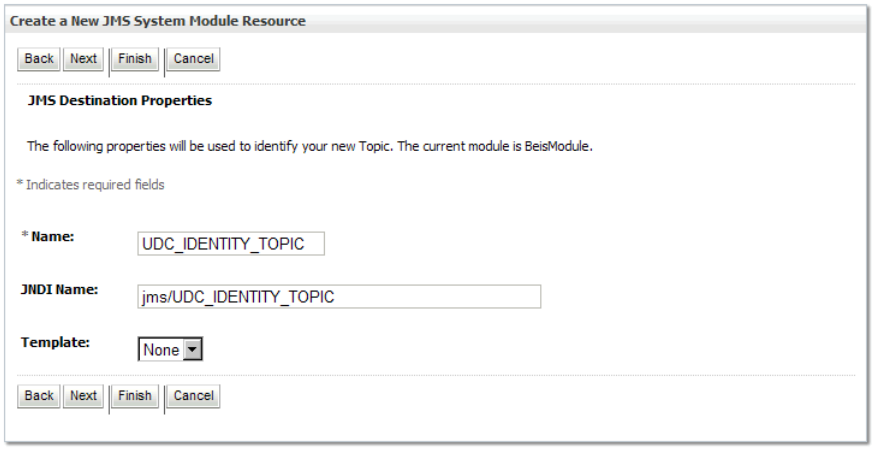
3. In the Change Center pane, click **Lock & Edit**.

4. On the Settings page, click **New**. The Create a New JMS System Module Resource page is displayed.



5. Select **Topic**.

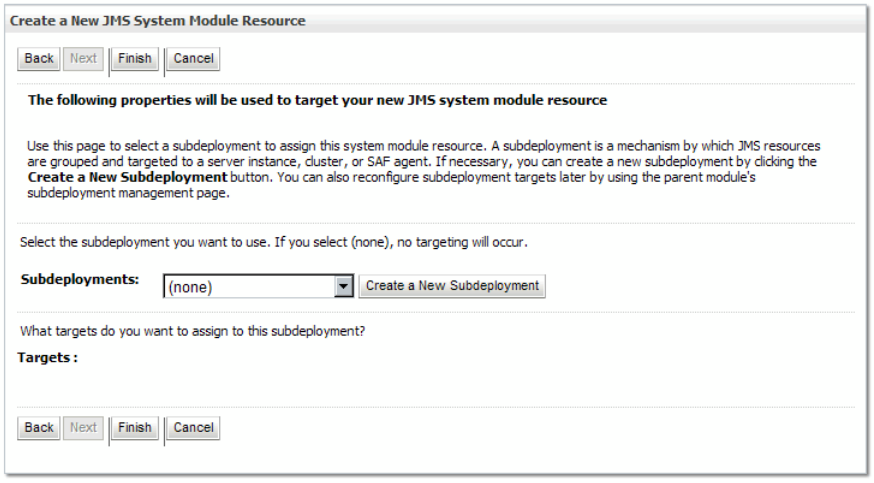
6. Click **Next**. The next page is displayed.



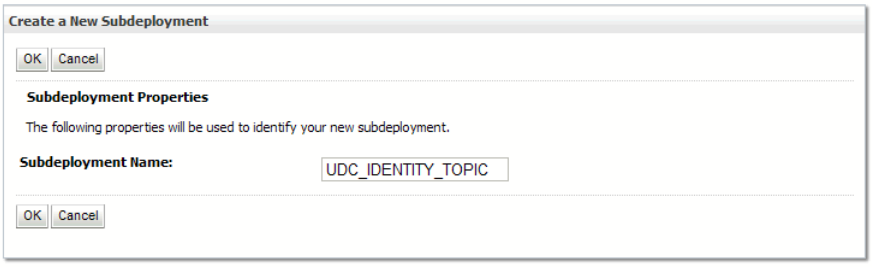
7. Enter the following information to create a topic:

Name	*UDC_IDENTITY_TOPIC*
JNDI Name	*jms/UDC_IDENTITY_TOPIC*

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

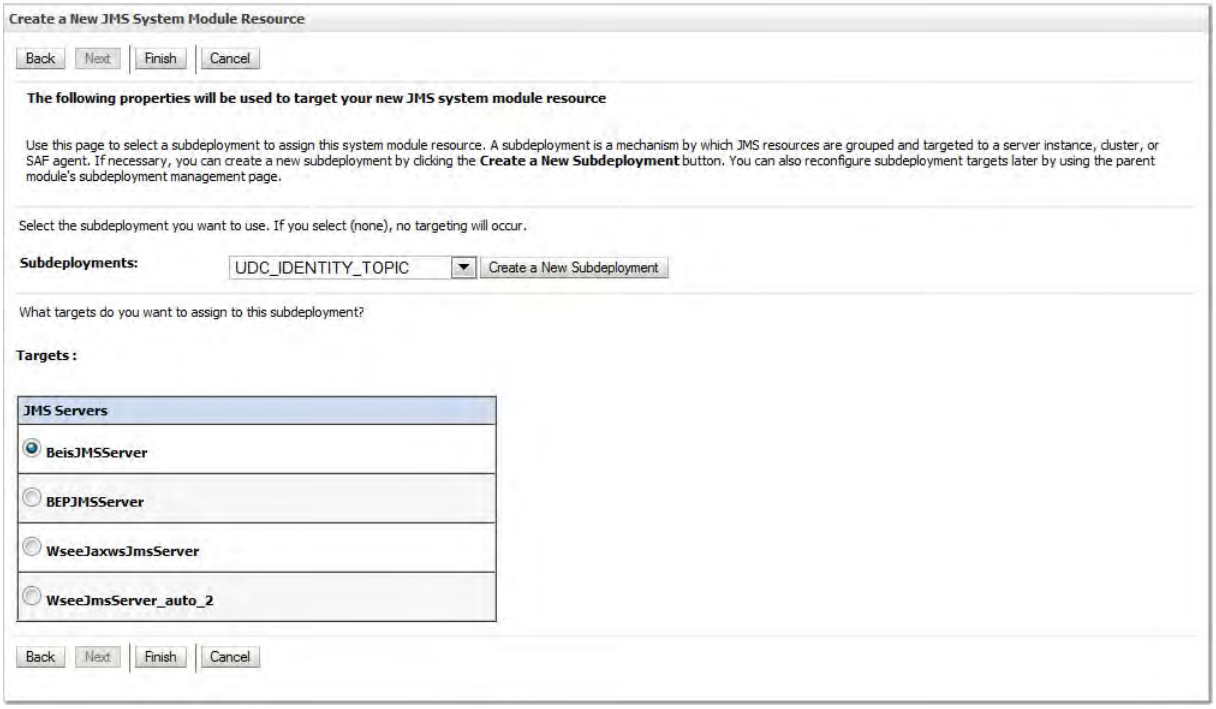


9. Click **Create a New Subdeployment**. The Create a New Subdeployment page is displayed.



10. Enter the name of the subdeployment (for example, *UDC_IDENTITY_TOPIC*) in the **Subdeployment Name** field.

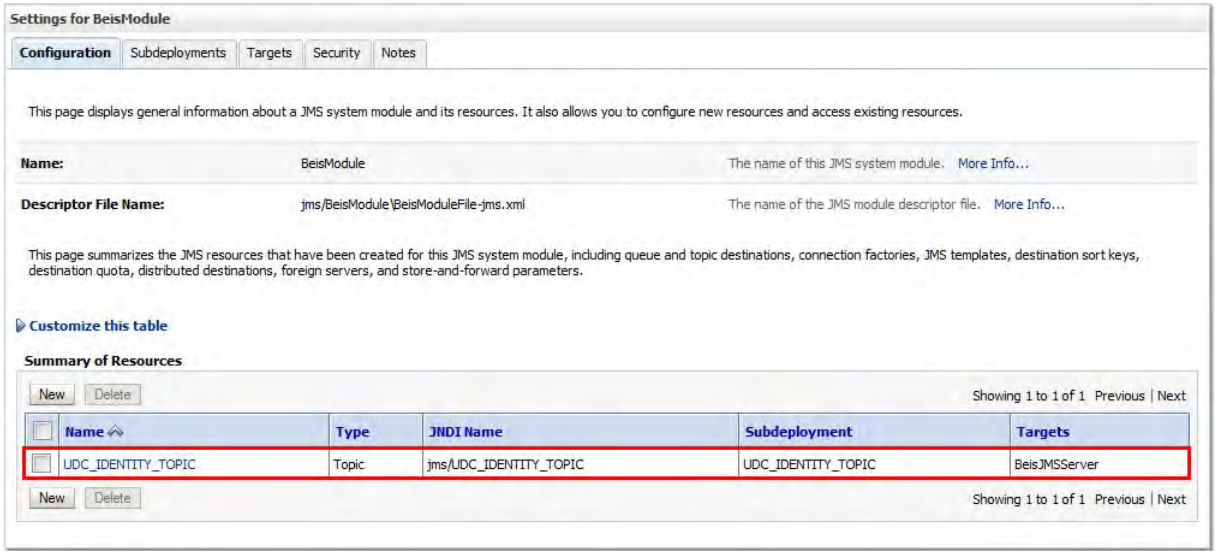
11. Click **OK**. The Create a New JMS System Module Resource page is redisplayed.



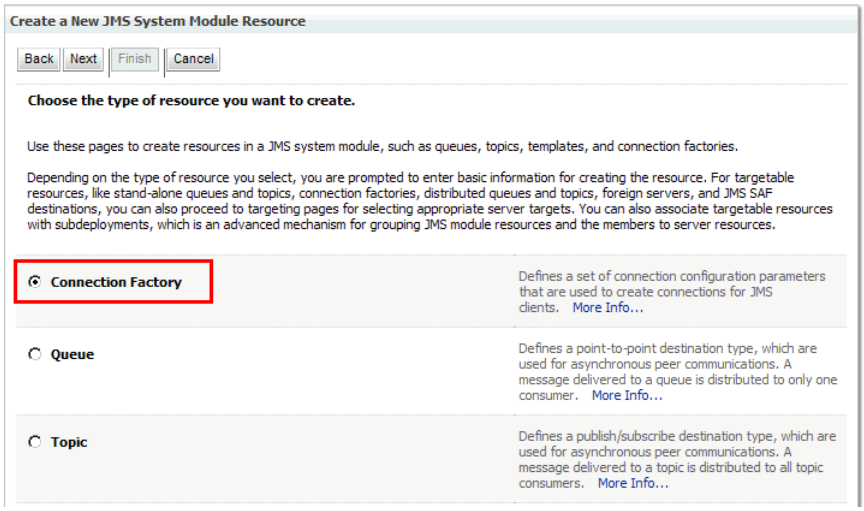
12. Select *UDC_IDENTITY_TOPIC* from the **Subdeployments** drop-down list.

13. Select a target server for the topic.

14. Click **Finish**. The Settings page is redisplayed with the new topic.

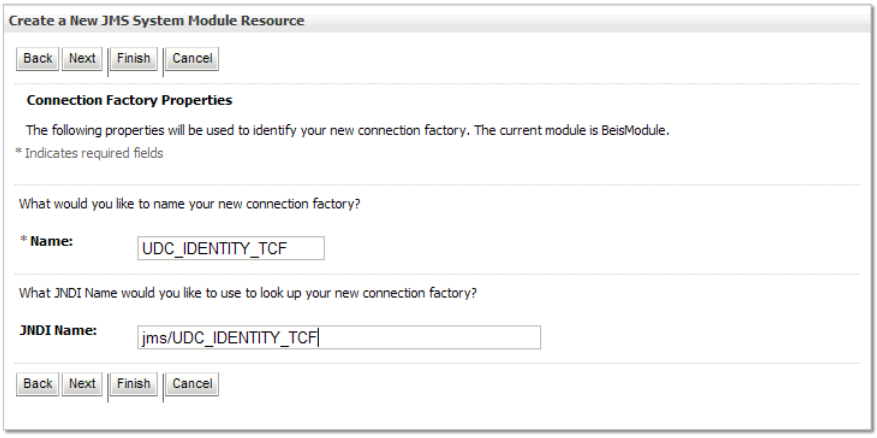


15. Click **New**. The Create a New JMS System Module Resource page is displayed.



16. Select **Connection Factory**.

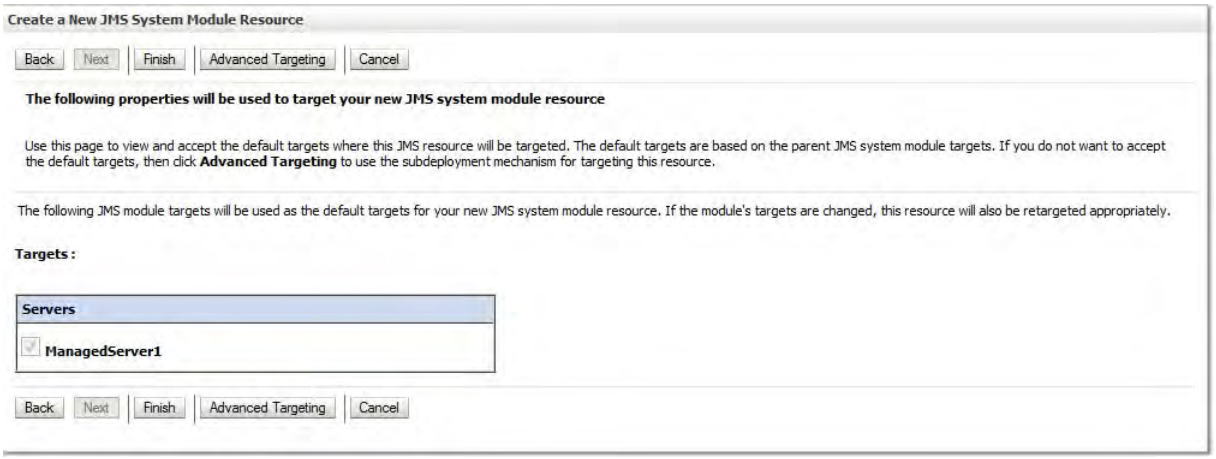
17. Click **Next**. The next page is displayed.



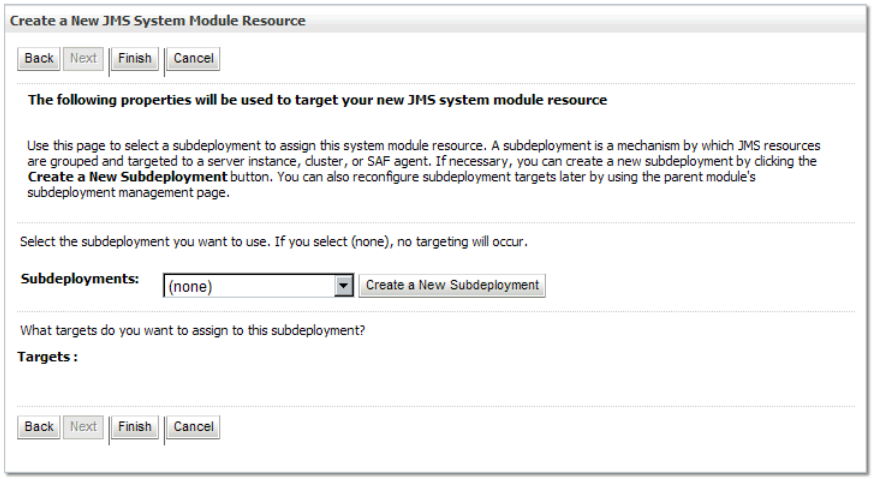
18. Enter the following information to create a connection factory:

Name	*UDC_IDENTITY_TCF*
JNDI Name	*jms/UDC_IDENTITY_TCF*

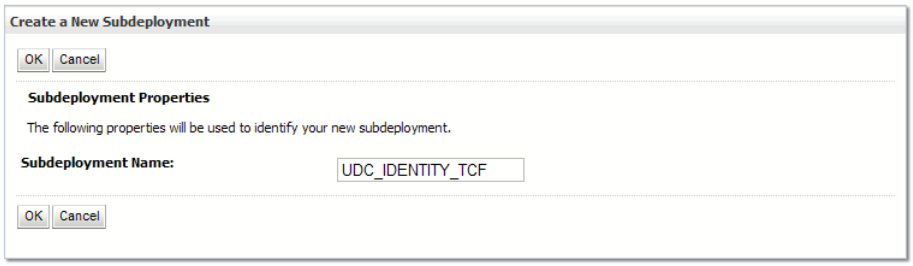
19. Click **Next**. The next page is displayed.



20. Click **Advanced Targeting**. The next page is displayed.

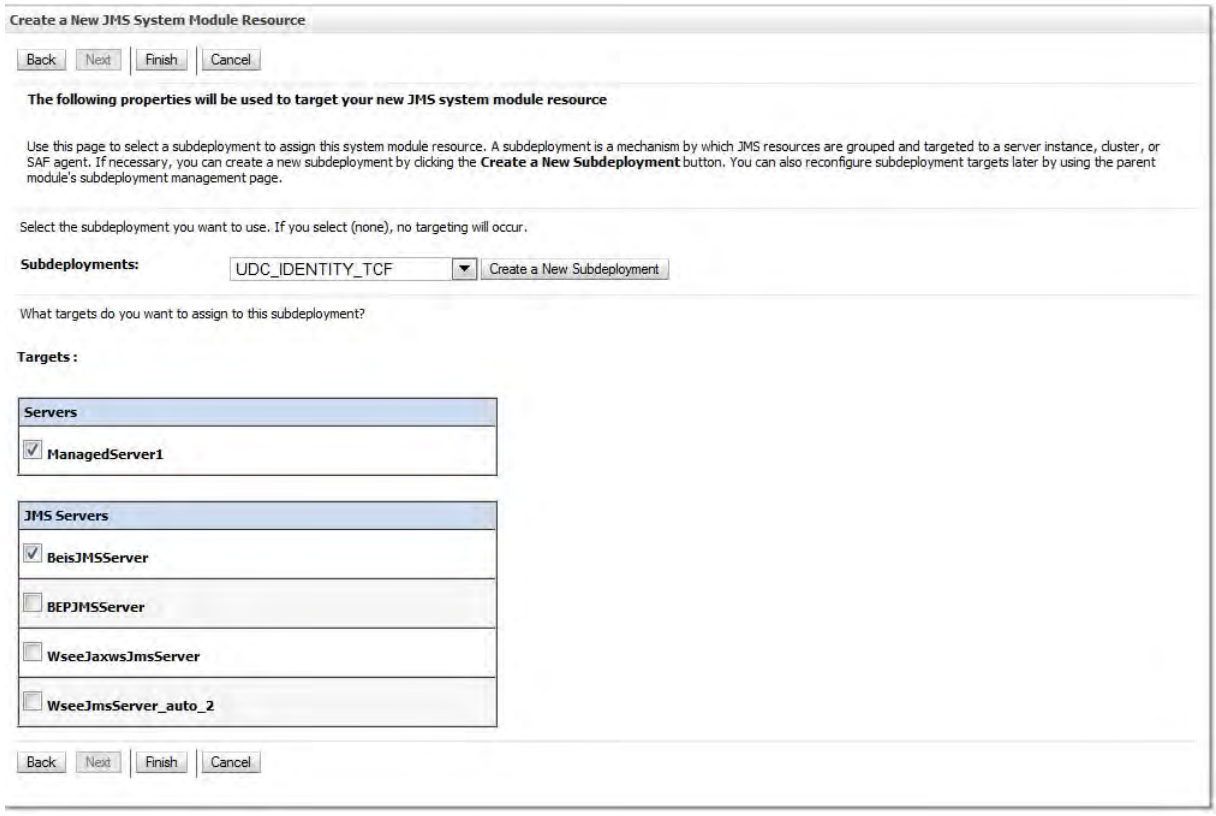


21. Click **Create a New Subdeployment**. The Create a New Subdeployment page is displayed.



22. Enter the name of the subdeployment (for example, *UDC_IDENTITY_TCF*) in the **Subdeployment Name** field.

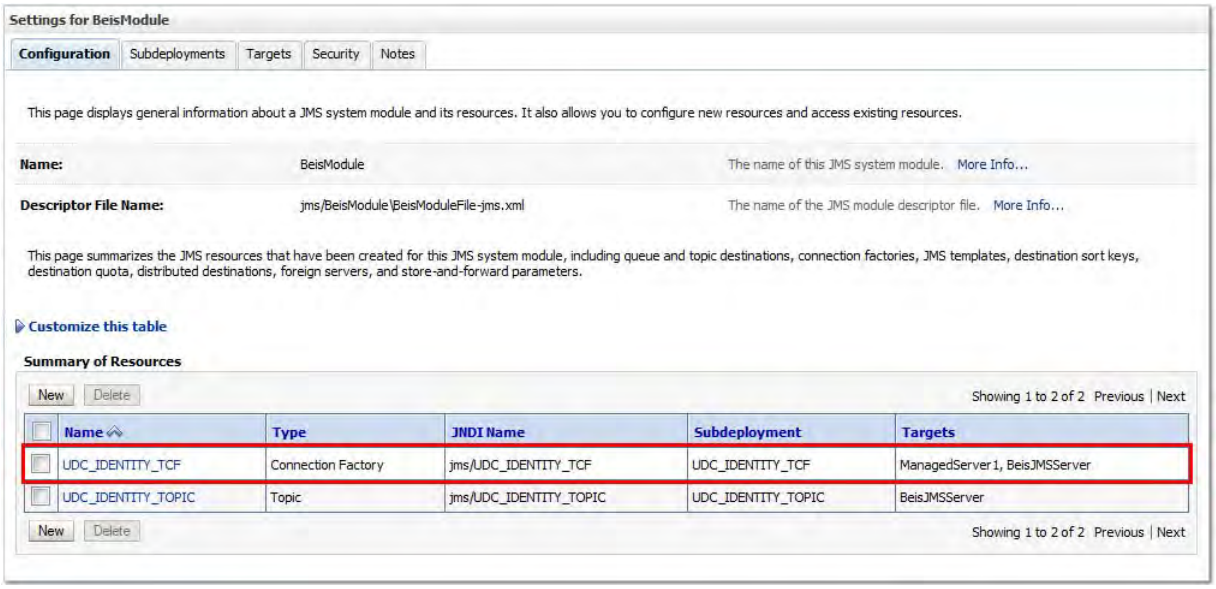
23. Click **OK**. The Create a New JMS System Module Resource page is redisplayed.



24. Select *UDC_IDENTITY_TCF* from the **Subdeployments** drop-down list.

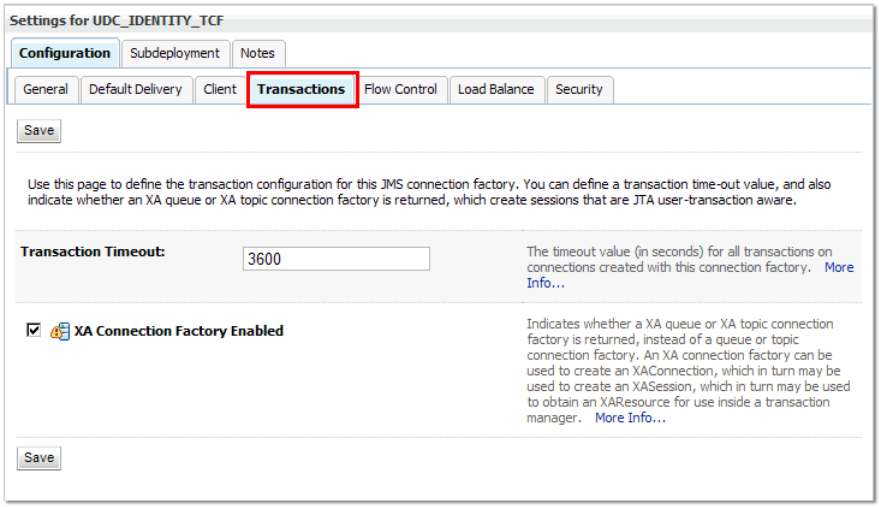
25. Select a target server and a JMS server for the connection factory.

26. Click **Finish**. The Settings page is redisplayed with the new connection factory.



27. Click the name of the connection factory that you just created. The Settings page for the connection factory is displayed.

28. Select the Transactions tab.



29. Select **XA Connection Factory Enabled**.

30. Click **Save**.

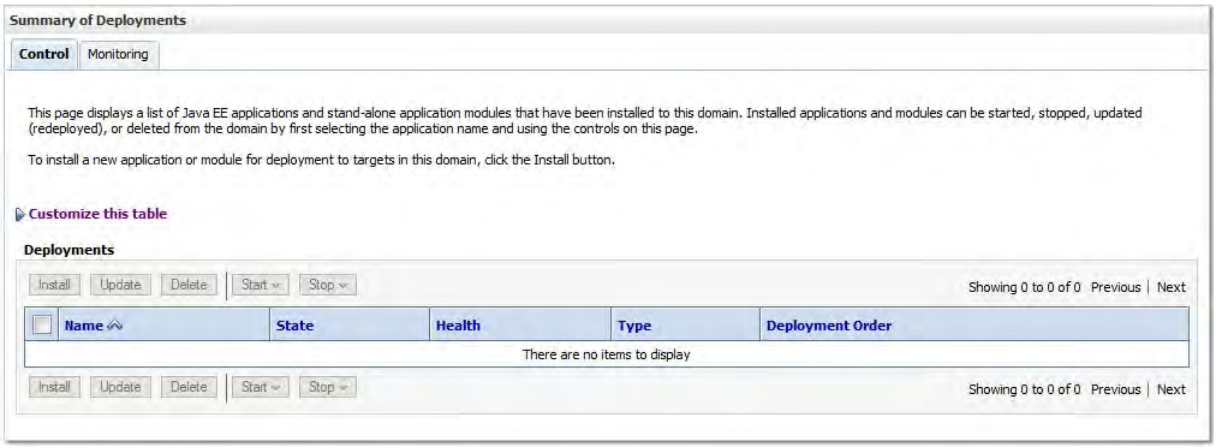
31. In the Change Center pane, click **Activate Changes**.

Step 12 Deploy the Banner Identity Gateway

Use the following steps to install the Banner Identity Gateway to the Oracle WebLogic Server.

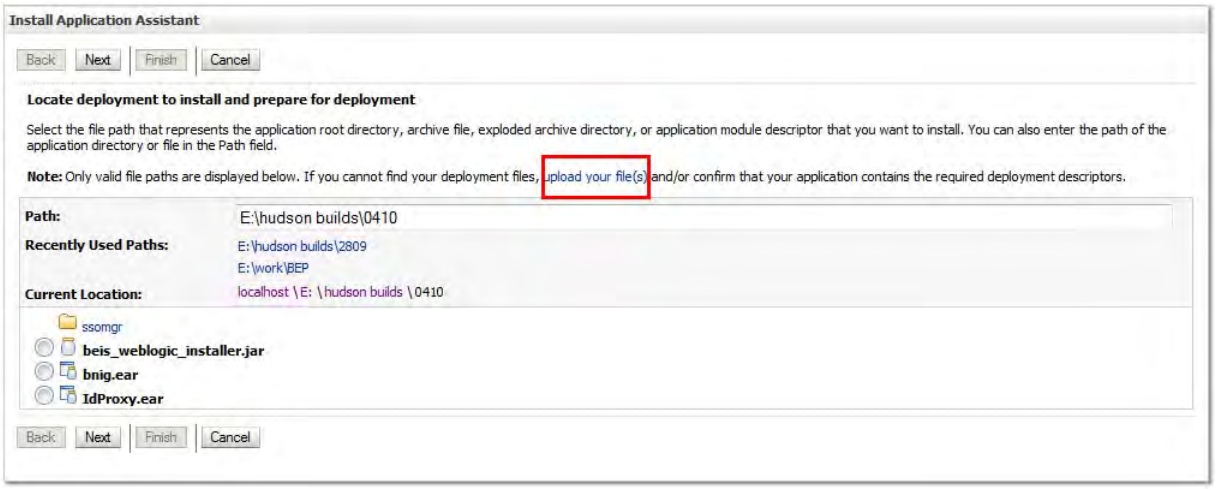
1. In the Domain Structure pane, click **Deployments**.

The Summary of Deployments page is displayed.

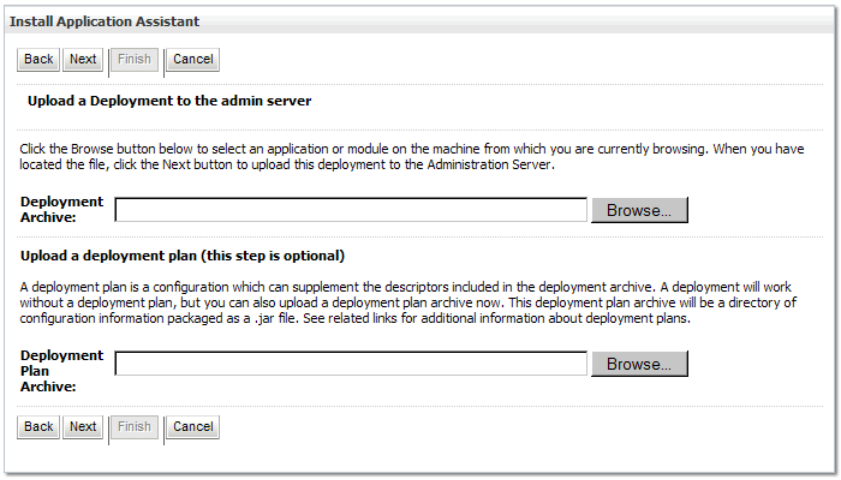


2. In the Change Center pane, click **Lock & Edit**.

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



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

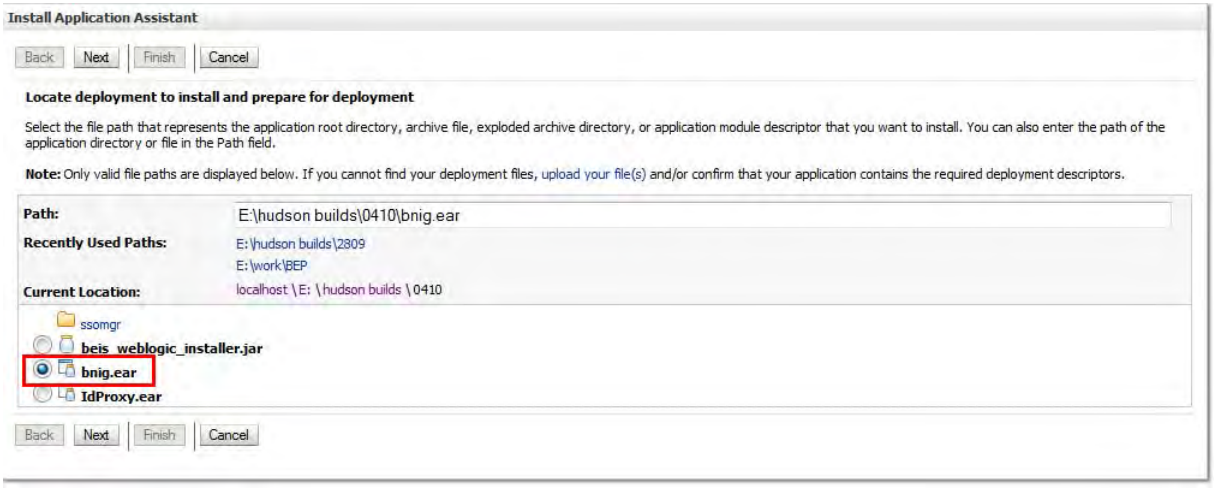


5. Select the file to be uploaded:

   5.1. In the **Deployment Archive** field, click **Browse** and navigate to the `bnig.ear` file.

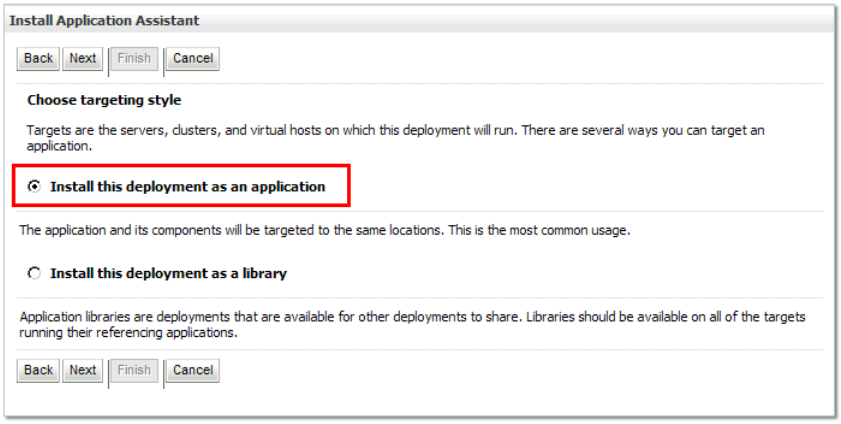
   5.2. Select the file and click **Open**.

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



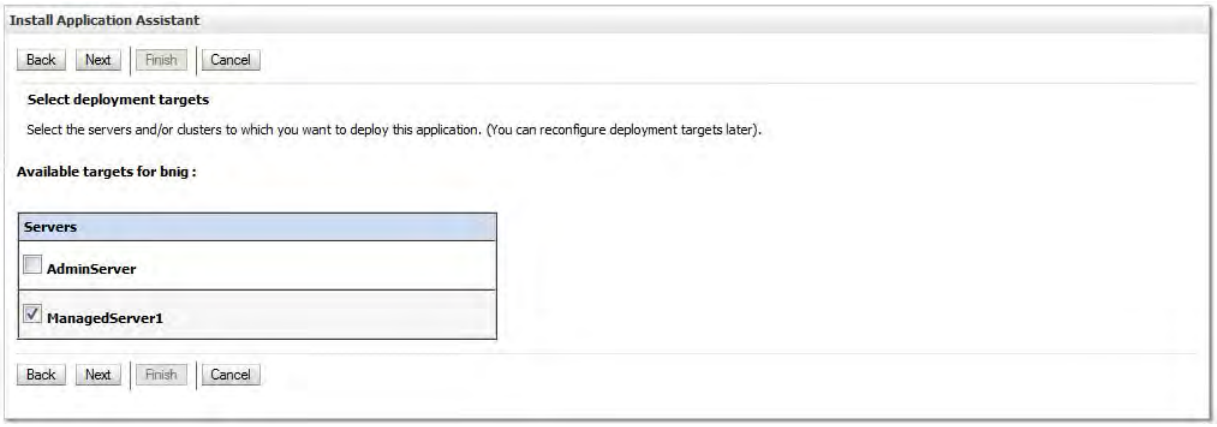
7. Select the `bnig.ear` file from the list.

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



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

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

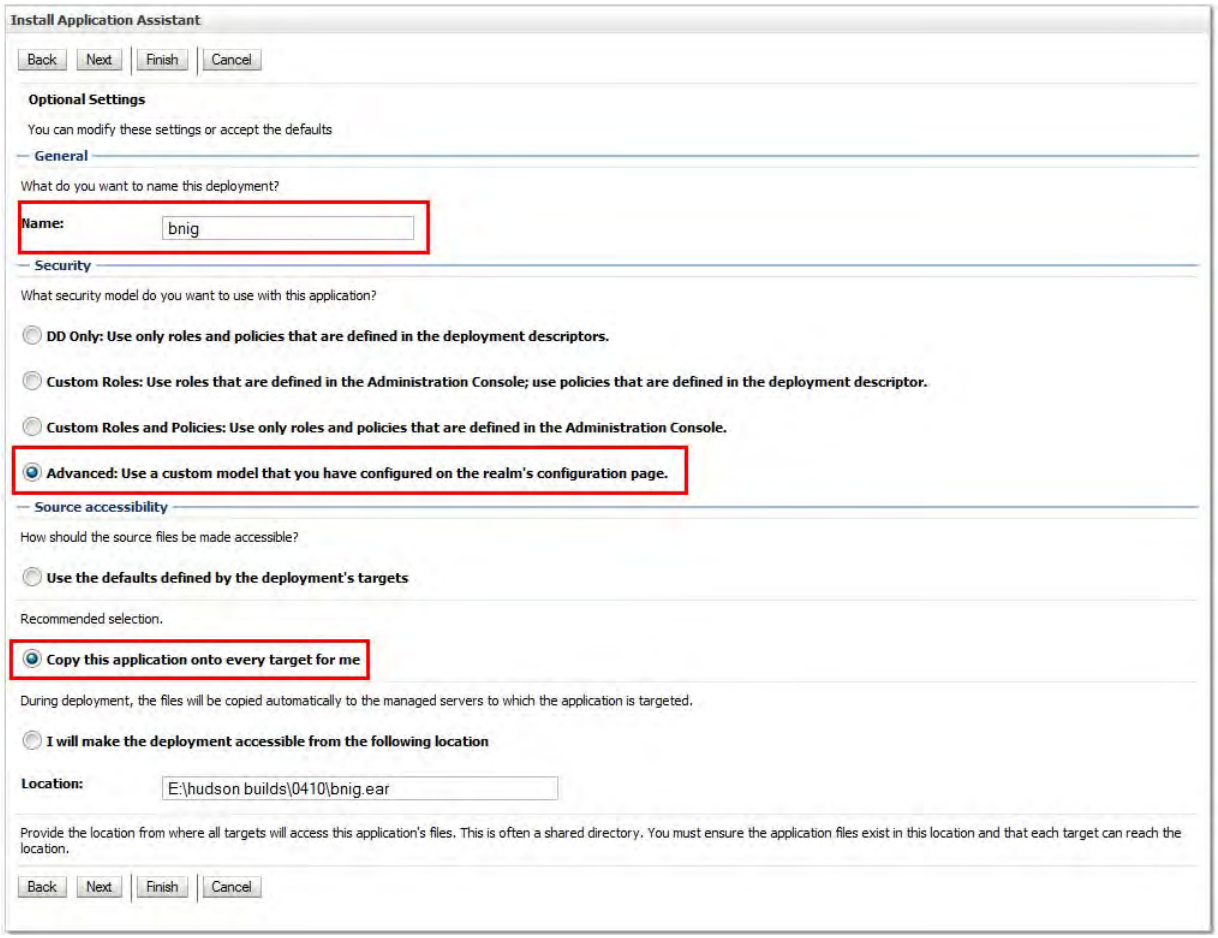


11. Select the server where the application should be deployed. (The application can be installed on an existing server.)

Note

SunGard Higher Education recommends deploying applications to a WebLogic Managed Server and not to the Administration Server. If you do not see the preceding page, you should check your WebLogic server configuration to ensure that a Managed Server is available for deployment of applications. If a Managed Server is not available, the application will be deployed to the Administration Server, which is not a recommended configuration. For more information, consult the Oracle WebLogic Server Documentation Library. ■

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

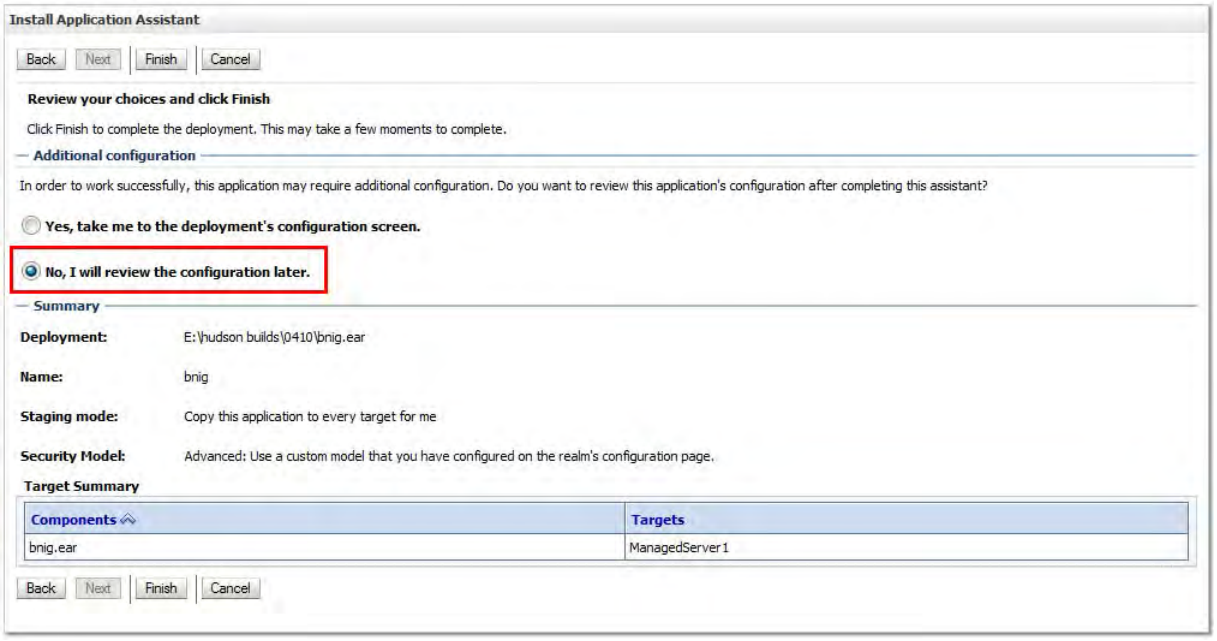


13. Enter a name for the application (for example, *bnig*) in the **Name** field.

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

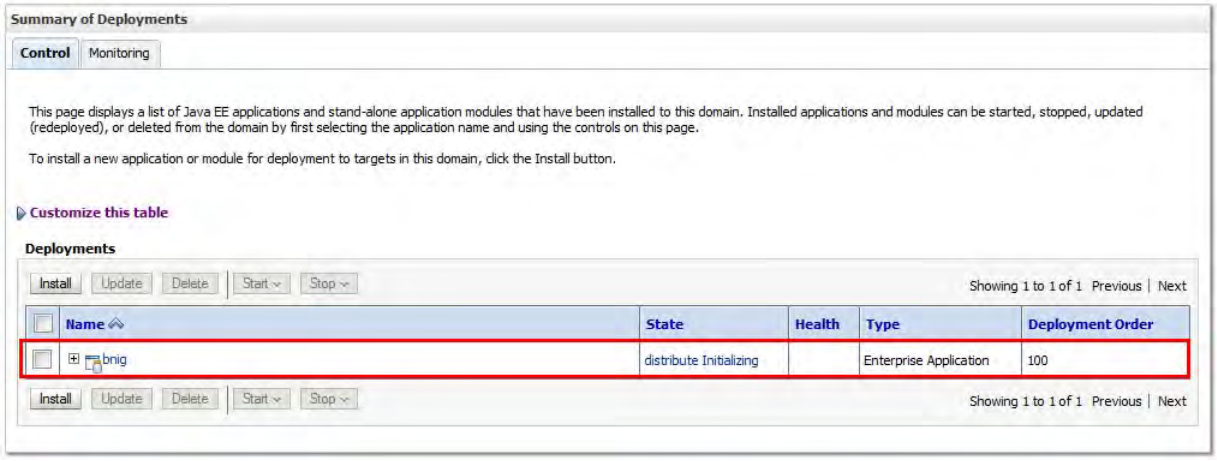
15. Select **Copy this application onto every target for me**.

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



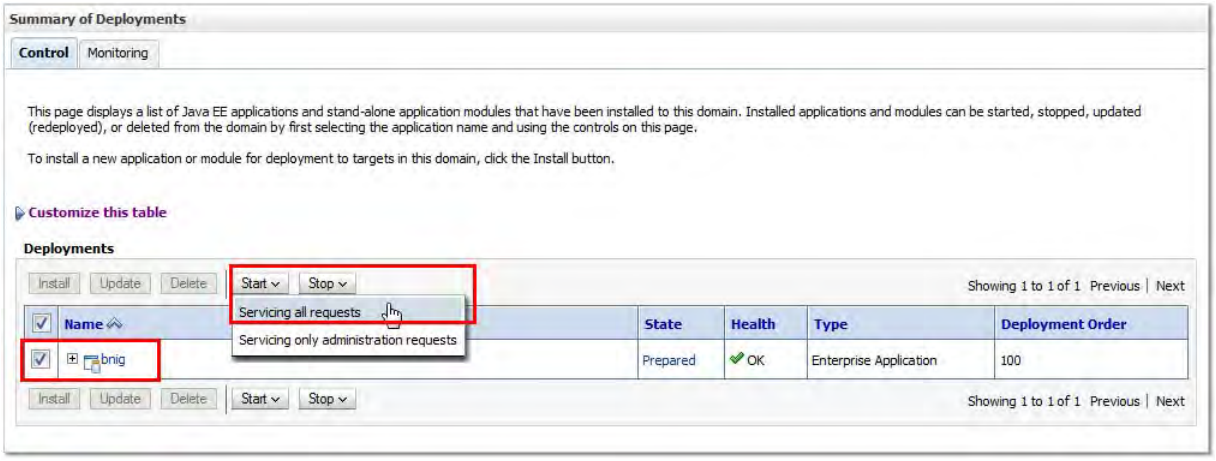
17. Select **No, I will review the configuration later**.

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



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

20. Start the newly deployed application as follows:



20.1. Select the newly deployed application.

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

20.3. Click **Yes**.

Configuration

You can use the Banner Identity Gateway administrative interface to configure the Gateway.

Configure Gateway metadata

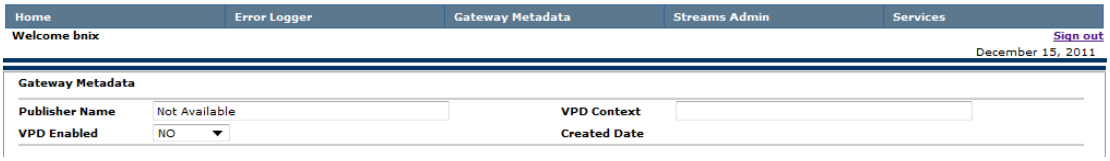
Use the following steps to configure the Gateway metadata.

1. Connect to the Banner Identity Gateway:

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

2. Log in with the user name and password that were mapped to the `bnixadmin` role (OAS) or the `bnigAdminGroup` group (Oracle WebLogic). The welcome page is displayed.

3. Select Gateway Metadata > Update from the menu bar. The Gateway Metadata page is displayed.



4. In the **Publisher Name** field, enter the unique name of the Banner instance that provides user identity information.

5. Click **Save**.

Modify Gateway metadata

Only one configuration for the Gateway metadata is permitted. Use the following steps to modify the Gateway metadata:

1. Connect to the Banner Identity Gateway:

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

2. Log in with the user name and password that were mapped to the `bnixadmin` role (OAS) or the `bnigAdminGroup` group (Oracle WebLogic). The welcome page is displayed.

3. Select Gateway Metadata > Update from the menu bar. The Gateway Metadata page displays the metadata attributes.

4. Edit the attributes. All attributes, except **Created Date**, can be edited.

5. Click **Save**.

If the Gateway metadata is not configured, the following message is displayed:

```
No data Found
```

If the Gateway metadata is not configured and more than one user tries to save the configuration, the following message is displayed:

```
Publisher Already Created
```

Administration

You can use the Banner Identity Gateway administrative interface to perform the following tasks:

- "Search for and edit errors"
- "Display all edits for an error"
- "Remove an error log"
- "View the status of Oracle Streams processes"
- "Start Oracle Streams IAM processes"
- "Stop Oracle Streams IAM processes"
- "Load updated capture rules"

The administrative interface is accessed at the following URL:

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

Warning

If any changes to the URL were made during installation, the new URL must be specified.

Log in with the user name and password that were mapped, during installation, to the `bnixadmin` role (OAS) or the `bnigAdminGroup` group (Oracle WebLogic 11g).

Search for and edit errors

Use the following steps to display and edit errors that the Gateway encountered.

1. Select Error Logger from the menu bar on the Gateway administrative interface. The UDC Identity Error Log is displayed.

2. Enter your search criteria in the top part of the page.

3. Click **Query**. The page displays errors that meet the search criteria.

   Displayed information includes the exception that was raised, the current status of the error, and the date and time when the error occurred. By default, displayed errors are filtered by status. Deleted errors are not displayed.

4. To see details for an error, select the error and click **View Details.** The Error Log Details page is displayed.

5. To edit an error, use the following steps:

   5.1. Edit information in the **Event** field. This part of the page can be edited and restored in the database. The entry must be a valid document format for the identity event.

   5.2. Click **Update**.

Display all edits for an error

Use the following steps to display all changes that were made to an error.

1. Select Error Logger from the menu bar on the Gateway administrative interface. The UDC Identity Error Log is displayed.

2. Enter your search criteria in the top part of the page.

3. Click **Query**. The page displays errors that meet the search criteria.

4. Select the error you want to display.

5. Click **Audit**. The UDC Identity Error Log Audit page is displayed.

6. Select the error that you want to display.

7. Click **View Details**. The Error Log Details page is displayed with all the changes made to the error. Audit information displayed on the page *cannot* be edited.

Remove an error log

You can erase or remove a single error log or all error logs:

- To erase or remove an error log, select the error and click **Delete**. The error is flagged as deleted but is not physically removed from the system.
- To erase or remove all logs, click **Delete All**. This marks all errors as deleted.

Deleted records can still be queried from this page. At the top of the page in the **Error Status** drop-down list, select the *Deleted* option to view the deleted records.

View the status of Oracle Streams processes

Oracle Streams processes capture and apply changes that occur in the Banner database. Specifically, the Oracle Streams IAM capture and apply processes deal with changes to Banner identity information. If a problem occurs with message production, you can display information about the Oracle Streams IAM processes as a starting point for troubleshooting the problem.

Select Streams Admin from the menu bar on the Gateway administrative interface. The Banner Streams Identity Processes page displays information about the capture and apply processes:

Process	Displayed Information
Capture process	This section displays details on the status of the capture process and queue. A status of *ENABLED* indicates the capture process is running.
Apply process	This section displays details on the status of the apply process and its consumption of a specified capture queue. A status of *IDLE*, *DEQUEUE MESSAGES*, or *SCHEDULE MESSAGES* indicates the apply process is running.

Start Oracle Streams IAM processes

Use the following steps to start Oracle Streams IAM processes.

1. Select Streams Admin from the menu bar on the Gateway administrative interface. The Banner Streams Process for Identity page is displayed.

2. Verify that the desired process is not already running:

   2.1. Before starting the capture process, **Capture Process State** must equal *ABORTED*.

   2.2. Before starting the apply process, **Apply Process State** must be null.

3. Select one of the following from the **Banner Streams Ops.** drop-down list:

   Start capture
   Start apply
   Start capture and apply

4. Click **Apply**.

Stop Oracle Streams IAM processes

Stopping the IAM capture and apply processes might be necessary when upgrading middle-tier consumers of events published by the apply process or when changing the capture process configuration. Use the following steps to stop Oracle Streams IAM processes.

1. Select Streams Admin from the menu bar on the Gateway administrative interface. The Banner Streams Process for Identity page is displayed.

2. Verify that the desired process is running:

   2.1. Before stopping the capture process, **Capture Process State** must equal *ENABLED*.

   2.2. Before stopping the apply process, **Apply Process State** must equal *IDLE*, *DEQUEUE MESSAGES*, or *SCHEDULE MESSAGES*.

3. Select one of the following from the **Banner Streams Ops.** drop-down list:

   Stop capture
   Stop apply
   Stop capture and apply

4. Click **Apply**.

Load updated capture rules

When capture rules are modified on the Streams Rules Configuration Form (GUASADM), the updated rules must be loaded to the Oracle Streams environment. Use the following steps to load updated IAM capture rules.

1. Select Streams Admin from menu bar on the Gateway administrative interface. The Banner Streams Process for Identity page is displayed.

2. Select *Load Capture Rules* from the **Banner Streams Ops.** drop-down list.

3. Click **Apply**. This stops the capture and apply processes, reconfigures the Oracle Streams environment with new rules, and re-starts the processes.

8 Enterprise Identity Proxy Services

The Enterprise Identity Proxy Services (Identity Proxy) application propagates identity to SPML-enabled applications. The application includes the following functionality:

- Consumes UDCIdentity messages from the UDCIdentity Topic
- Sends the messages, wrapped inside SPML requests, to designated Provisioning Service Providers (PSPs)
- Creates and manages UDC identities

The Identity Proxy can deliver a UDCIdentity message to multiple PSPs. Responses from the PSPs are stored and can be viewed by the administrator.

You can filter messages that are posted to a particular PSP by providing an XPath expression that must be matched for a message to be sent to the endpoint. This is done by configuring the Identity Proxy to send messages to a specific Provisioning Service Provider (PSP). (See "Add or update a PSP configuration" on page 8-55.) For example, you can configure the Identity Proxy to send UDCIdentity messages only for those persons who have a *STUDENT* institution role. This prevents the PSP from receiving UDCIdentity messages in which it is not interested.

This chapter gives instructions for installing the Identity Proxy, configuring the Identity Proxy, and performing administrative tasks.

Installation options

You can either install the Identity Proxy with a manual or an automated process.

Manual installation

You can use the application server console to manually install the Identity Proxy. The Identity Proxy is packaged in two zip files for manual installation:

- `Deployables\OC4J\Banner_IdentityProxy_full_release_Oracle.zip` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\Banner_IdentityProxy_full_release_Weblogic.zip` is used for installation on Oracle WebLogic Server 11g.

This chapter gives instructions for a manual installation on both servers.

Automated installation

You can use the automated Banner Enterprise Identity Services Installer to install the Banner® Identity Gateway and the Enterprise Identity Proxy Services in one installation process. If you are using a windows environment, the installer provides a graphical user interface. If you are using a non-windows environment, the installer provides a command line mode.

The installer is packaged in two archive files:

- `Deployables\OC4J\beis_oc4j_installer.jar` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\beis_weblogic_installer.jar` is used for installation on Oracle WebLogic Server 11g.

The installer can be used for new installations only; it cannot be used for upgrades. Refer to Chapter 9, “Automated Installer” for instructions on using the installer.

Installation on Oracle Application Server

`Deployables\OC4J\Banner_IdentityProxy_full_release_Oracle.zip` is used for installation on Oracle Application Server 10.1.3.4/5. This zip file contains an archive file named `idproxy.ear`. This file provides an implementation of the SPML Request Authority.

The Identity Proxy can be installed on an existing Oracle Application Server. The Identity Proxy and Gateway must be installed together in the same OC4J instance. BEIS applications should be deployed separately from other applications so they can be managed independently.

Note

The following steps are used for a manual installation of the Identity Proxy. If you prefer to use the automated installer, refer to Chapter 9, “Automated Installer”.

Use the following steps to install the Identity Proxy on OAS 10.1.3.4/5.

- Step 1, “Configure the database user and schema”
- Step 2, “Change processing parameters (optional)”
- Step 3, “Define the data source”
- Step 4, “Configure the security role and user”
- Step 5, “Configure the JMS queues”

- Step 6, “Install the Identity Proxy”
- Step 7, “Configure logging”

Step 1 Configure the database user and schema

Use the following steps to create the database user, tables, and packages that are required by the Identity Proxy.

1. Extract the `Banner_IdentityProxy_full_release.zip` archive file.
2. Go to the directory where the file was extracted. This directory is referred to as `<IDSVC_INSTALLER_HOME>`.

   ```
   cd <IDSVC_INSTALLER_HOME>/db-scripts/users
   ```

3. Run SQL*Plus and connect as DBA.
4. Execute the `identmgr.sql` script:

   ```
   sqlplus> @identmgr
   ```

5. When prompted, enter the following information:
   - Schema name for the Enterprise Identity Proxy Services (*identmgr*)
   - Tablespace name for the Enterprise Identity Proxy Services schema (for example, *identmgr_tb*)
   - Name of the datafile with the complete path (for *example, /u01/app/oracle/ORDBMS/10.2.0/dbs/idproxy.dbf*)
   - Password for the Enterprise Identity Proxy Services schema (for example, *u_pick_it*)
6. Close SQL*Plus.

   ```
   sqlplus> exit
   ```

7. Change the directory to `<IDSVC_INSTALLER_HOME>/db-scripts/tables`.

   ```
   cd <IDSVC_INSTALLER_HOME>/db-scripts/tables
   ```

8. Run SQL*Plus and connect as the `identmgr` user.
9. Execute the following scripts to create the tables:

   ```
   sqlplus> @setup
   sqlplus> exit
   ```

10. Change to the directory <IDSVC_INSTALLER_HOME>/db-scripts/packages.

11. Run SQL*Plus and connect as the identmgr user.

12. Execute the following command to create the database packages:

```
sqlplus> @iokp_build
```

13. Exit SQL *Plus.

Step 2 Change processing parameters (optional)

The Identity Proxy is delivered with default values for processing parameters in the spml.properties file, which is embedded in the application. These parameters can be changed to improve performance, depending on the capabilities of the PSPs to which the Identity Proxy provisions.

For most installations, changes are not necessary. For new installations, it is recommended that changes are not made and that this step is skipped. However, if your installation requires performance tuning, some parameters can be adjusted to meet your requirements.

Use the following steps to change any processing parameter values in the properties file.

1. Copy IdProxy.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 IdProxy.ear
   ```

   The extract contains an archive named IdProxyEJB.jar.

3. From <EAR_HOME> execute the following command:

   ```
   jar xvf IdProxyEJB.jar spml.properties
   ```

4. Open spml.properties, which is extracted under <EAR_HOME>.

5. Edit any of the following default values to meet your requirements:

Parameter	Description	Default Value
spml_retry_count	Number of times sending the UDCIdentity message to a particular endpoint is retried. After this number of times, the message is marked as *FAILURE*.	*5*

Parameter	Description	Default Value
`spml_batch_size`	Maximum number of UDCIdentity messages sent to a Provisioning Service Target (PST) in one task invocation.	*60*
`spml_task_interval`	Delay before starting the task for the first time (milliseconds). Only for the very first invocation.	*30000* (30 seconds)
`spml_task_delay`	Time period between repeated task executions (milliseconds).	*5000* (5 seconds)

6. Save the changes.

7. From `<EAR_HOME>` execute the following command to rebuild `IdProxyEJB.jar`:

   ```
   jar uvf IdProxyEJB.jar spml.properties
   ```

8. After this command runs successfully, delete the `spml.properties` file under the `<EAR_HOME>` directory.

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

   ```
   jar cvf IdProxy.ear *.war *.jar lib/* META-INF/*
   ```

The rebuilt `IdProxy.ear` is used for installation.

Step 3 Define the data source

A data source provides the connection properties to the Banner database. Use the following steps to define the data source that is used to access the database schema created for the Identity Proxy.

1. Connect to the Oracle Enterprise Manager:

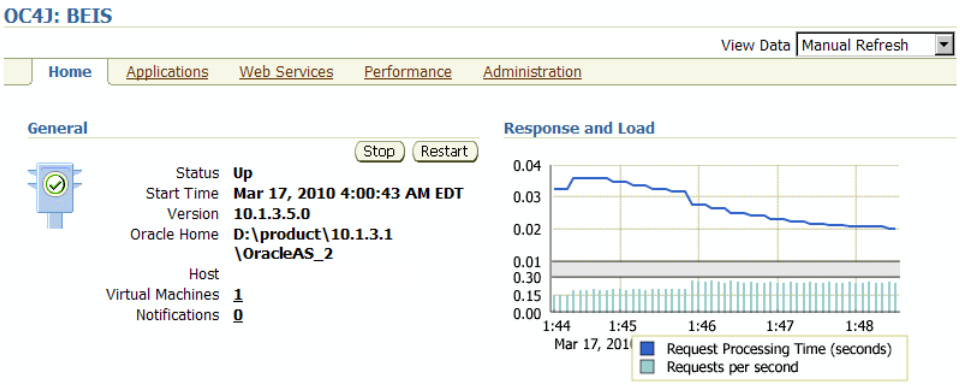
   ```
   http://<host>:<port>/em
   ```

   The console is displayed.

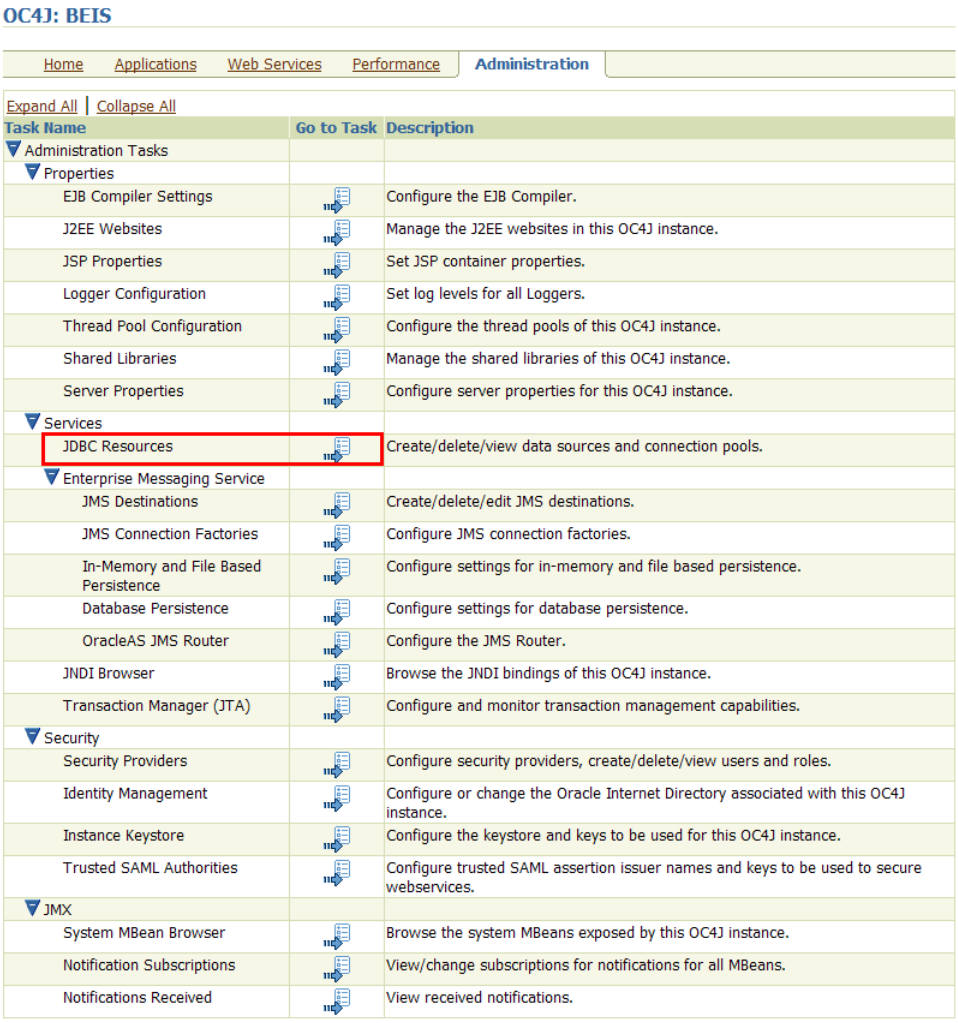
2. Click the name of the OC4J instance that will host the Identity Proxy.

   The application must be installed in the same instance with the Banner Identity Gateway application. Banner Enterprise Identity Services (BEIS) applications must be installed separately from other applications so they can be independently managed.

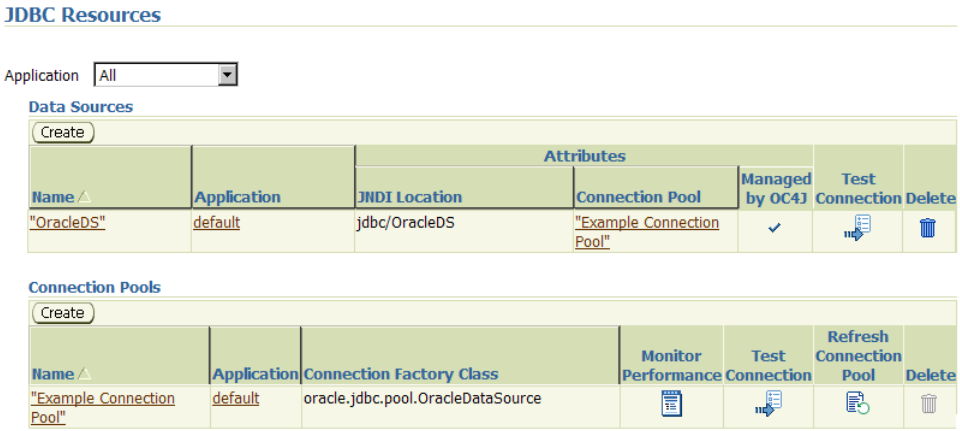
The Home page for the selected instance is displayed.



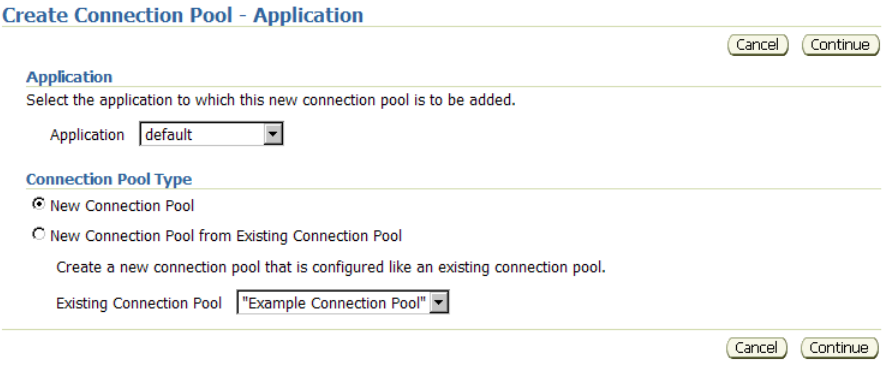
3. Select the **Administration** tab. A list of tasks is displayed.



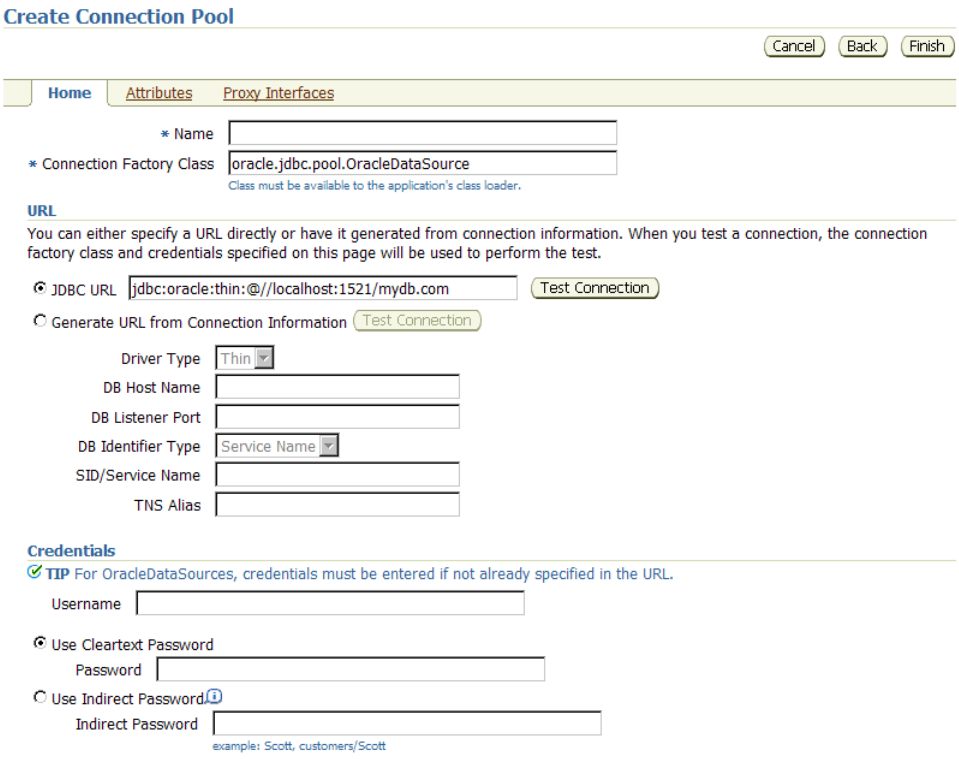
4. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.



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



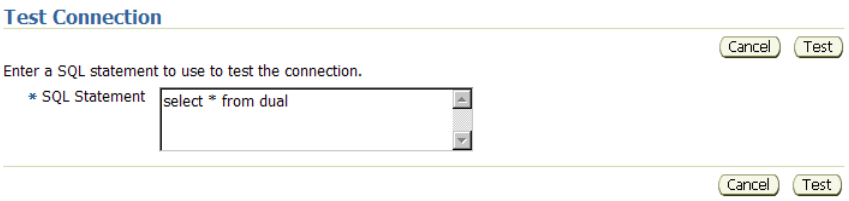
6. Click **Continue**. The Create Connection Pool page is displayed.



7. Enter the following information to set up the connection pool for the `identmgr` schema:

Name	*identmgr_banner_pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*identmgr*
Use Cleartext Password	Select **Use Cleartext Password** and enter a password for the `identmgr` schema.

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



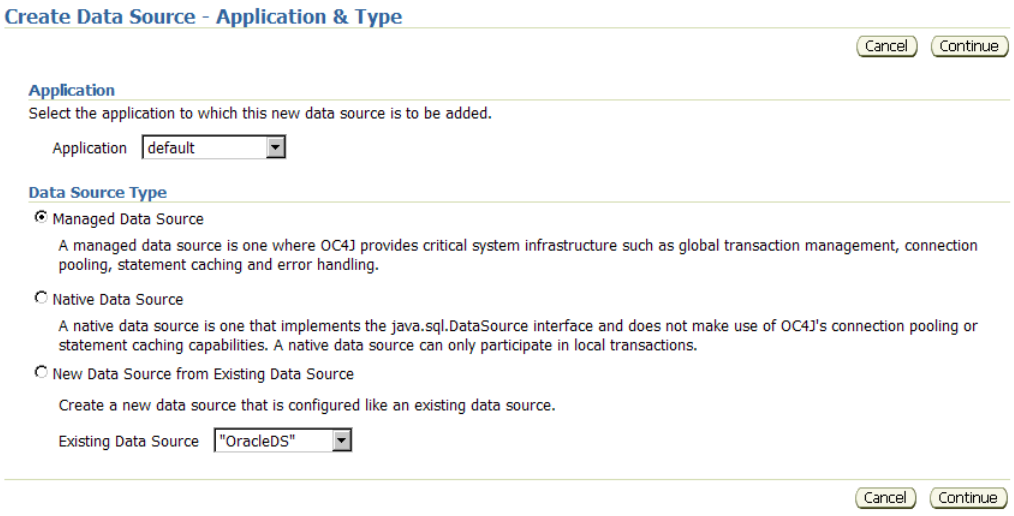
9. Click **Test** to test the connection pool for the `identmgr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

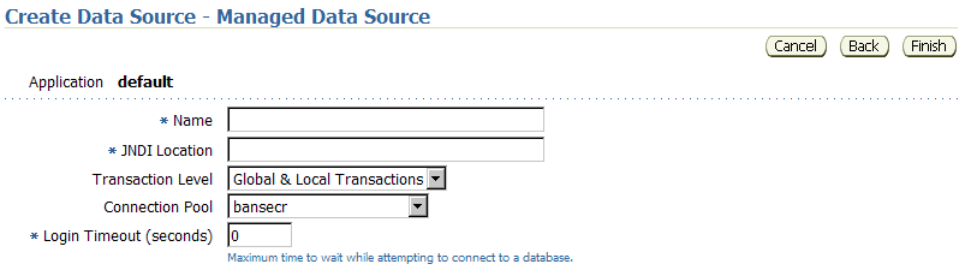
   9.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

10. Click **Finish**.

11. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.



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



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

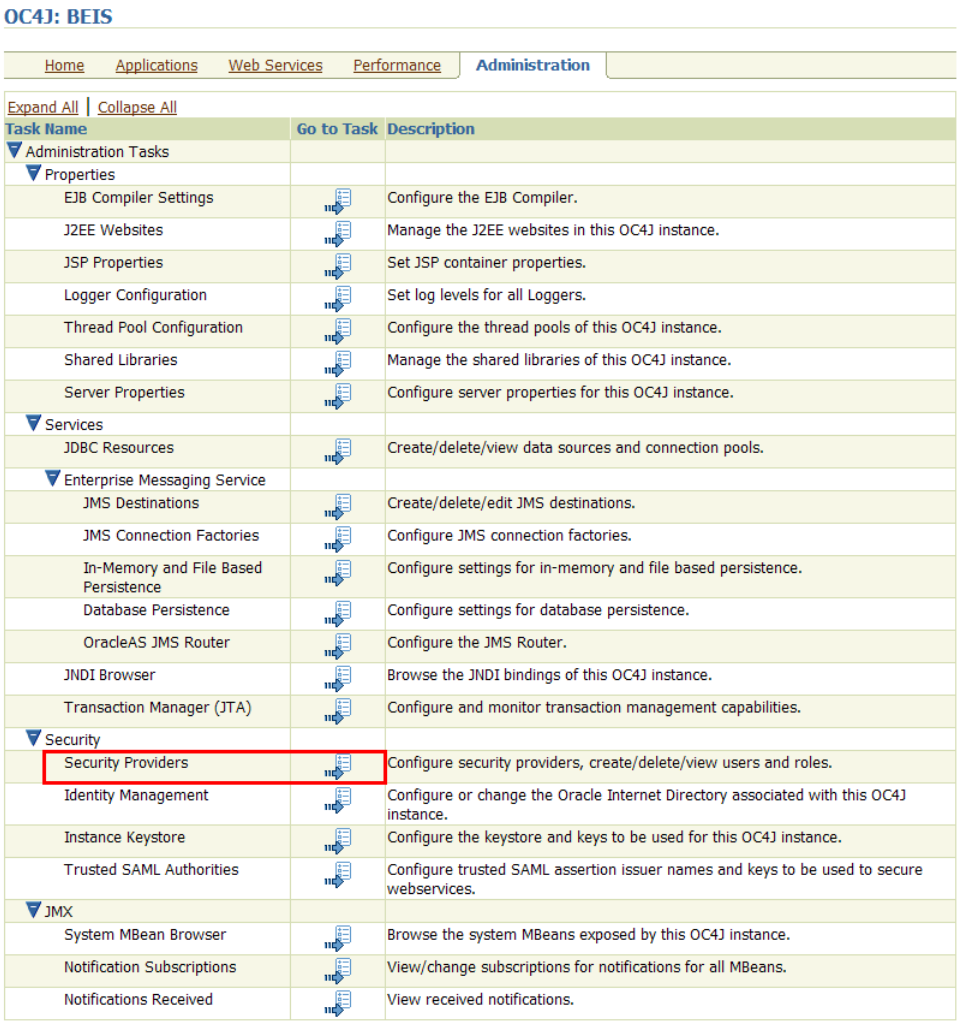
Name	*Identmgr_banner*
JNDI Location	*jdbc/identmgr*
Connection Pool	*identmgr_banner_pool*

14. Click **Finish**.

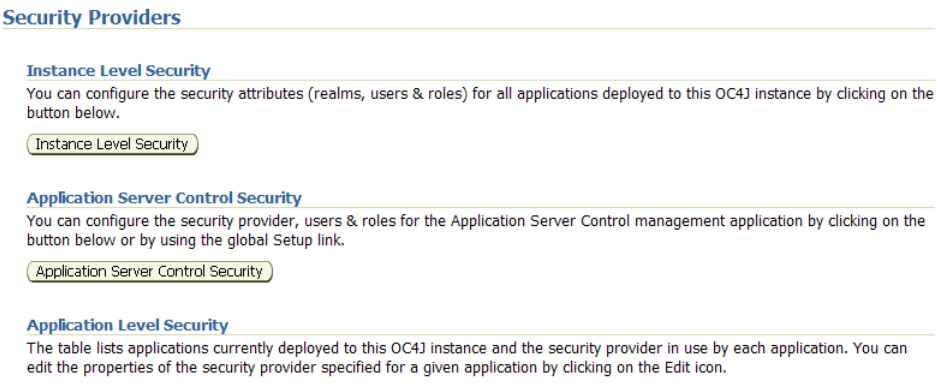
Step 4 Configure the security role and user

Use the following steps to add the `idpadmin` role and an administrative user to the Enterprise Identity Proxy Services application. This role and user are required for accessing the Identity Proxy administrative interface.

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

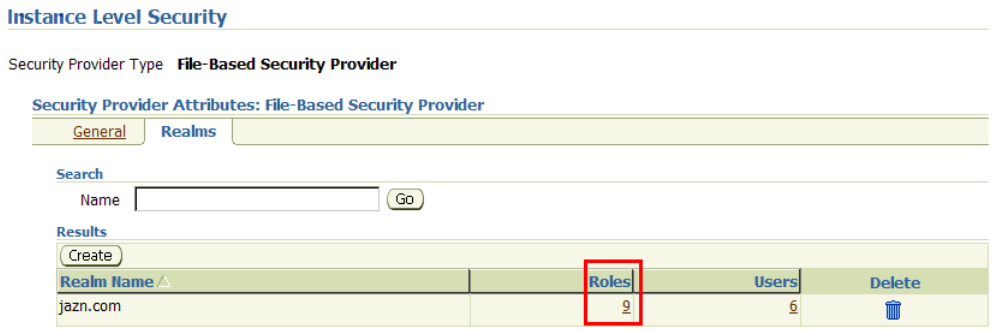


2. Select **Security Providers** in the Security section. The Security Providers page is displayed.



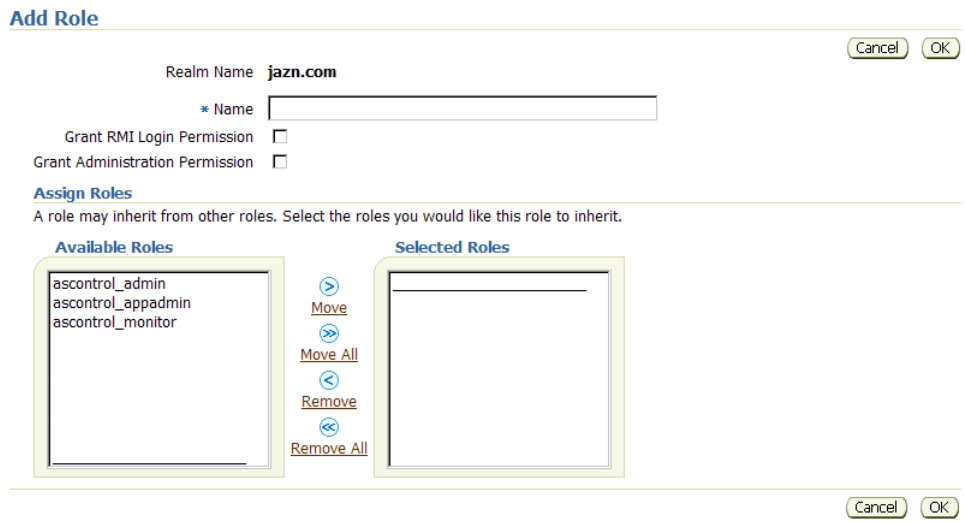
3. Click **Instance Level Security**. The Instance Level Security page is displayed.

4. Select the **Realms** tab.



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

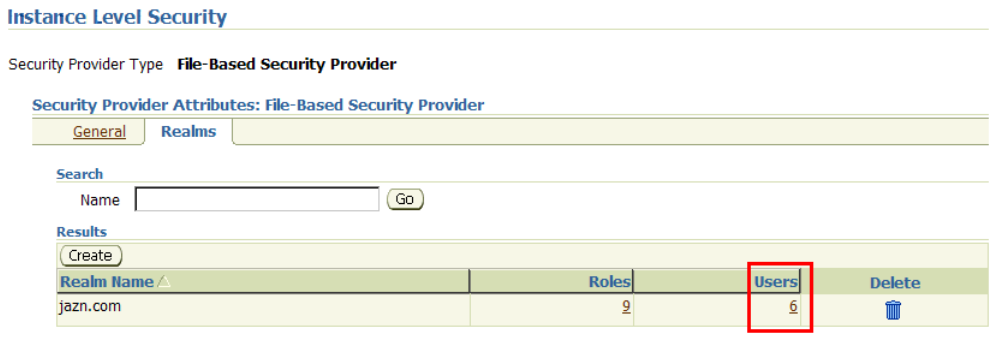
6. Click **Create**. The Add Role page is displayed.



7. Enter *idpadmin* in the **Name** field.

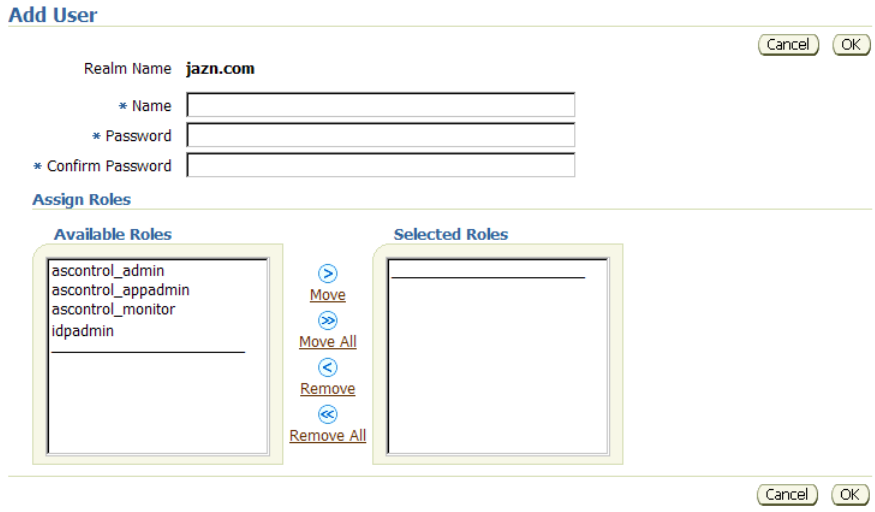
8. Click **OK**. The Roles page is redisplayed with the new role.

9. Return to the Instance Level Security page.



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

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



12. Enter the following information to create a user:

Name	*idproxy* (This is an example. Enter the name of your choice.)
Password	Password used by the user to log in to the Identity Proxy administrative interface
Confirm Password	Confirmation of the password

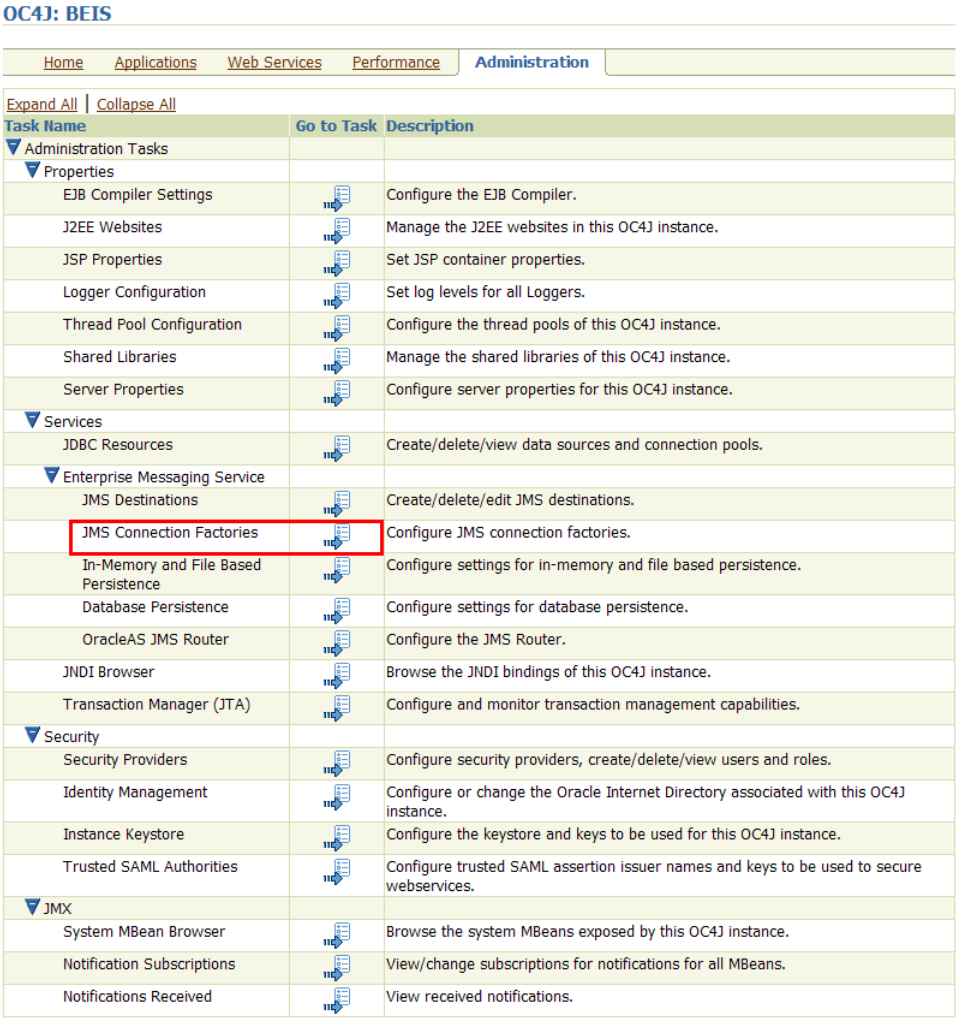
13. In the Assign Roles section, select the *idpadmin* role in the **Available Roles** list and move it to the **Selected Roles** list.

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

Step 5 Configure the JMS queues

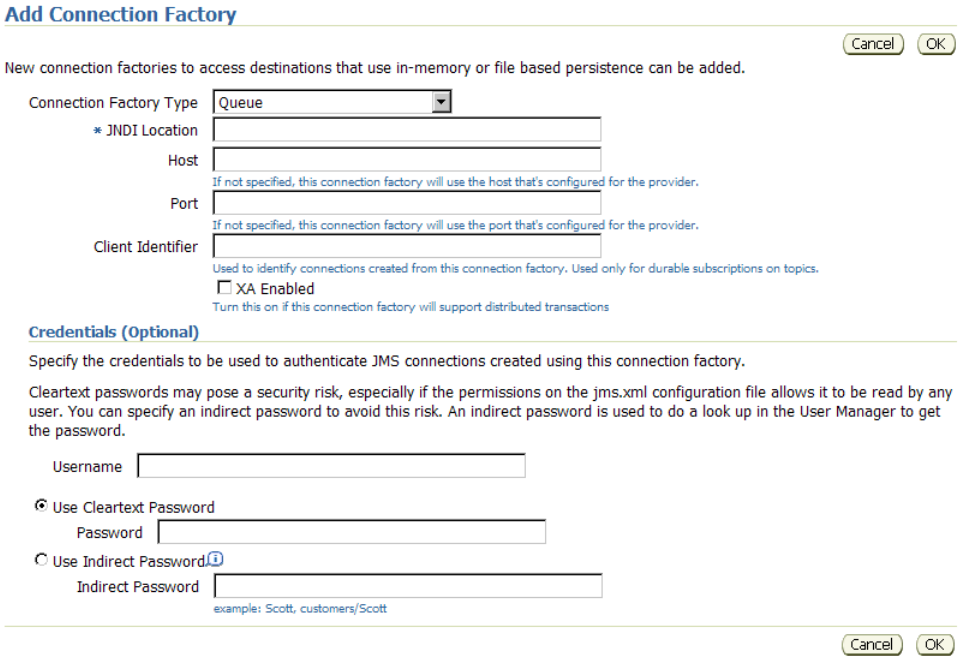
The Enterprise Identity Proxy Services application uses Java messaging to move UDCIdentity messages among its various services. Use the following steps to add the required JMS queues for the Identity Proxy.

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



2. Select **JMS Connection Factories** from the Enterprise Messaging Service section. The JMS Connection Factories page is displayed.

3. Click **Create New**. The Add Connection Factory page is displayed.



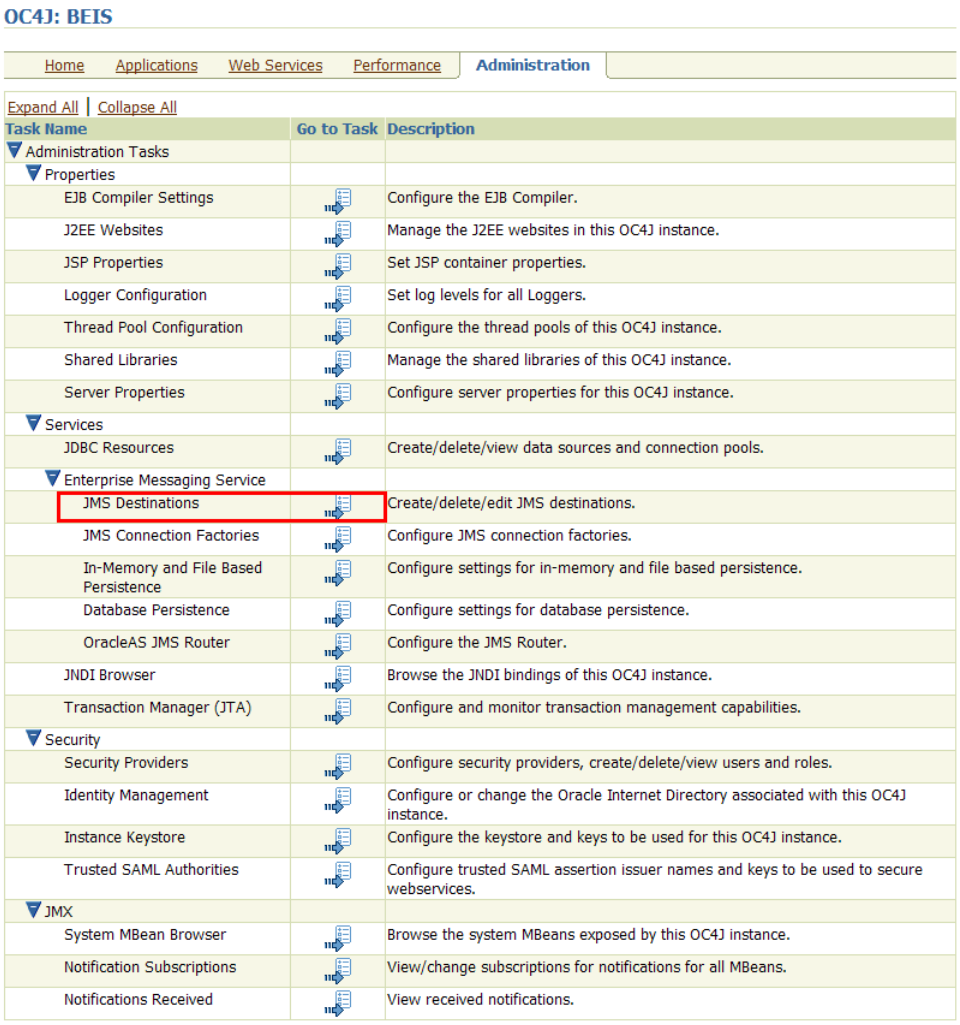
4. Enter the following information to create a connection factory:

Connection Factory Type *Queue*

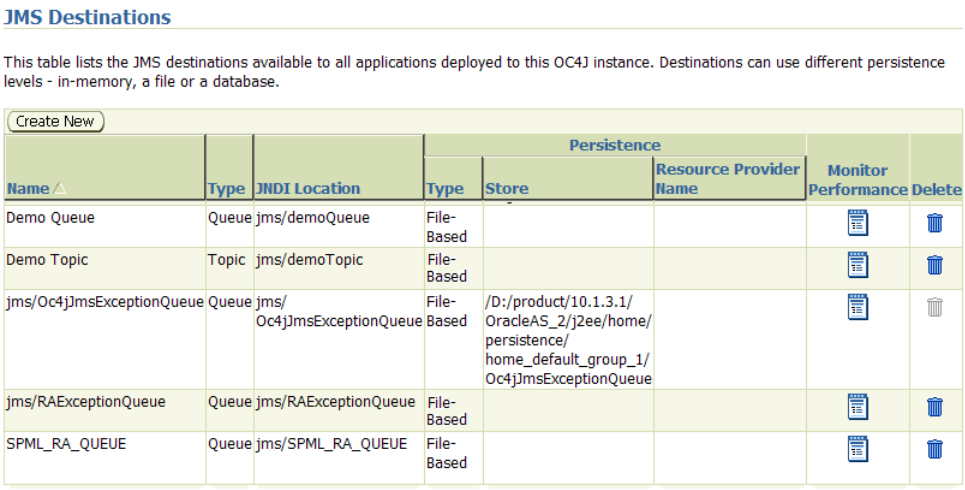
JNDI Location *jms/SPML_RA_QCF*

5. Click **OK**. The JMS Connection Factories page is redisplayed.

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



7. Select **JMS Destinations** in the Enterprise Messaging Service section. The JMS Destinations page is displayed.



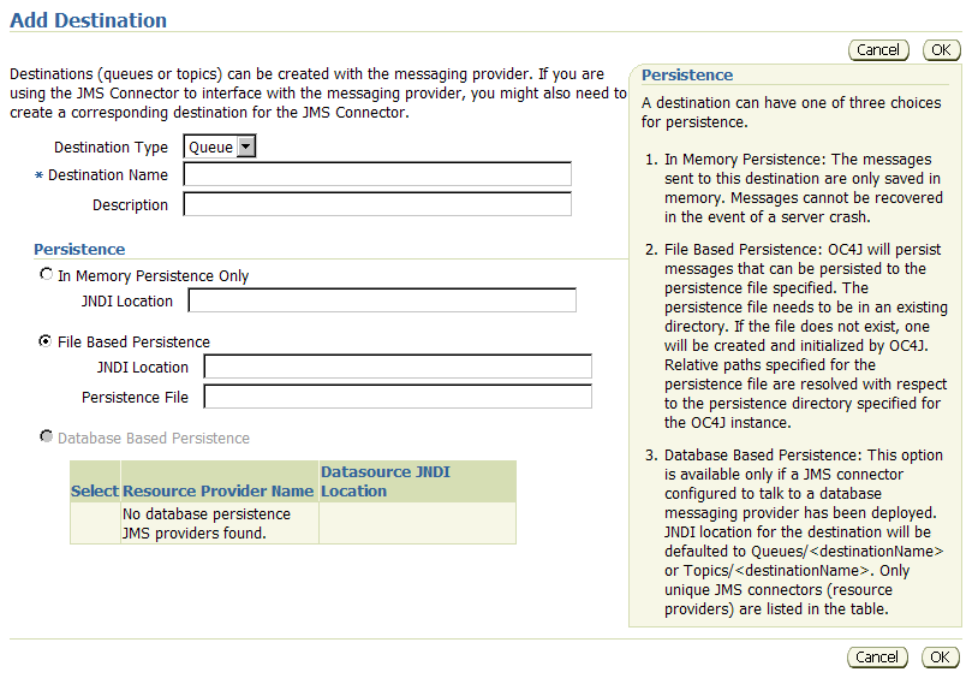
JMS Destinations

This table lists the JMS destinations available to all applications deployed to this OC4J instance. Destinations can use different persistence levels - in-memory, a file or a database.

Create New

Name	Type	JNDI Location	Persistence Type	Persistence Store	Persistence Resource Provider Name	Monitor Performance	Delete
Demo Queue	Queue	jms/demoQueue	File-Based				
Demo Topic	Topic	jms/demoTopic	File-Based				
jms/Oc4jJmsExceptionQueue	Queue	jms/Oc4jJmsExceptionQueue	File-Based	/D:/product/10.1.3.1/OracleAS_2/j2ee/home/persistence/home_default_group_1/Oc4jJmsExceptionQueue			
jms/RAExceptionQueue	Queue	jms/RAExceptionQueue	File-Based				
SPML_RA_QUEUE	Queue	jms/SPML_RA_QUEUE	File-Based				

8. Click **Create New**. The Add Destination page is displayed.



Add Destination

Cancel OK

Destinations (queues or topics) can be created with the messaging provider. If you are using the JMS Connector to interface with the messaging provider, you might also need to create a corresponding destination for the JMS Connector.

Destination Type: Queue

* Destination Name:

Description:

Persistence

In Memory Persistence Only

JNDI Location:

File Based Persistence

JNDI Location:

Persistence File:

Database Based Persistence

Select	Resource Provider Name	Datasource JNDI Location
	No database persistence JMS providers found.	

Persistence

A destination can have one of three choices for persistence.

1. In Memory Persistence: The messages sent to this destination are only saved in memory. Messages cannot be recovered in the event of a server crash.
2. File Based Persistence: OC4J will persist messages that can be persisted to the persistence file specified. The persistence file needs to be in an existing directory. If the file does not exist, one will be created and initialized by OC4J. Relative paths specified for the persistence file are resolved with respect to the persistence directory specified for the OC4J instance.
3. Database Based Persistence: This option is available only if a JMS connector configured to talk to a database messaging provider has been deployed. JNDI location for the destination will be defaulted to Queues/<destinationName> or Topics/<destinationName>. Only unique JMS connectors (resource providers) are listed in the table.

Cancel OK

9. Enter the following information to add a JMS destination:

Destination Type *Queue*

Destination Name *SPML_RA_QUEUE*

Description *SPML_RA_QUEUE*

10. Select **File Based Persistence** and enter the following information:

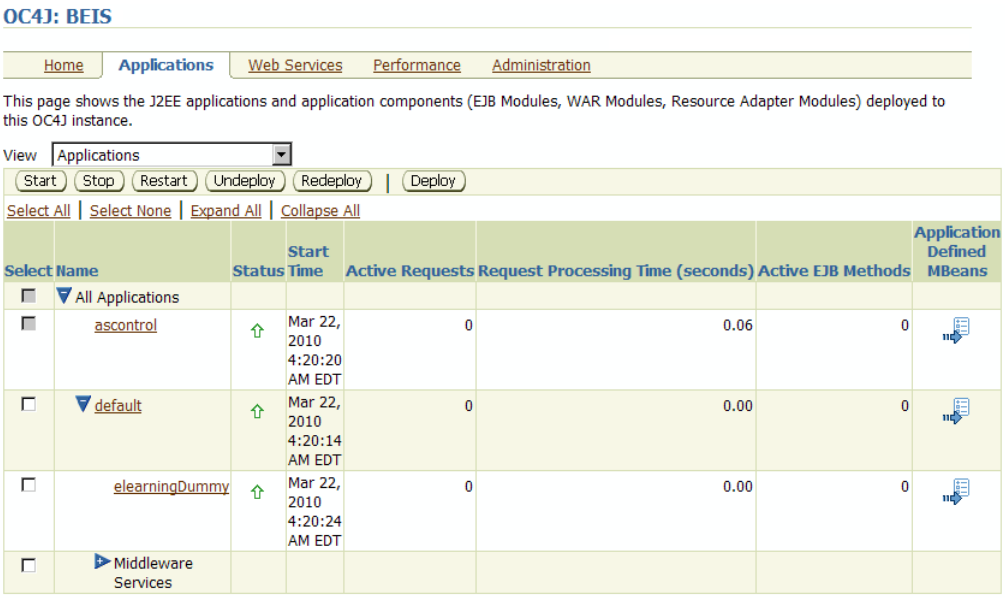
JNDI Location	*jms/SPML_RA_QUEUE*
Persistence File	*SPML_RA_QUEUE*

11. Click **OK**. The JMS Destinations page is redisplayed.

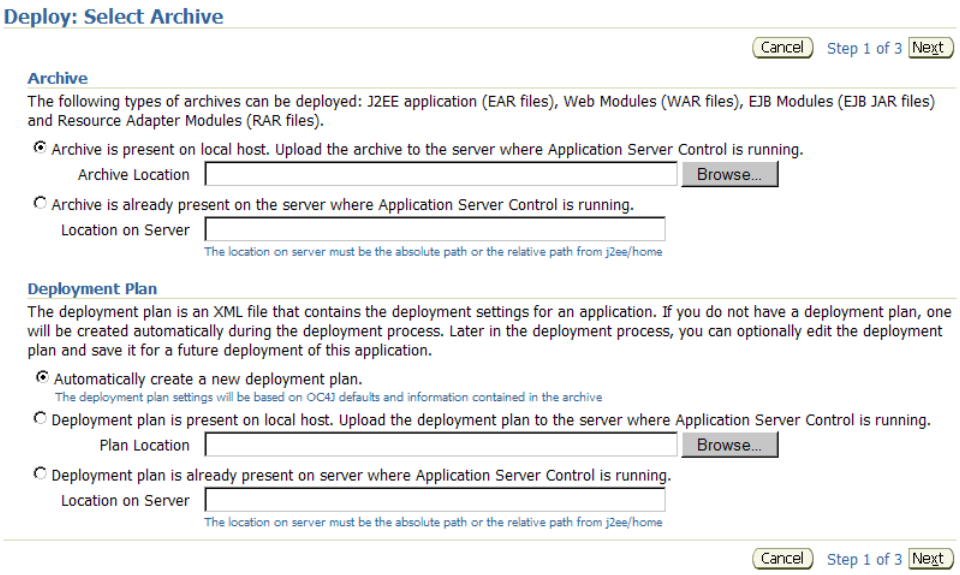
Step 6 Install the Identity Proxy

Use the following steps to install the Identity Proxy to the Oracle Application Server.

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



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



3. Select the file to be uploaded:

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

   3.2. In the **Archive Location** field, click **Browse** and navigate to the `idproxy.ear` file.

   3.3. Select the file and click **Open**.

4. Select the deployment plan for the application:

   4.1. 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.**

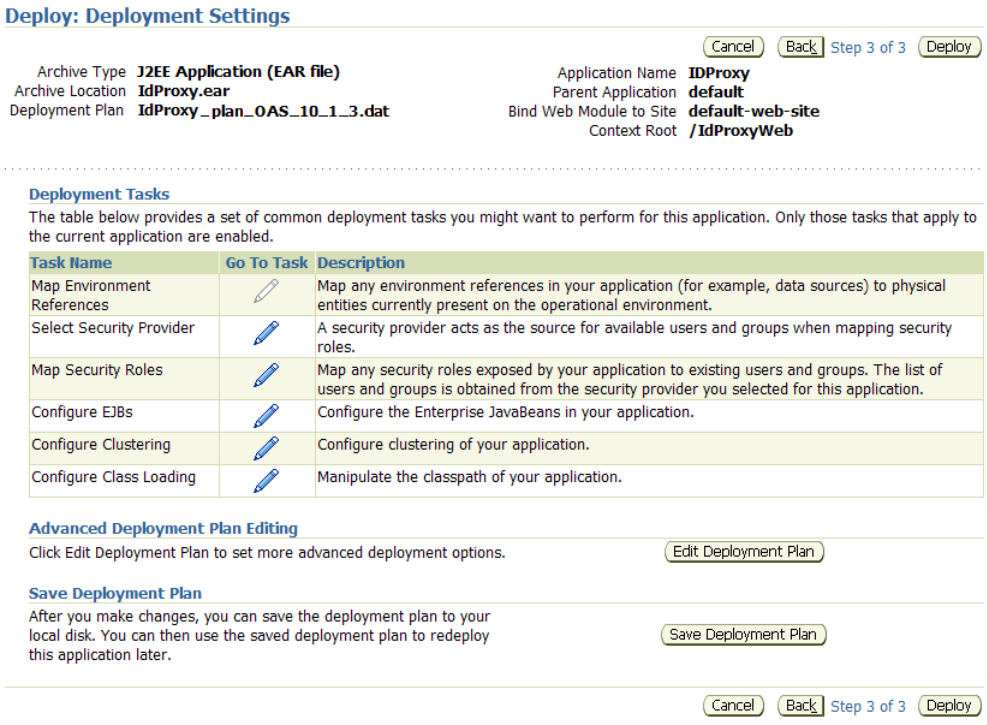
   4.2. In the **Plan Location** field, click **Browse** and navigate to the `IdProxy_plan_10_1_3.dat` file.

   4.3. Select the file and click **Open**.

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

6. Enter the name for the application (for example, *IDProxy*) in the **Application Name** field.

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



8. Click **Deploy** to accept the values and install the Identity Proxy. A deployment confirmation page is displayed.

9. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

Step 7 Configure logging

The Enterprise Identity Proxy Services application uses 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 file**. The default location is `<OAS_HOME>/j2ee/home/idp_application.log`. This location should be changed to the OC4J instance where the Enterprise Identity Proxy Services application is installed.
- **Logging level**. The default level is *INFO*, resulting in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide detailed logging, you should modify the log4j configurations.

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

1. Navigate to `<OAS_HOME>/j2ee/<OC4J instance>/applications/IdProxy/IdProxyWeb/WEB-INF/classes`.
2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
`log4j.appender.out.File`	`idp_application.log`	`../<OC4J instance>/log/ idp_application.log`
`log4j.rootCategory`	`INFO,out`	`DEBUG,out`

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

Installation on Oracle WebLogic Server 11g

`Deployables\Weblogic\Banner_IdentityProxy_full_release_Weblogic.zip` is used for installation on Oracle WebLogic Server 11g. This zip file contains an archive file named `idproxy.ear`. This file provides an implementation of the SPML Request Authority.

Recommended configuration

The Identity Proxy must be installed in an Oracle WebLogic Basic Domain Managed Server with the Banner Identity Gateway. These components must be installed in the same

Managed Server. They must not be installed in an Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual server for BEIS and other middle-tier components. This server would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server instance should consist of the default Admin Server and at least two Managed Servers:

- One Managed Server for the Banner Identity Gateway and the Identity Proxy, which must be installed together
- One Managed Server for the Identity Data Export Utilities and the SSO Manager

If a domain based on the Basic Domain template already exists for middle-tier applications, the BEIS components can be installed in two separate Managed Servers in that domain, based on the preceding recommendation.

Refer to the Oracle WebLogic Server Documentation Library for details on creating a new domain and a new Managed Server.

Installation steps

Note

The following steps are used for a manual installation of the Identity Proxy. If you prefer to use the automated installer, refer to Chapter 9, "Automated Installer".

Use the following steps to install the Identity Proxy on Oracle WebLogic Server 11g (version 10.3.2).

- Step 1, "Configure the database user and schema"
- Step 2, "Customize properties"
- Step 3, "Define the data source"
- Step 4, "Configure the security group and user"
- Step 5, "Create a JMS server"
- Step 6, "Create a JMS module"
- Step 7, "Configure a JMS queue and connection factory"
- Step 8, "Install the Identity Proxy"

Step 1 Configure the database user and schema

Use the following steps to create the database user, tables, and packages that are required by the Identity Proxy.

1. Extract the `Banner_IdentityProxy_full_release.zip` archive file.

2. Go to the directory where the file was extracted. This directory is referred to as <`IDSVC_INSTALLER_HOME`>.

   ```
   cd <IDSVC_INSTALLER_HOME>/db-scripts/users
   ```

3. Run SQL*Plus and connect as DBA.

4. Execute the `identmgr.sql` script:

   ```
   sqlplus> @identmgr
   ```

5. When prompted, enter the following information:

   - Schema name for the Enterprise Identity Proxy Services (*identmgr*)
   - Tablespace name for the Enterprise Identity Proxy Services schema (for example, *identmgr_tb*)
   - Name of the datafile with the complete path (for *example, /u01/app/oracle/ORDBMS/10.2.0/dbs/idproxy.dbf*)
   - Password for the Enterprise Identity Proxy Services schema (for example, *u_pick_it*)

6. Close SQL*Plus.

   ```
   sqlplus> exit
   ```

7. Change the directory to <`IDSVC_INSTALLER_HOME`>`/db-scripts/tables`.

   ```
   cd <IDSVC_INSTALLER_HOME>/db-scripts/tables
   ```

8. Run SQL*Plus and connect as the `identmgr` user.

9. Execute the following scripts to create the tables:

   ```
   sqlplus> @setup
   sqlplus> exit
   ```

10. Change to the directory <`IDSVC_INSTALLER_HOME`>`/db-scripts/packages.`

11. Run SQL*Plus and connect as the `identmgr` user.

12. Execute the following command to create the database packages:

```
sqlplus> @iokp_build
```

13. Exit SQL *Plus.

Step 2 Customize properties

The following properties can be customized:

- **Logging** - The Identity Proxy uses Apache's log4j to log the activities performed by the application at runtime. The `idp_application.log` file is located at the following location:

  ```
  Oracle\Middleware\user_projects\domains\<domain_name>
  ```

  where `<domain_name>` is the name of the domain where the application is installed. This location cannot be changed.

  A property in the `log4j.properties` file determines the logging level. The default logging level is *INFO*, which results in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. You can modify the logging level if you want more detailed logging.

- **Processing parameters (optional)** - The Identity Proxy is delivered with default values for processing parameters in the `spml.properties` file, which is embedded in the application. These parameters can be changed to improve performance, depending on the capabilities of the PSPs to which the Identity Proxy provisions. For most installations, changes are not necessary. For new installations, it is recommended that changes are not made and that this step is skipped. However, processing parameters can be adjusted if your installation requires performance tuning.

Use the following steps to customize logging and the processing parameters.

1. Copy `IdProxy.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 IdProxy.ear
   ```

   The extract contains a Web archive named `IdProxyWeb.war` and an archive named `IdProxyEJB.jar`.

3. Use the following steps to customize logging.

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

   3.2. Navigate to `war_home` and execute the following command:

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

   3.3. Open `war_home/WEB-INF/classes/log4j.properties`.

   3.4. In the `log4j.properties` file, edit the `log4j.rootCategory` property as follows:

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

   3.5. Save the changes.

   3.6. From `war_home` execute the following command to rebuild the Web archive file:

   ```
   jar cvf <EAR_HOME>/IdProxyWeb.war *
   ```

4. Use the following steps to customize processing parameters:

   4.1. From `<EAR_HOME>` execute the following command:

   ```
   jar xvf IdProxyEJB.jar spml.properties
   ```

   4.2. Open `spml.properties,` which is extracted under `<EAR_HOME>`.

   4.3. Edit any of the following default values to meet your requirements:

Parameter	Description	Default Value
`spml_retry_count`	Number of times sending the UDCIdentity message to a particular endpoint is retried. After this number of times, the message is marked as *FAILURE*.	*5*
`spml_batch_size`	Maximum number of UDCIdentity messages sent to a Provisioning Service Target (PST) in one task invocation.	*60*
`spml_task_interval`	Delay before starting the task for the first time (milliseconds). Only for the very first invocation.	*30000* (30 seconds)
`spml_task_delay`	Time period between repeated task executions (milliseconds).	*5000* (5 seconds)

4.4. Save the changes.

4.5. From <EAR_HOME> execute the following command to rebuild IdProxyEJB.jar:

```
jar uvf IdProxyEJB.jar spml.properties
```

4.6. After this command runs successfully, delete the spml.properties file under the <EAR_HOME> directory.

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

```
jar cvf IdProxy.ear *.war *.jar lib/* META-INF/*
```

The rebuilt IdProxy.ear is used for installation.

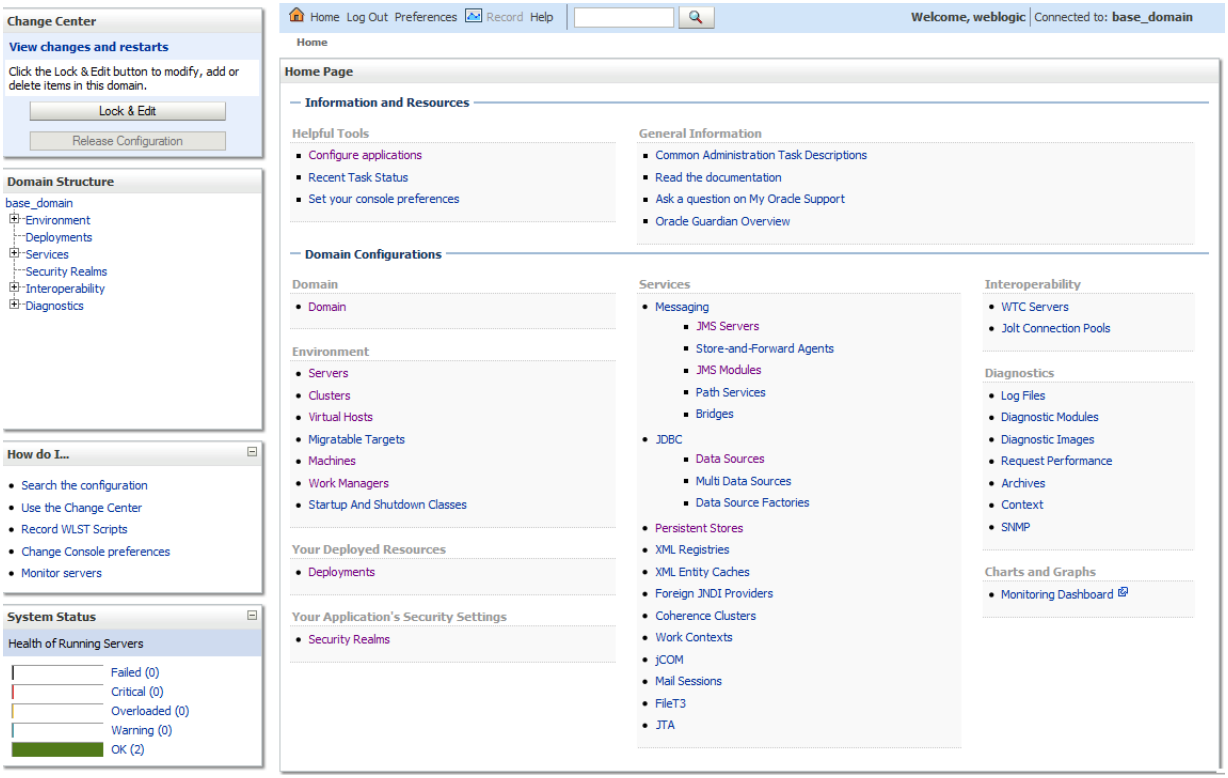
Step 3 Define the data source

A data source provides the connection properties to the Banner database. Use the following steps to define the data source that is used to access the database schema created for the Identity Proxy.

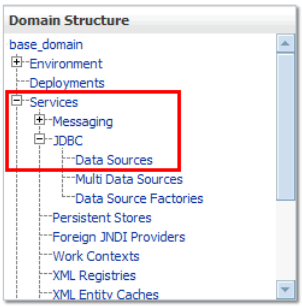
1. Connect to the Oracle WebLogic Server Administration Console:

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

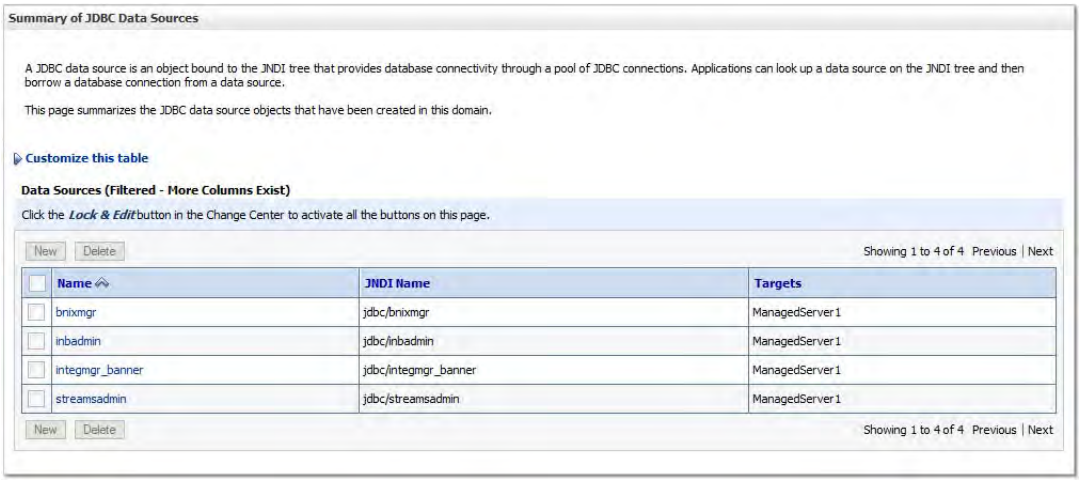
The Home Page is displayed.



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

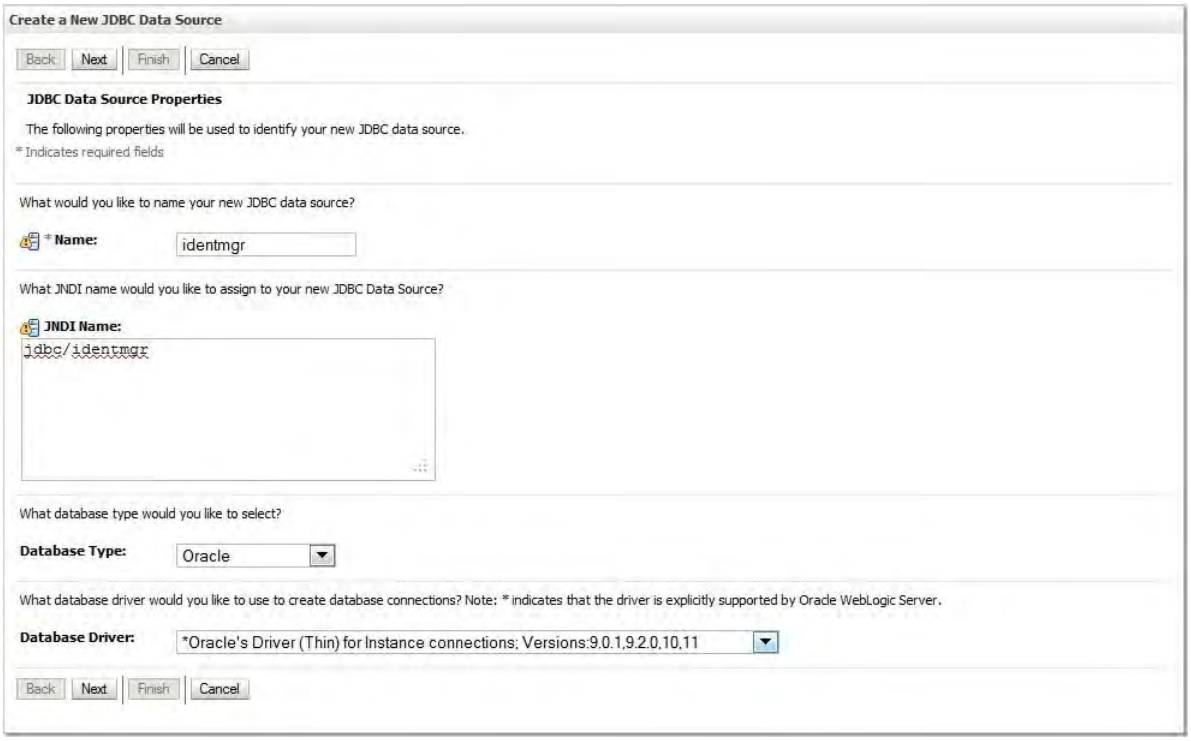


The Summary of JDBC Data Sources page is displayed.



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

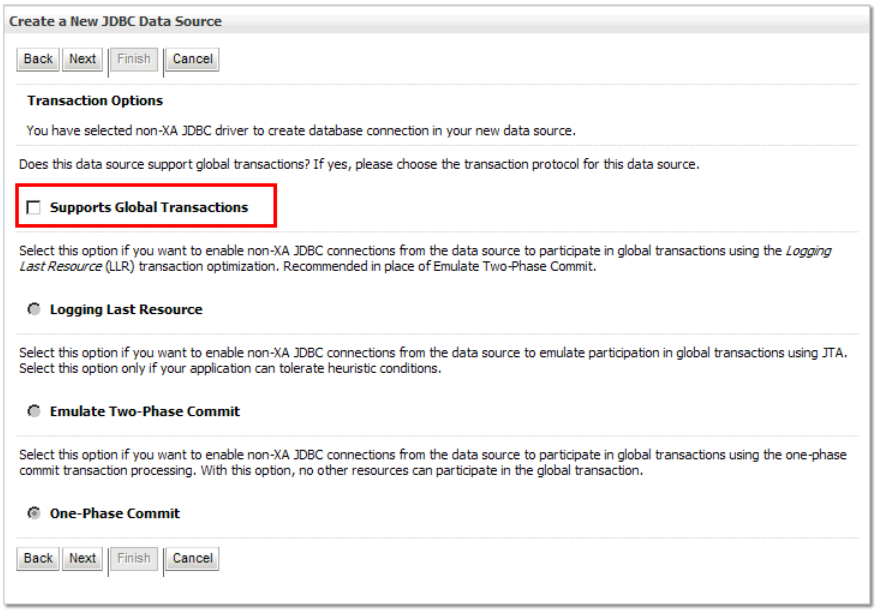
4. On the Summary of JDBC Data Sources page, click **New**. The Create a New JDBC Data Source page is displayed.



5. Enter the following data source properties:

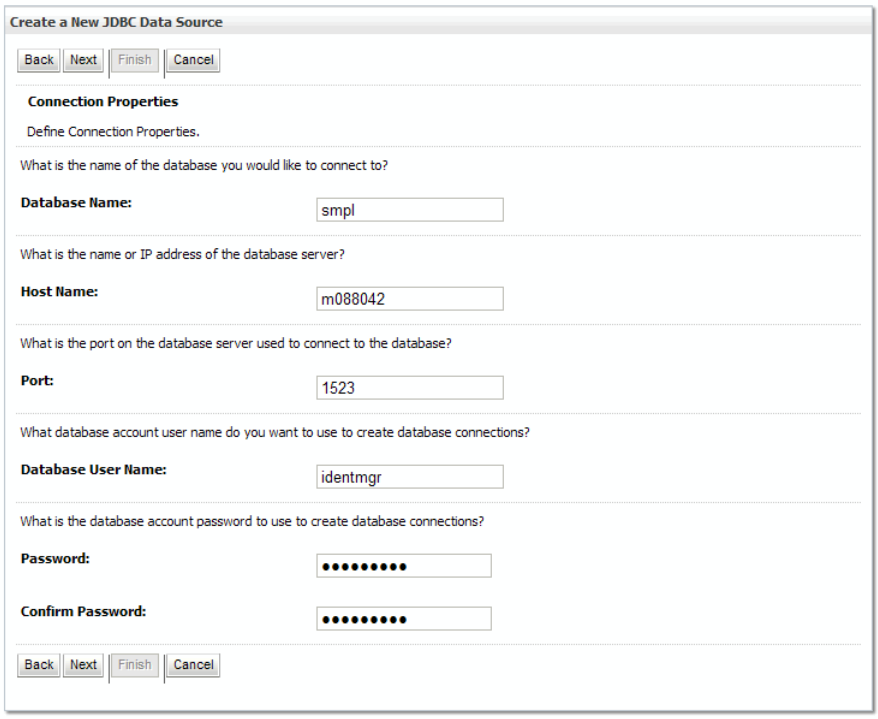
Name	*identmgr*
JNDI Name	*jdbc/identmgr*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

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



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

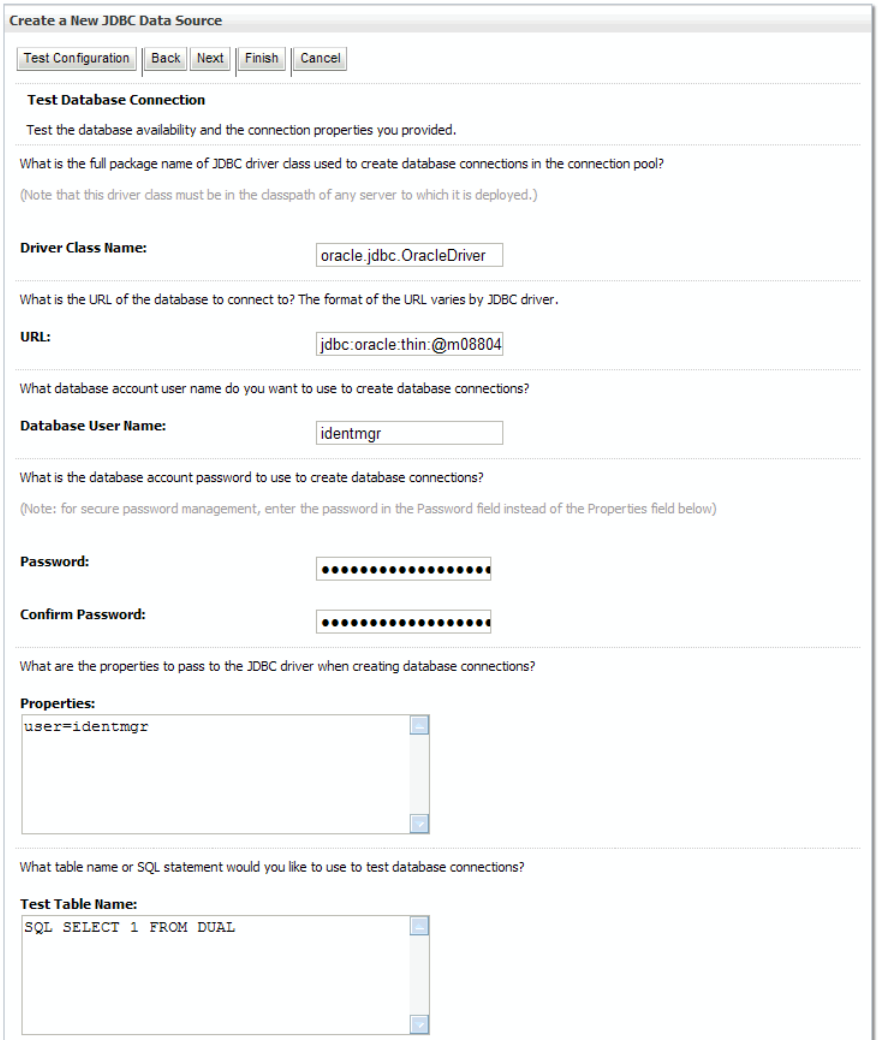
8. Click **Next**. The next page is displayed.



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	*identmgr*
Password	Password for the `identmgr` user
Confirm Password	Confirmation of the password

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



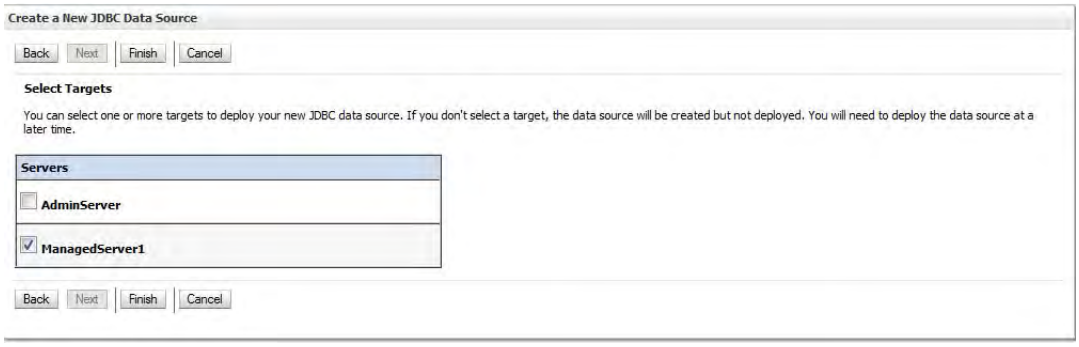
11. Verify the property values.

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.

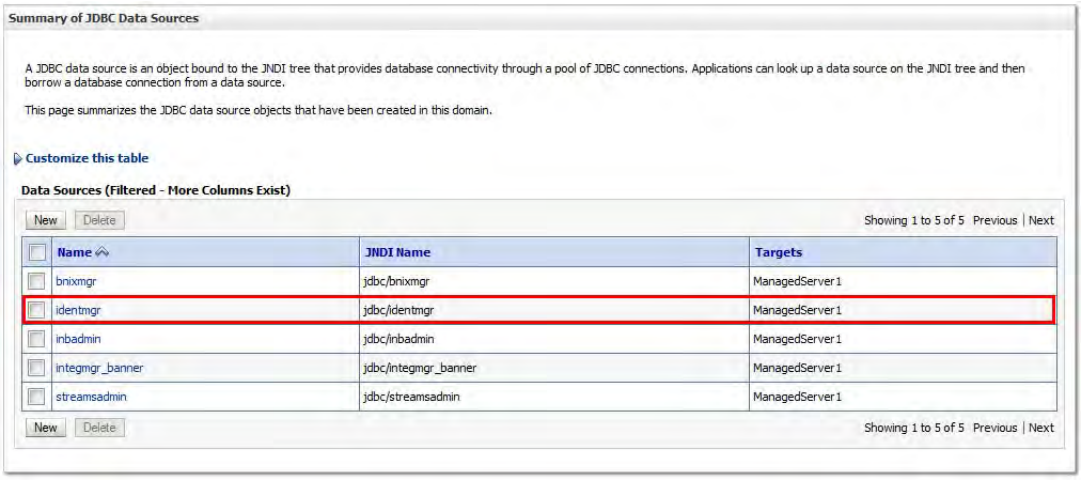
    12.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

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



14. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the Identity Proxy will be deployed.

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



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

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

Step 4 Configure the security group and user

Use the following steps to add the `idpAdminGroup` group and an administrative user to the Enterprise Identity Proxy Services application. This group and user are required for accessing the Identity Proxy administrative interface.

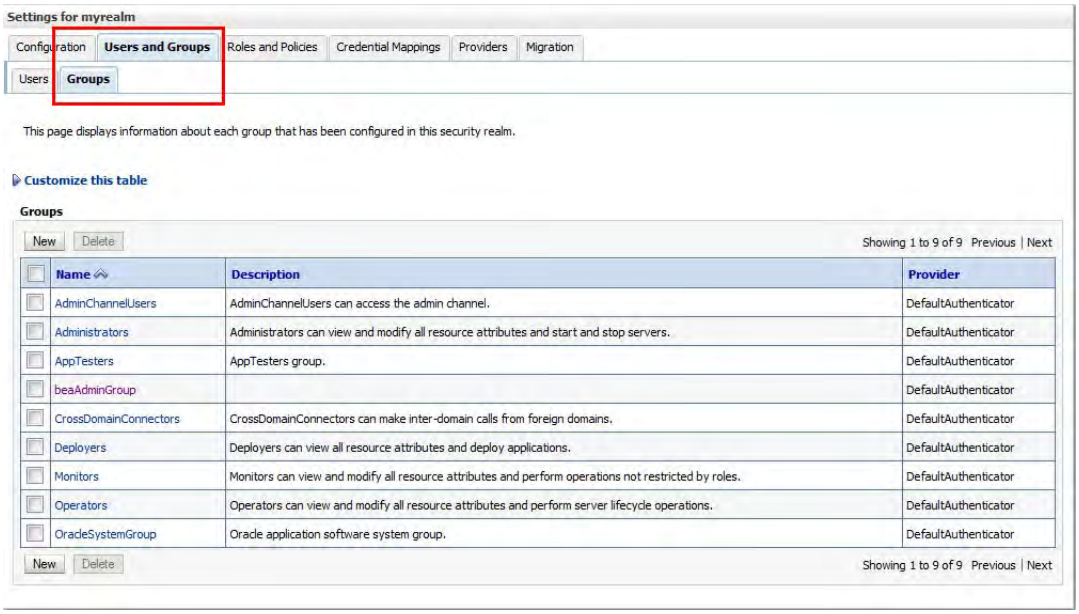
1. In the Domain Structure pane, click **Security Realms**.

The Summary of Security Realms page is displayed.

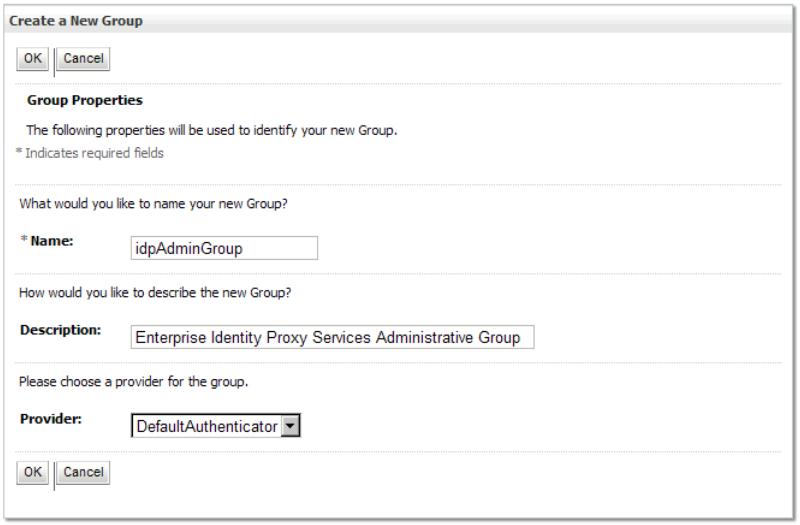
2. Click **myrealm**. The Settings for myrealm page is displayed.

3. Select the **Users and Groups** tab.

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



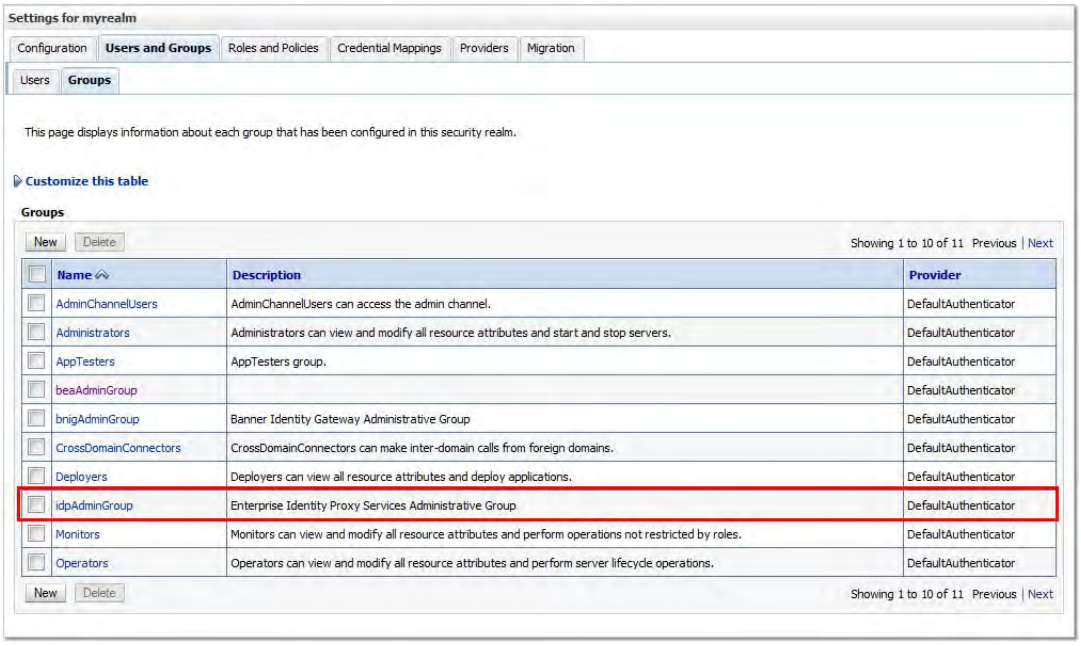
5. Click **New**. The Create a New Group page is displayed.



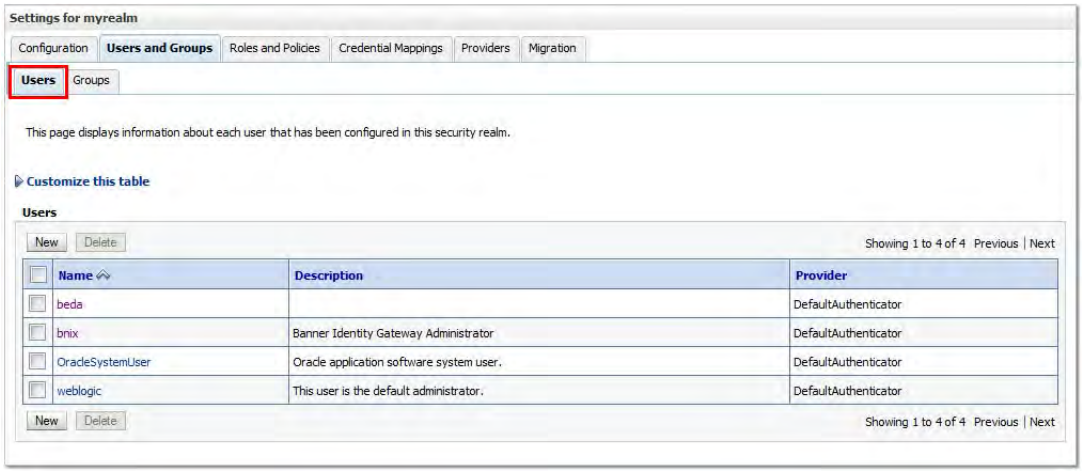
6. Enter the following information to create a group:

Name	*idpAdminGroup*
Description	*Enterprise Identity Proxy Services Administrative Group*
Provider	*DefaultAuthenticator*

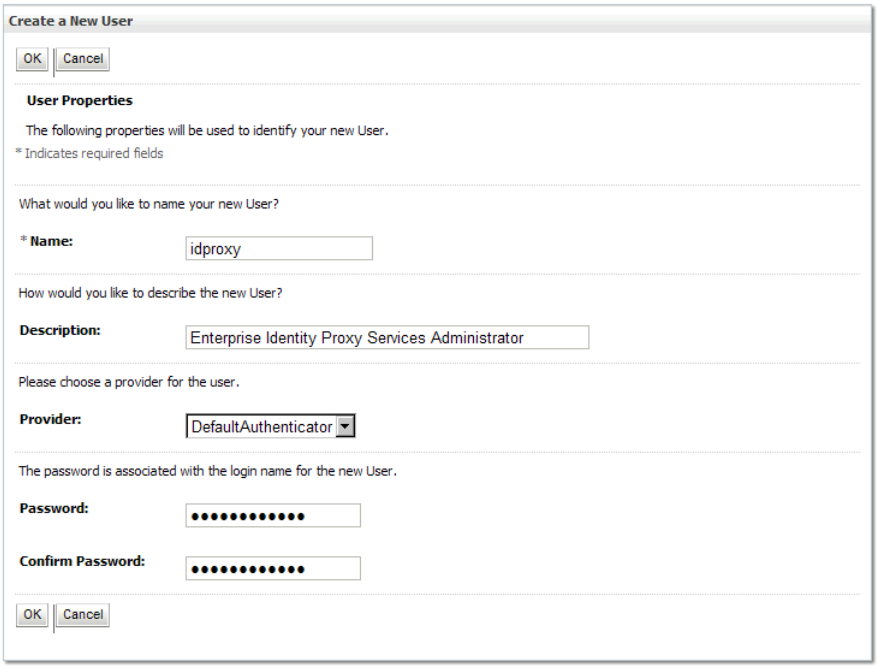
7. Click **OK**. The table of groups is redisplayed with the new group.



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



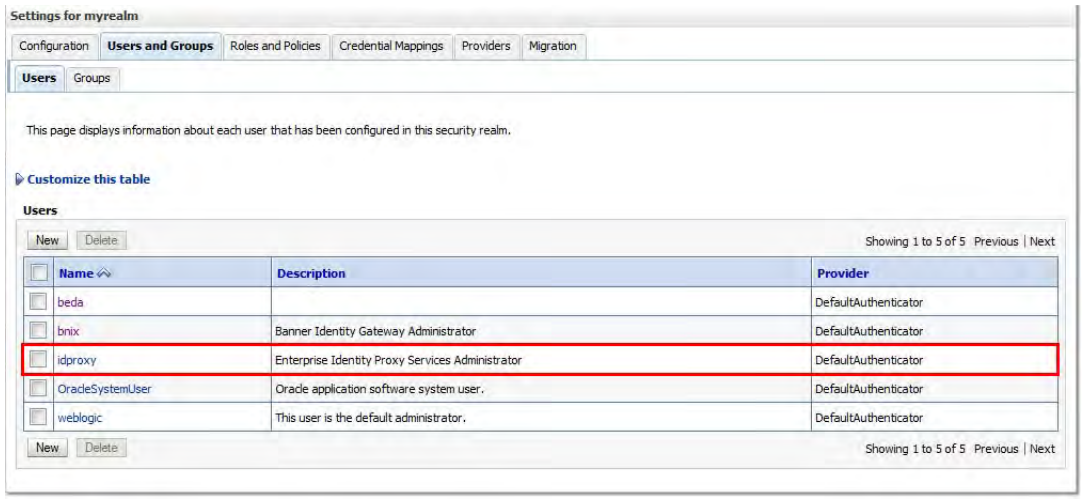
9. Click **New**. The Create a New User page is displayed.



10. Enter the following information to create a user:

Name	*idproxy* (This is an example. Enter the name of your choice.*)*
Description	*Enterprise Identity Proxy Services Administrator*
Provider	*DefaultAuthenticator*
Password	Password used to log in to the Identity Proxy administrative interface
Confirm Password	Confirmation of the password

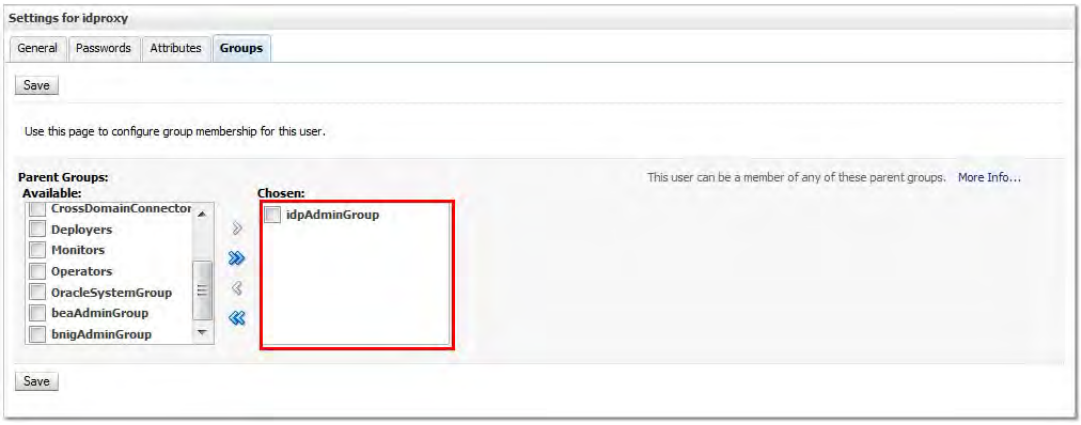
11. Click **OK**. The table of users is redisplayed with the new user.



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 *idpAdminGroup* in the **Available** list and move it to the **Chosen** list.



15. Click **Save**.

Step 5 Create a JMS server

BEIS uses Java messaging to move UDCIdentity messages among its various services. A JMS server acts as a management container for JMS resources such as connection factories, queues, and topics. All BEIS components share a JMS server.

Verify that a JMS server was created for BEIS. If a JMS server was not created, use Step 9, "Create a JMS server" on page 7-56 to create a JMS server for BEIS.

Step 6 Create a JMS module

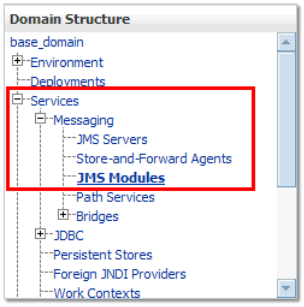
A JMS module is used to store JMS resources such as connection factories, queues, and topics. All BEIS components share a JMS module.

Verify that a JMS module was created for BEIS. If a JMS module was not created, use Step 10, "Create a JMS module" on page 7-59 to create a JMS module for BEIS.

Step 7 Configure a JMS queue and connection factory

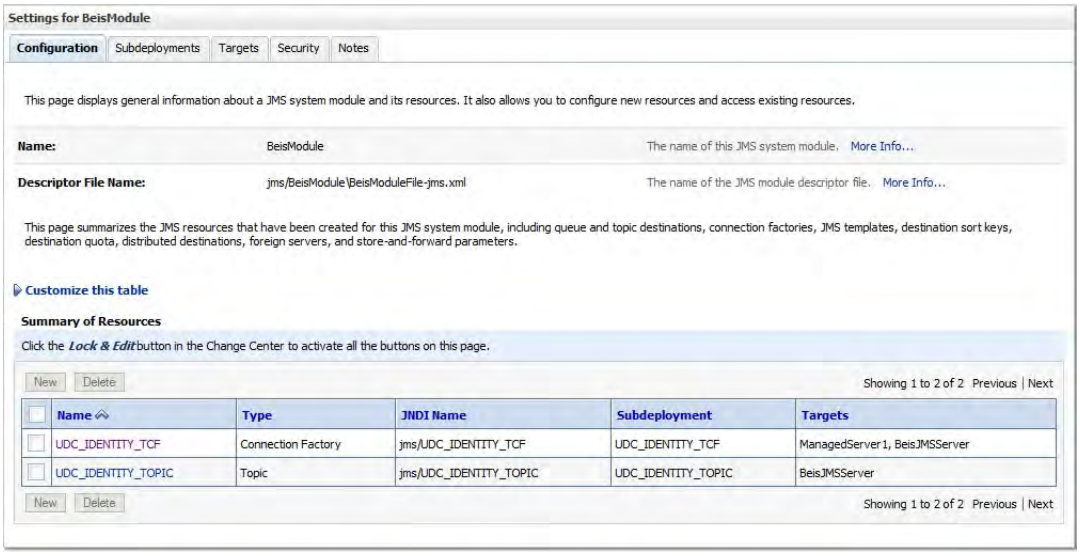
Use the following steps to add a JMS queue and connection factory for the Identity Proxy.

1. In the Domain Structure pane, expand and click **Services -> Messaging -> JMS Modules**.



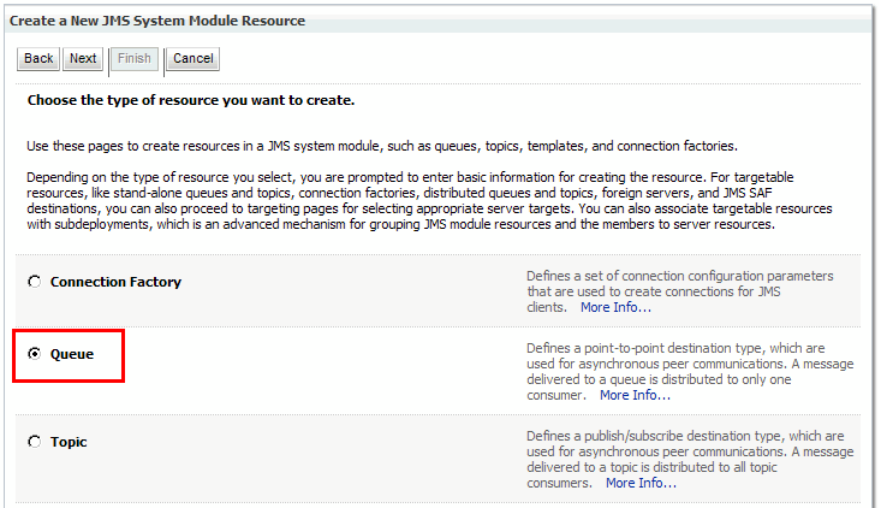
The JMS Modules page is displayed.

2. Click the name of the module that was created for BEIS. The Settings page for the module is displayed.



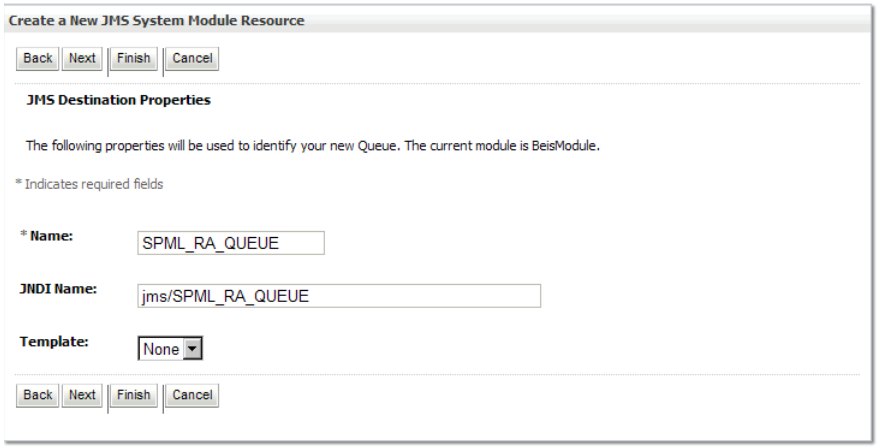
3. In the Change Center pane, click **Lock & Edit**.

4. On the Settings page, click **New**. The Create a New JMS System Module Resource page is displayed.



5. Select **Queue**.

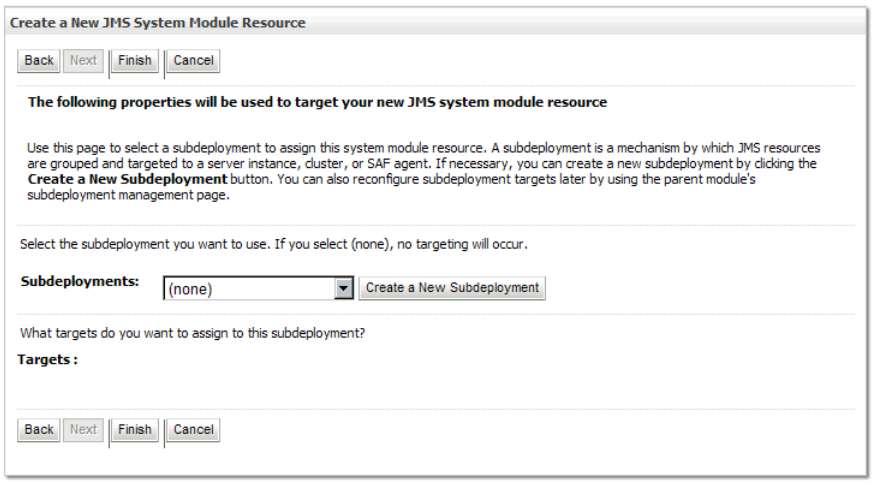
6. Click **Next**. The next page is displayed.



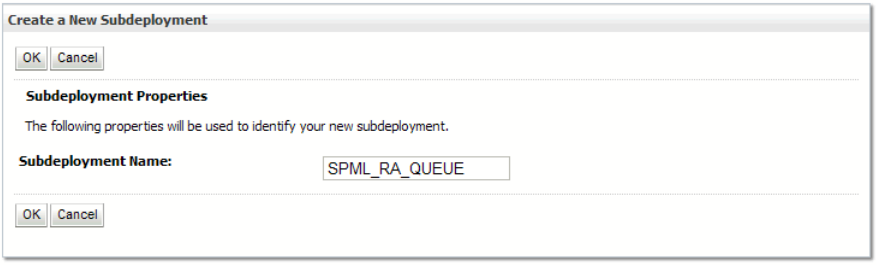
7. Enter the following information to create a queue:

Name	*SPML_RA_QUEUE*
JNDI Name	*jms/SPML_RA_QUEUE*

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

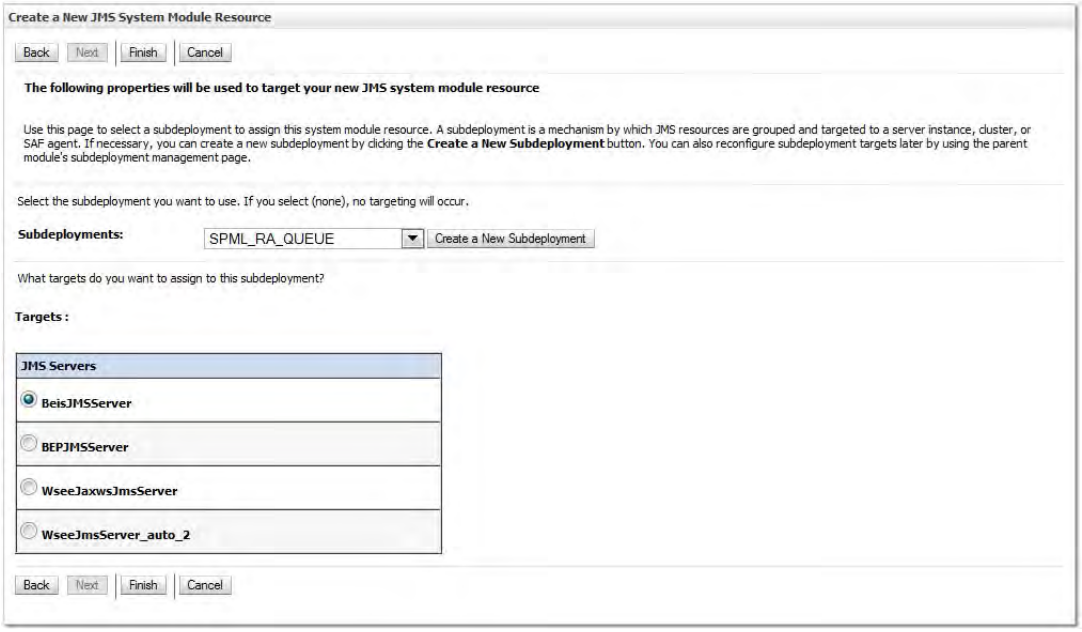


9. Click **Create a New Subdeployment**. The Create a New Subdeployment page is displayed.



10. Enter the name of the subdeployment (for example, *SPML_RA_QUEUE*) in the **Subdeployment Name** field.

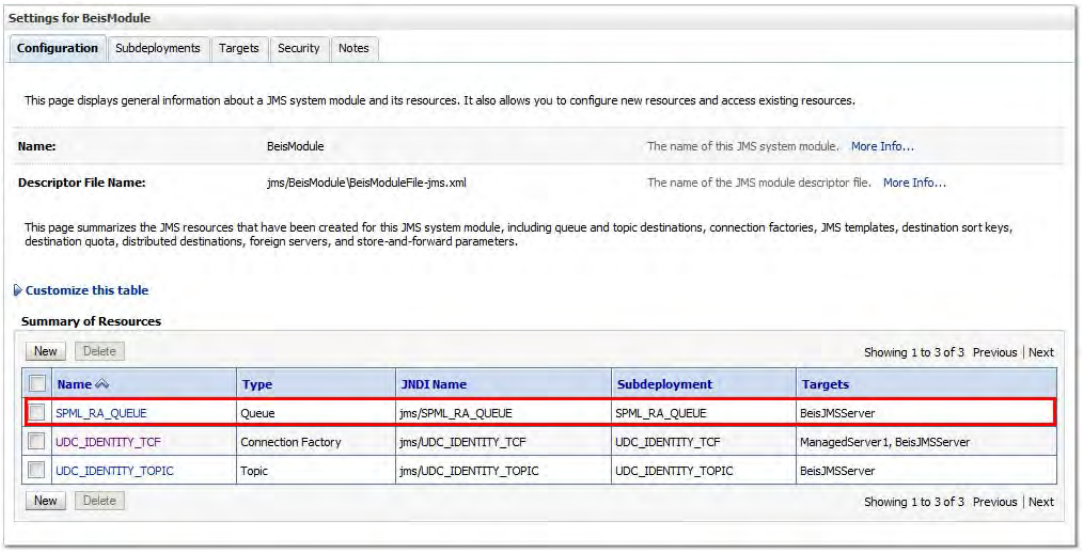
11. Click **OK**. The Create a New JMS System Module Resource page is redisplayed.



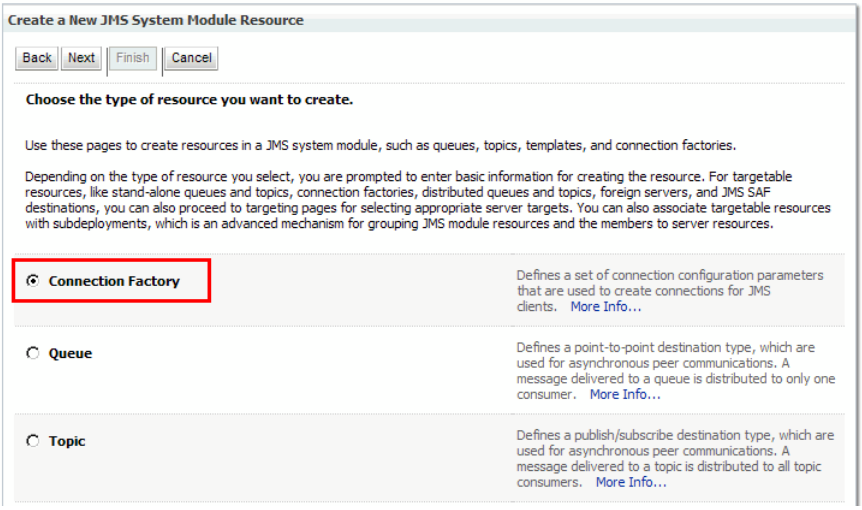
12. Select *SPML_RA_QUEUE* from the **Subdeployments** drop-down list.

13. Select a target server(s) for the topic.

14. Click **Finish**. The Settings page is redisplayed with the new queue.

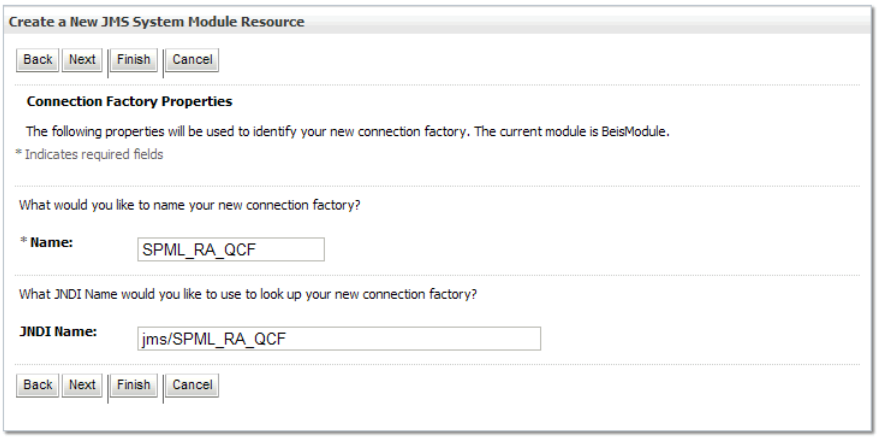


15. Click **New**. The Create a New JMS System Module Resource page is displayed.



16. Select **Connection Factory**.

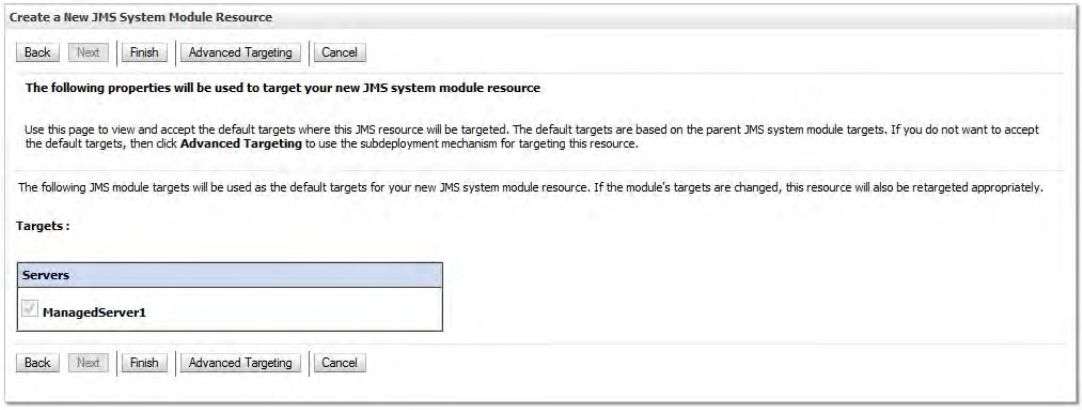
17. Click **Next**. The next page is displayed.



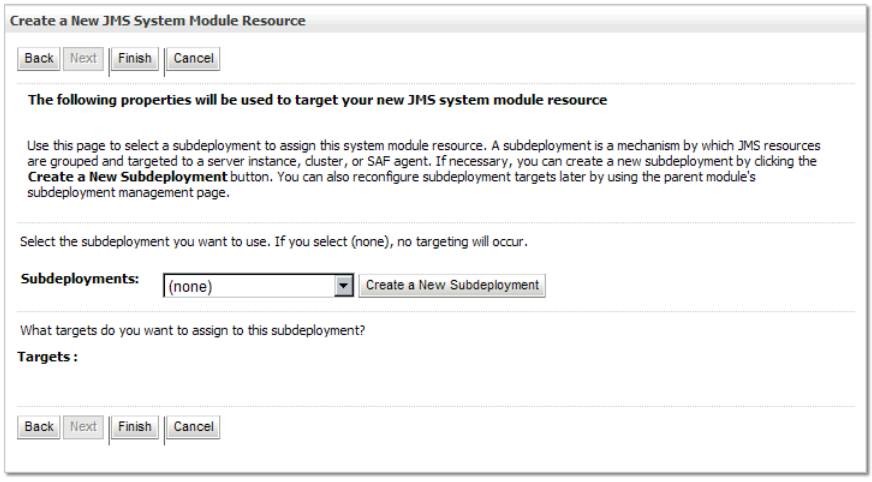
18. Enter the following information to create a connection factory:

Name	*SPML_RA_QCF*
JNDI Name	*jms/SPML_RA_QCF*

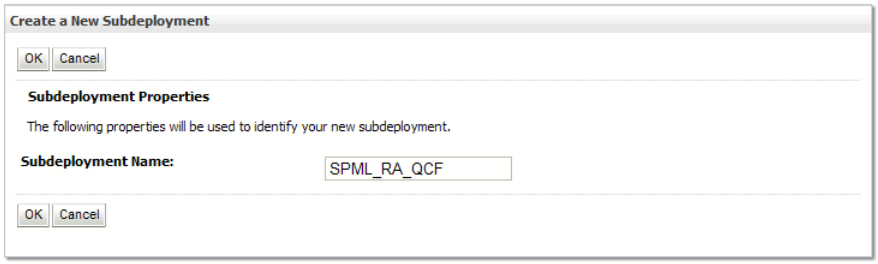
19. Click **Next**. The next page is displayed.



20. Click **Advanced Targeting**. The next page is displayed.

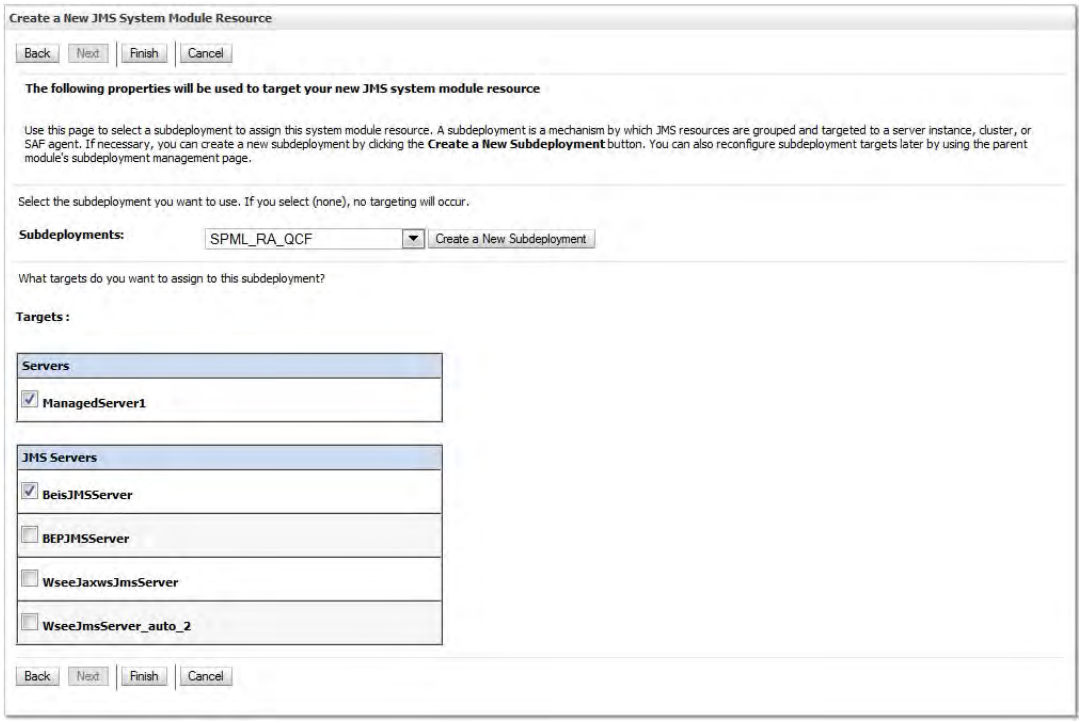


21. Click **Create a New Subdeployment**. The Create a New Subdeployment page is displayed.



22. Enter the name of the subdeployment (for example, *SPML_RA_QCF*) in the **Subdeployment Name** field.

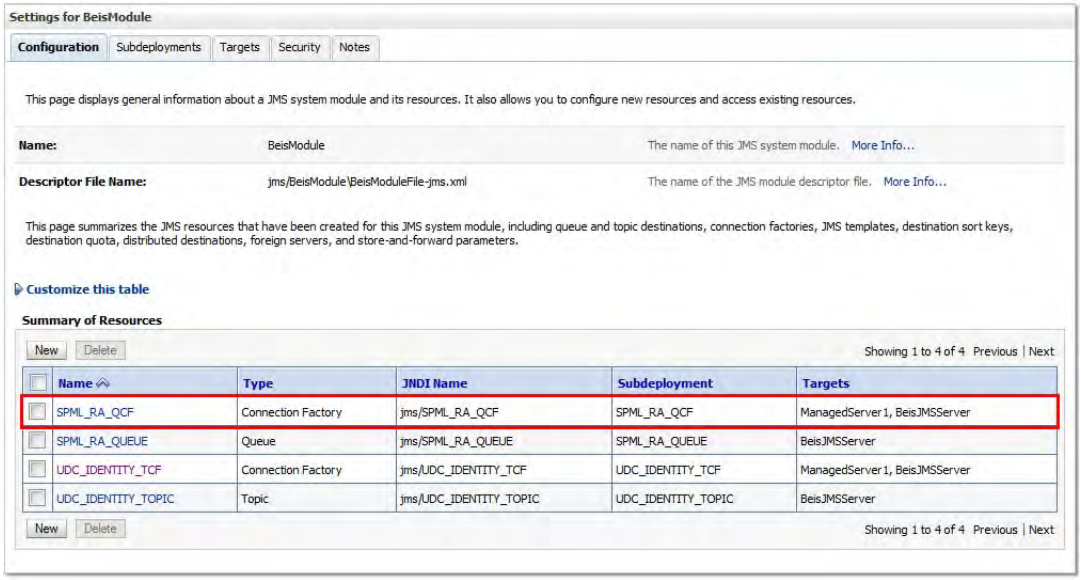
23. Click **OK**. The Create a New JMS System Module Resource page is redisplayed.



24. Select *SPML_RA_QCF* from the **Subdeployments** drop-down list.

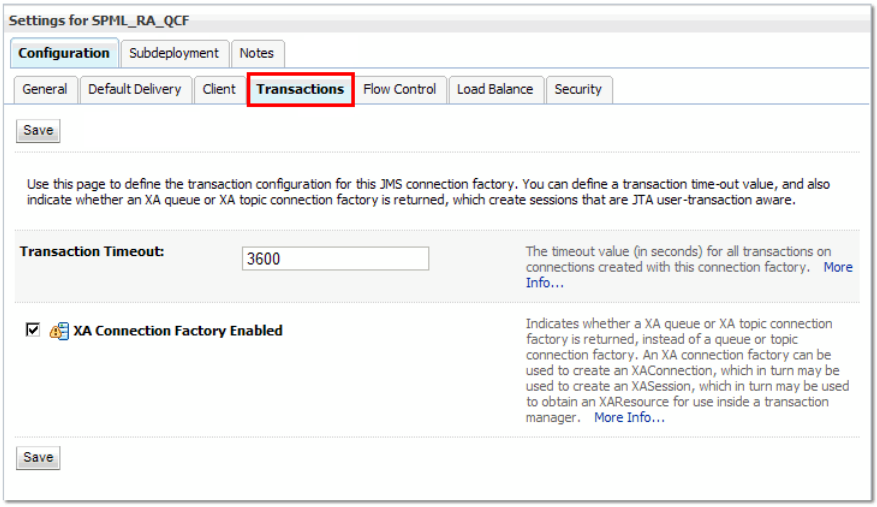
25. Select a target server and a JMS server for the connection factory.

26. Click **Finish**. The Settings page is redisplayed with the new connection factory.



27. Click the name of the connection factory that you just created. The Settings page for the connection factory is displayed.

28. Select the **Transactions** tab.



29. Select **XA Connection Factory Enabled**.

30. Click **Save**.

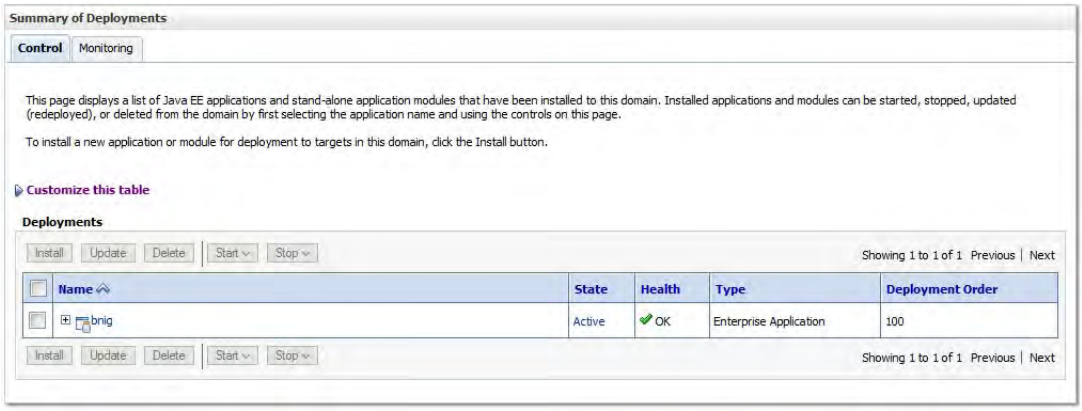
31. In the Change Center pane, click **Activate Changes**.

Step 8 Install the Identity Proxy

Use the following steps to install the Enterprise Identity Proxy Services application to the Oracle WebLogic Server.

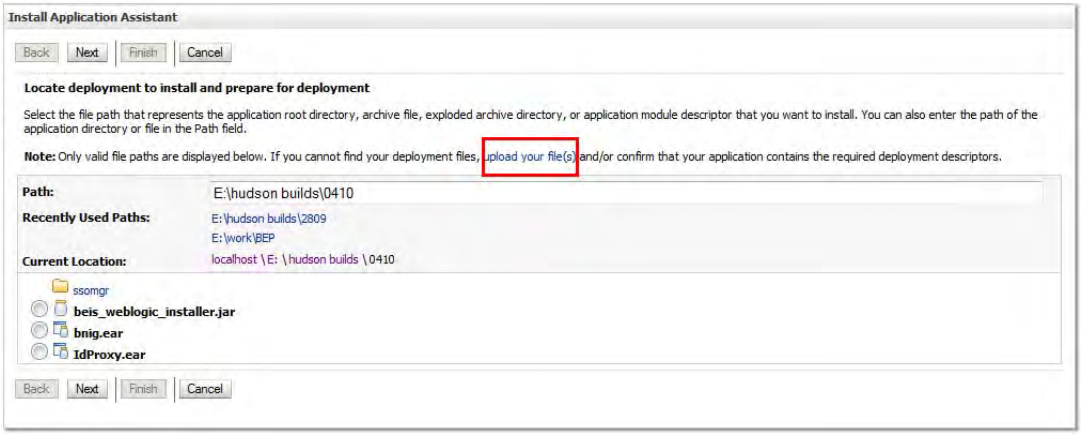
1. In the Domain Structure pane, click **Deployments**.

The Summary of Deployments page is displayed.

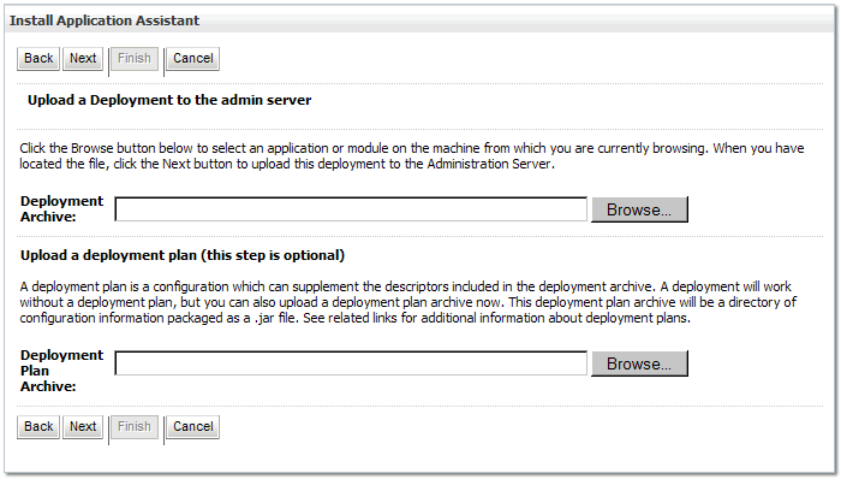


2. In the Change Center pane, click **Lock & Edit**.

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



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

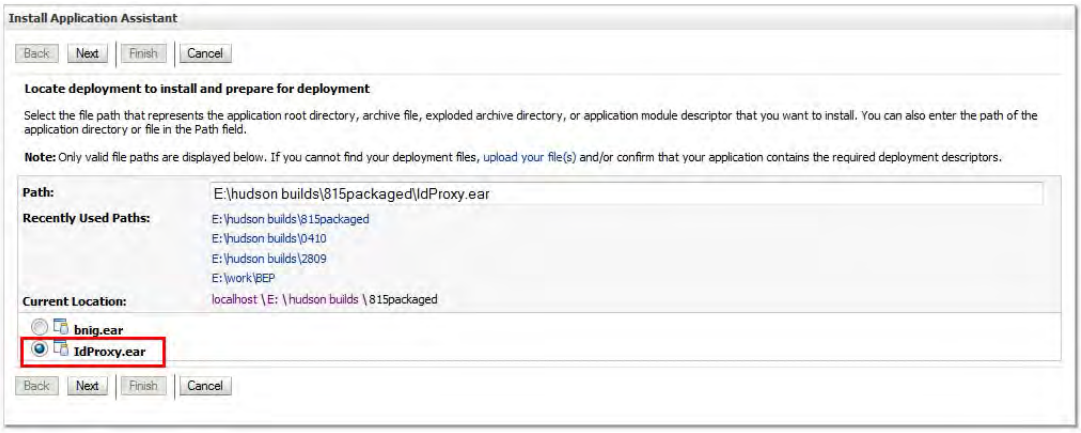


5. Select the file to be uploaded:

   5.1. In the **Deployment Archive** field, click **Browse** and navigate to the `IdProxy.ear` file.

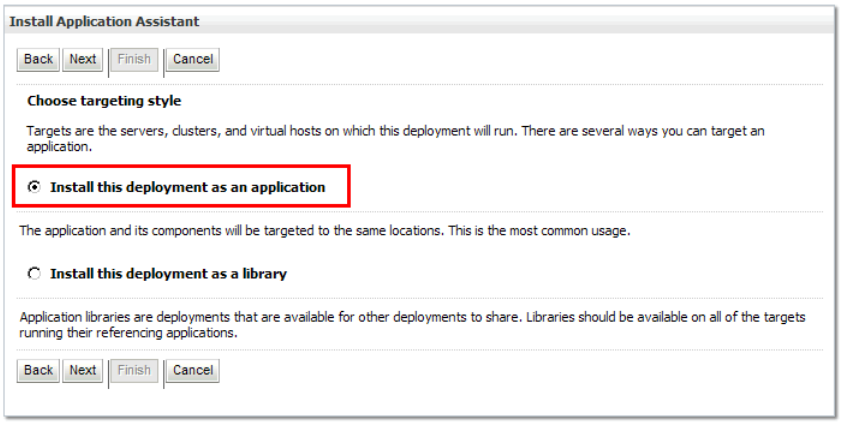
   5.2. Select the file and click **Open**.

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



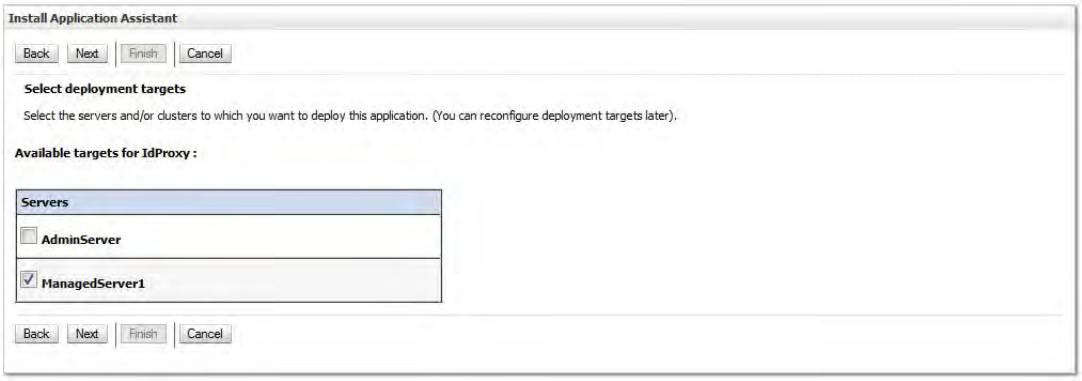
7. Select the `IdProxy.ear` file from the list.

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



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

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

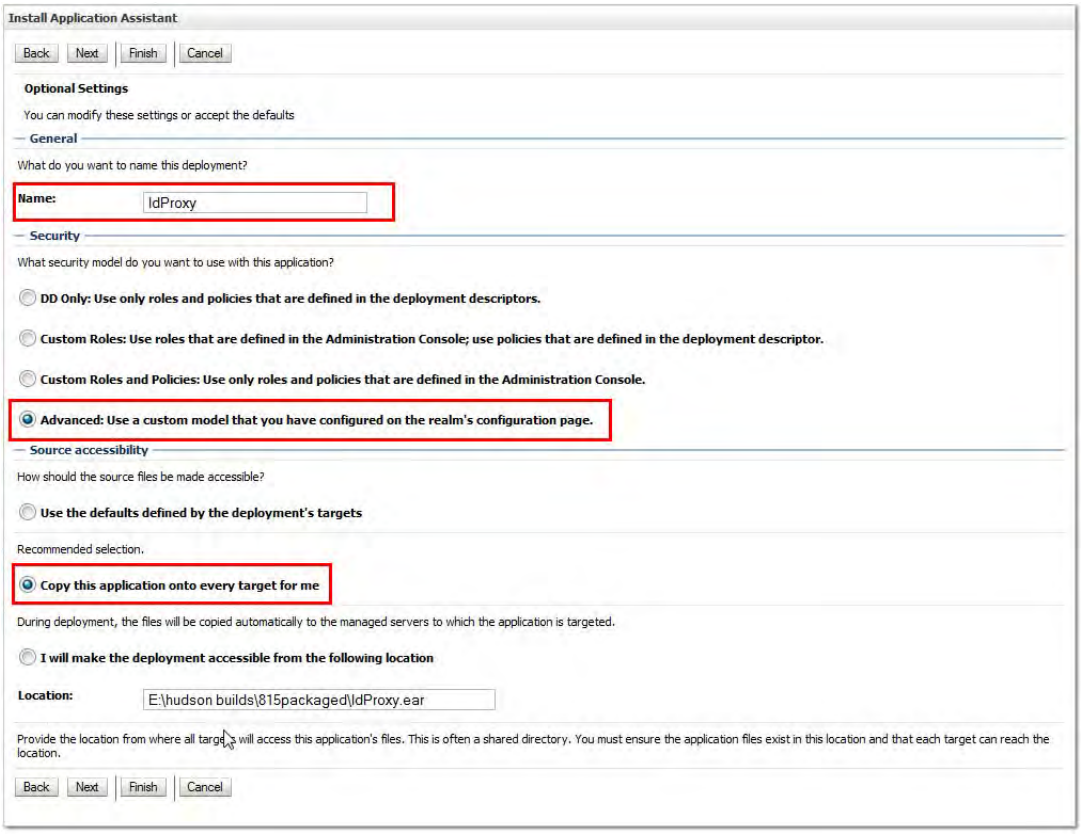


11. Select the server where the application should be deployed. (The application can be installed on an existing server.)

Note

SunGard Higher Education recommends deploying applications to a WebLogic Managed Server and not to the Administration Server. If you do not see the preceding page, you should check your WebLogic server configuration to ensure that a Managed Server is available for deployment of applications. If a Managed Server is not available, the application will be deployed to the Administration Server, which is not a recommended configuration. For more information, consult the Oracle WebLogic Server Documentation Library. ■

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

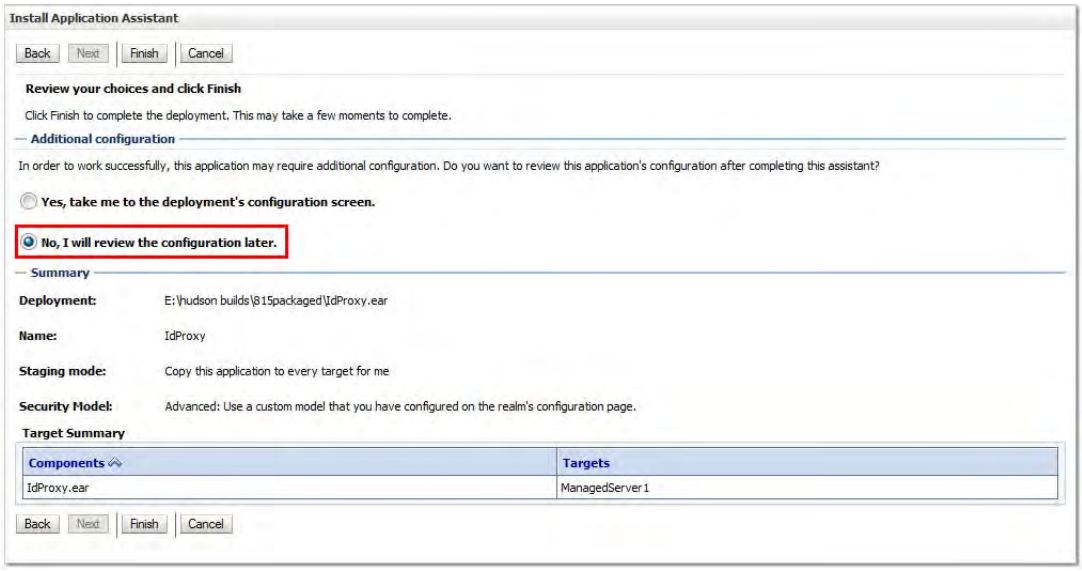


13. Enter a name for the application (for example, *IdProxy*) in the **Name** field.

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.

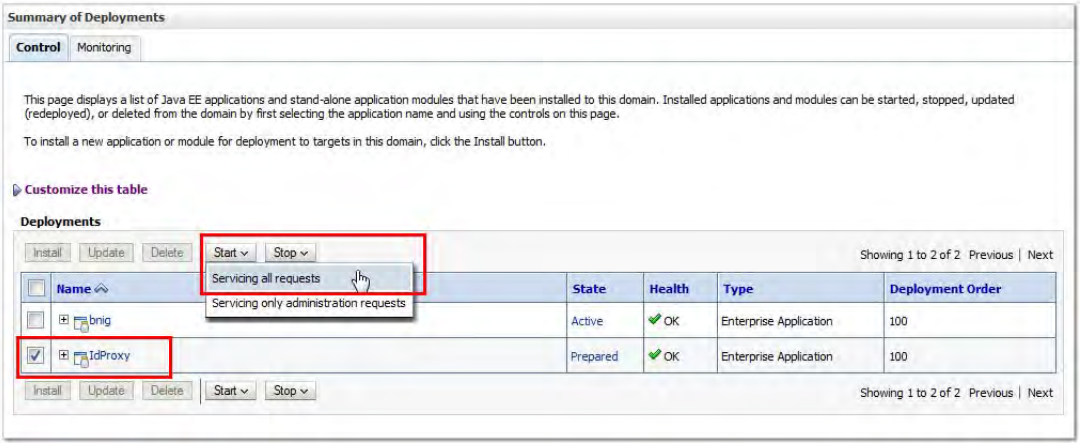


17. Select **No, I will review the configuration later**.

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

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

20. Start the newly deployed application as follows:



20.1. Select the newly deployed application.

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

20.3. Click **Yes**.

Configuration of PSPs

The Enterprise Identity Proxy Services application acts as an SPML Request Authority, making provisioning requests to configured Provisioning Service Providers (PSPs). You must configure the location of each PSP that should receive the UDCIdentity message whenever a person record in Banner is created or modified.

Note

Luminis® Platform 5.x does not provide an SPML 2.0 endpoint, so it cannot be established as a PSP in Enterprise Identity Proxy Services. The Identity Proxy application must still be installed to process messages from the UDCIdentity Topic. Banner Integration for eLearning must be used to provision user accounts in Luminis Platform 5.x.

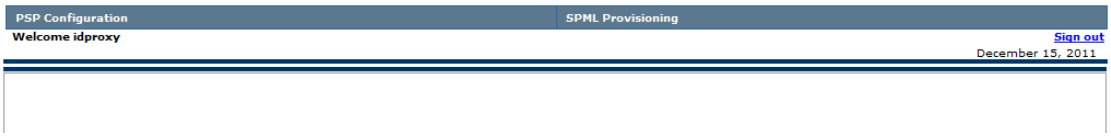
Add or update a PSP configuration

Use the following steps to configure PSPs for the SPML Request Authority.

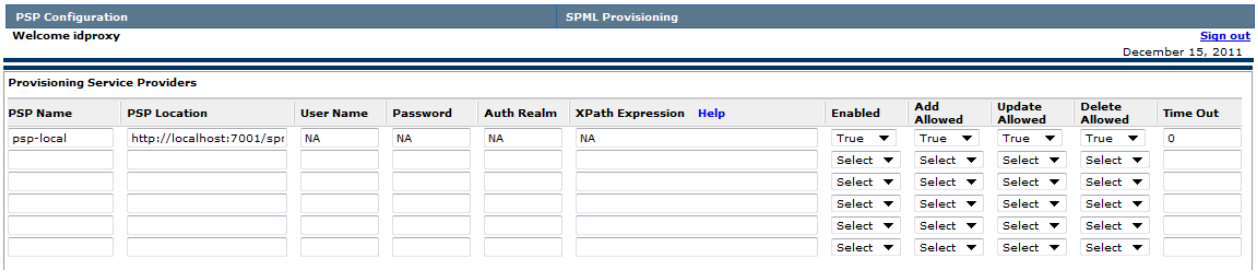
1. Connect to the Identity Proxy:

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

2. Log in with the user name and password that were mapped to the `idpadmin` role (OAS) or the `idpAdminGroup` group (Oracle WebLogic). The following page is displayed:



3. Select PSP Configuration > Add/Update PSP from the menu bar. The Provisioning Service Providers page is displayed.



4. Enter the following information for each PSP:

PSP Name — Name of the Provisioning Service Provider. The name should be unique, descriptive, and not assigned to an existing PSP. Avoid names such as *default*.

PSP Location — URL of the Provisioning Service Provider. Location of the Web service that provides PSP functionality.

Note: Basic authorization needs three pieces of information: user name, password, and realm.

User Name — User name sent to the application for HTTP basic authentication. Enter *NA* if not received. Do not leave blank.

Password — Password sent to the PSP. Enter *NA* if not received. Do not leave blank.

Auth Realm Authentication realm to which the user name/password combination applies. Enter *NA* if not received. Do not leave blank.

Note: If provisioning to Luminis® Platform 4.x, enter *Luminis Webservices*.

XPath Expression XPath expression used to filter messages. The XPath expression must be matched for the message to be sent to the PSP.

Enter *NA* if no filtering of messages is required. Do not leave blank.

All attribute names should be preceded with the prefix “prx:” to identify the namespace.

Any syntax error in the XPath expression might prevent the message from being posted to the PSP.

Examples follow on page 8-56.

Enabled *True*

Add Allowed *True*

Update Allowed *True*

Delete Allowed *True*

Time Out Time (in seconds) when a request times out. Enter *0* (zero) if a request never times out. Maximum value is *300*.

5. Click **Add/Update**.

Note

If you duplicate a PSP name, the duplicate is displayed in a Duplicate Provisioning Target section of the page when you click **Add /Update**. You must correct the name and click **Add /Update** again to save the changes.

Example XPath expressions

Example 1

The following XPath expression sends only those messages that have an InstitutionRoles element that contains the role *BASICPERSON*.

```
/prx:UDCIdentity/prx:InstitutionRoles/
prx:institutionrole[prx:role='BASICPERSON']
```

Example 2

The following XPath expression sends only those messages that have an InstitutionRoles element that contains the role *STUDENT*.

```
/prx:UDCIdentity/prx:InstitutionRoles/
prx:institutionrole[prx:role='STUDENT']
```

Example 3

The following XPath expression sends messages based on the Extension attributes. Only messages that have a *CREDENTIAL* attribute or a *SPRIDEN_PIDM* are sent.

```
/prx:UDCIdentity/prx:Extension/
prx:Attribute[prx:name='CREDENTIAL']
/prx:UDCIdentity/prx:Extension/
prx:Attribute[prx:name='SPRIDEN_PIDM']
```

Delete a PSP configuration

Use the following steps to delete a Provisioning Service Provider (PSP) configuration.

1. Select PSP Configuration > View PSP from the menu bar on the Identity Proxy administrative interface. The Provisioning Service Providers page is displayed.
2. Select the PSP configuration to be deleted.
3. Click **Delete**.

Administration

You can use the Enterprise Identity Proxy Services administrative interface to perform the following tasks:

- "View message details"
- "Delete a message that is creating an error"
- "Search for messages"

The administrative interface is accessed at the following URL:

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

Log in with the user name and password that were mapped, during installation, to the `idpadmin` role (OAS) or the `idpAdminGroup` group (Oracle WebLogic).

View message details

Use the following steps to view details for SPML messages that are being processed by BEIS.

1. Select SPML Provisioning > View Messages from the menu bar on the Identity Proxy administrative interface. The Message Details page is displayed.

2. Select a message.

3. Click **View Details.**

   You can view information such as status, created date, comments, UDC document, and SPML response. You can determine whether the message passed successfully and what action was taken on it.

   The status is one of the following values:

Status	Description
PENDING	The Identity Proxy is processing the SPML message. The message is not yet posted to the PSP, or it is being retried (up to five tries) due to a failed attempt.
SUCCESS	The SPML message was posted successfully to the PSP.
FAILURE	The SPML message failed after five tries.
REJECTED	The Identity Proxy rejected the transmission of the SPML message for one of the following reasons: • The PSP is disabled. • The PSP operation level is disabled (Add allowed, Update allowed, or Delete allowed is *false*). • The XPath expression was not matched.

4. If the message was unsuccessful, click **View Errors**. The error details are captured and stored separately. You are shown the Resource Errors and what caused them.

Delete a message that is creating an error

Use the following steps to delete a SPML message that is creating an error.

1. Select SPML Provisioning > View Messages from the menu bar on the Identity Proxy administrative interface. The Message Details page is displayed.

2. Select the message.

3. Click **Delete**.

Search for messages

Use the following steps to search for SPML messages that are being processed by BEIS.

1. Select SPML Provisioning > View Messages from the menu bar on the Identity Proxy administrative interface. The Message Details page is displayed.

2. Enter your search criteria in the top part of the page. You can search for messages based on the following search criteria:

   - From date and to date
   - Start time and end time
   - PSP name
   - Message status (for example, failed messages)

3. Click **Query**. The page displays messages that meet the search criteria.

9 Automated Installer

The Banner® Enterprise Identity Services (BEIS) Installer can be used to install the Banner Identity Gateway and the Enterprise Identity Proxy Services in one automated process. The installer can be used for new installations only. The installer cannot be used for upgrades. The installer cannot be used for other BEIS components.

The installer is packaged in two archive files:

- `Deployables\OC4J\beis_oc4j_installer.jar` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\beis_weblogic_installer.jar` is used for installation on Oracle WebLogic Server 11g.

This chapter gives instructions for using the installer to install the Gateway and Identity Proxy on both servers.

Note

If you prefer to install the components manually, use the instructions in Chapter 7, "Banner Identity Gateway" and Chapter 8, "Enterprise Identity Proxy Services". You must install the components manually if you want to change the processing properties for the Identity Proxy.

Prerequisites

The following prerequisites must be met before you use the installer:

- Java 1.6 must be installed, and environment variable `JAVA_HOME` must be set.
- All components from previous BEIS installations must be removed. This includes database connection pools, data sources, JMS destinations, connection factories, and configured security users.
- The `bnixmgr` and `identmgr` schemas and the corresponding tablespaces and data files must be dropped in the database that will be used for the BEIS installation.
- An Oracle Streams environment for BEIS is required.
- Banner must be configured to provide data for account provisioning. (Refer to Chapter 5, "Banner Configuration".)

Modes for using the installer

There are two modes for using the installer:

- When the utility is run on a server with a windowing environment, a graphical user interface (GUI) is presented that groups related options on pages.
- When the utility is run on a server without a windowing environment, each option is displayed in command-line mode with a default value. You can accept the default value or enter another value.

The instructions in this chapter are based on using the GUI. The options displayed on GUI pages are identical to the options displayed in command line mode.

Installation on Oracle Application Server

`Deployables\OC4J\beis_oc4j_installer.jar` is used for installation on Oracle Application Server 10.1.3.4/5.

The Gateway and Identity Proxy can be installed on an existing Oracle Application Server. They must be installed together in the same OC4J instance. BEIS applications should be deployed separately from other applications so they can be managed independently.

Use the following steps to use the automated installer to install the Gateway and the Identity Proxy on Oracle Application Server 10.1.3.4/5.

- Step 1, “Note the deployer URL”
- Step 2, “Configure and install the applications”
- Step 3, “Configure security roles and users”
- Step 4, “Configure logging”

Step 1 Note the deployer URL

Note the deployer URL. The configuration of the deployer URL is different for SOA Suite 10 g (10.1.3.4/5) and OC4J standalone servers:

Server	Deployer URL

SOA Suite 10g	`deployer:oc4j:<ormis:>opmn://<opmn host>:<opmn port>/<oc4j instance name>`
	Example:
	`deployer:oc4j:opmn://m038071:6004/Ant_Installer_Test`
	Port 6004 is the opmn request port.

Standalone OC4J server	`deployer:oc4j:localhost:23791` Port 23791 is the rmi port.

Step 2 Configure and install the applications

Use the following steps to install the Gateway and Identity Proxy with the automated installer.

1. Open a command prompt and navigate to the directory where the installer jar file is located.

2. Choose one of the following:

   2.1. If you are using a windowing environment, run the following command:

   ```
   E:\AntInstaller>java -jar beis_oc4j_installer.jar
   ```

   The installer is extracted, and a splash screen is displayed.

2.2. If you are using a non-windowing environment, run the following command:

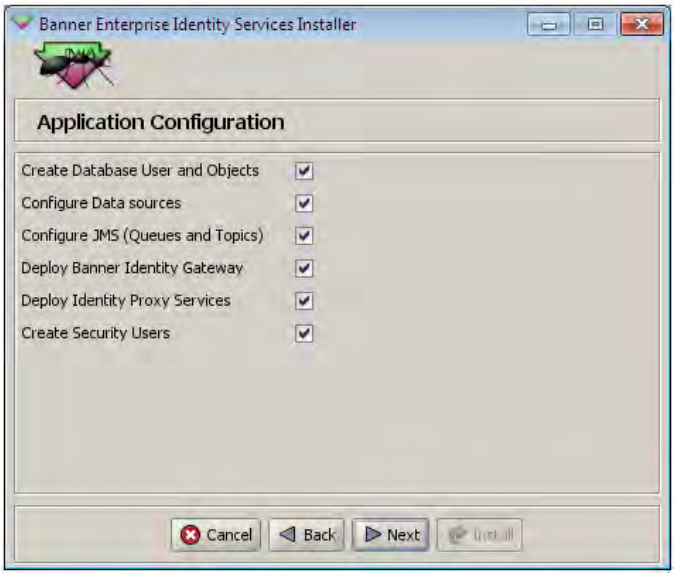
```
[oracle@octopus Ant_Installer_Test]$ java -jar
beis_oc4j_installer.jar
```

The self extractor is loaded. If the environment has graphics, a splash screen is displayed. Otherwise, a command line interface is displayed.

Note

The remaining instructions are based on using a graphical user interface (GUI). The options displayed on GUI pages are identical to the options displayed in command line mode. In command line mode, each option is displayed with the default value in brackets. To accept the default value, press **Enter** on your keyboard. To enter a different value, enter the correct value and press **Enter** on your keyboard. ■

3. Click **Next**. The Application Configuration page is displayed.



4. Select each check box. (In command line mode, press **Enter** to accept the default value for a component, or enter **N** and press **Enter** to skip a component.)

5. Click **Next**. The Password for integmgr page is displayed.

6. Enter the password of the `integmgr` schema.

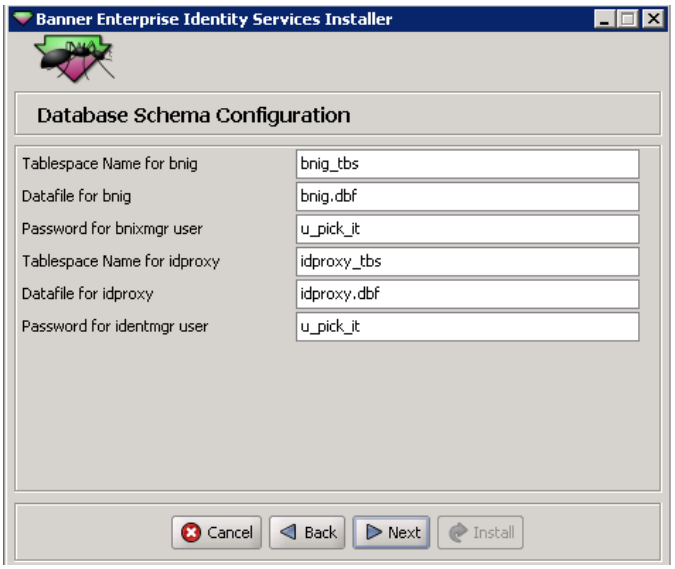
7. Click **Next**. The Database Connection Configuration page is displayed.

8. Enter the following information to configure the database configuration:

Database Host Name	IP address of the database server
Database Port	Port on the database server that is used to connect to the database

Service Name	Name of the database to which you are connecting
DBA username	*baninst1*
Password for DBA username	Password for the `baninst1` user

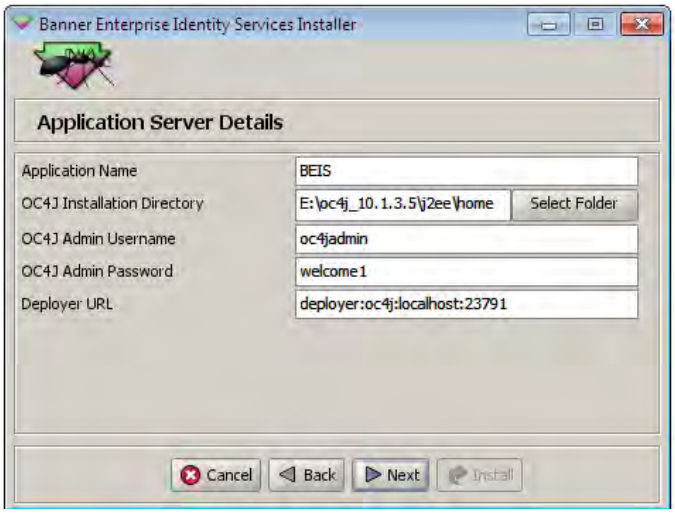
9. Click Next. The Database Schema Configuration page is displayed.



10. Enter the following information to configure the database schema:

Tablespace Name for bnig	Tablespace name for the Banner Identity Gateway schema (for example, *bnig_tbs*)
Datafile for bnig	Name of the datafile with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0/dbs/bnig.dbf*)
Password for bnixmgr user	Password for the Banner Identity Gateway schema (for example, *u_pick_it*)
Tablespace Name for idproxy	Tablespace name for the Identity Proxy schema (for example, *idproxy_tbs*)
Datafile for idproxy	Name of the datafile with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0/dbs/idproxy.dbf*)
Password for identmgr user	Password for the Identity Proxy schema (for example, *u_pick_it*)

11. Click **Next**. The Application Server Details page is displayed.

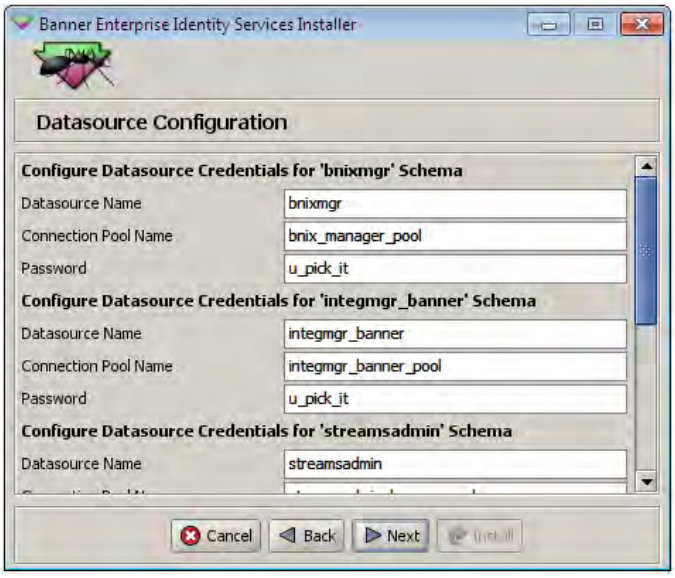


12. Enter the following information to configure the application server:

Application Name	Name for the application (for example, *BEIS*)
OC4J Installation Directory	Installation directory path where the application is to be installed
OC4J Admin Username	User name used to log in to the Oracle Enterprise Manager console
OC4J Admin Password	Password used to log in to the Oracle Enterprise Manager console

Deployer URL	**For standalone OC4J server:**
	`deployer:oc4j:localhost:23791`
	Port 23791 is the rmi port.
	For OAS SOA Suite:
	`deployer:oc4j:<ormis:>opmn://<opmn host>:<opmn port>/<oc4j instance name>`
	Example:
	`deployer:oc4j:opmn://m038071:6004/Ant_Installer_Test`
	Port 6004 is the opmn request port.

13. Click **Next**. The Datasource Configuration page is displayed.



14. Enter the following information to configure data source credentials for the `bnixmgr` schema:

Datasource Name	*bnixmgr*
Connection Pool Name	*bnix_manager_pool*
Password	Password for the `bnixmgr` schema

15. Enter the following information to configure data source credentials for the `integmgr_banner` schema:

Datasource Name	*integmgr_banner*
Connection Pool Name	*integmgr_banner_pool*
Password	Password for the `integmgr_banner` schema

16. Enter the following information to configure data source credentials for the `streamsadmin` schema:

Datasource Name	*streamsadmin*
Connection Pool Name	*streamsadmin_banner_pool*
Password	Password for the `streamsadmin` schema

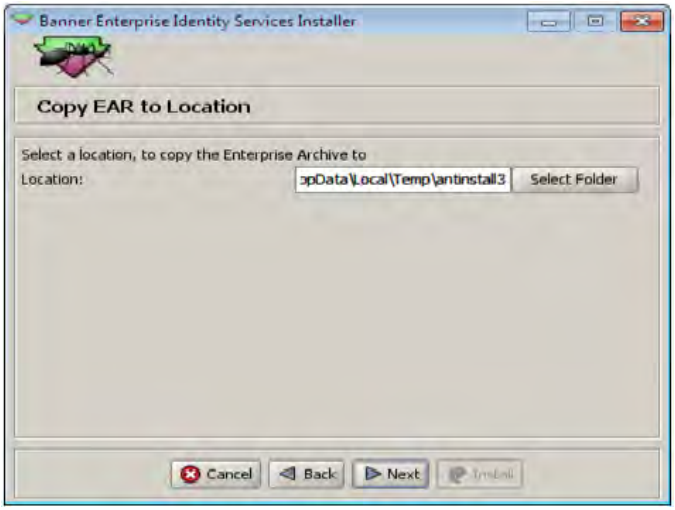
17. Enter the following information to configure data source credentials for the `Identmgr` schema:

Datasource Name	*identmgr*
Connection Pool Name	*identmgr*
Password	Password for the `Identmgr` schema

18. Enter the following information to configure data source credentials for the `Inbadmin` schema:

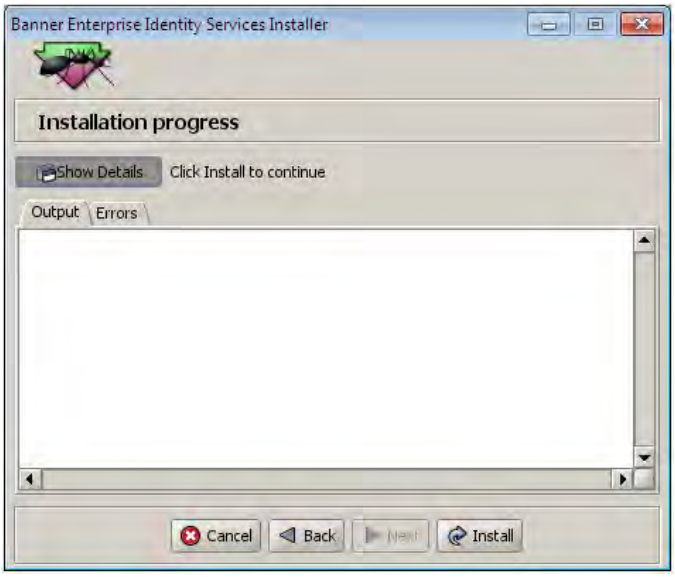
Datasource Name	*Banner_Security*
Connection Pool Name	*Banner_Security_pool*
Password	Password for the `Inbadmin` schema

19. Click **Next**. The Copy EAR to Location page is displayed.



20. Select the folder where the configured ear file should be copied before deployment to the application server. The file can be used for subsequent, manual deployments.

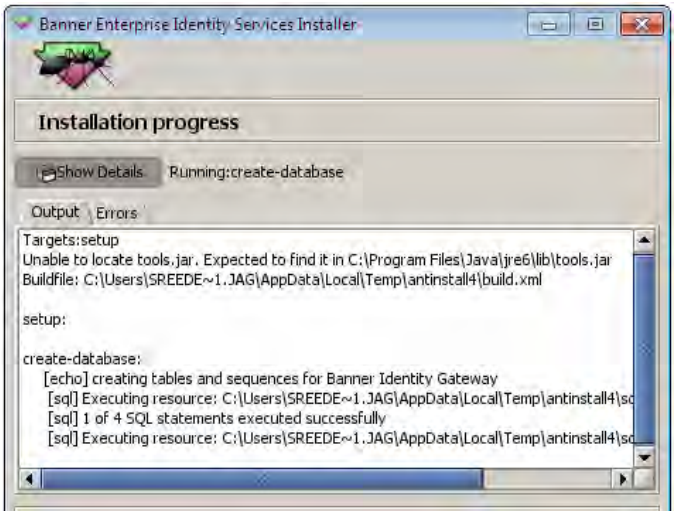
21. Click **Next**. The Installation Progress page is displayed.



22. Click **Show Deta**ils.

23. Click **Install**.

The installation starts. Installation details are displayed as the installation progresses.



The message “Install Finished” is displayed when the installation is complete.

Step 3 Configure security roles and users

On Oracle Application Server 10.1.3.4/5, you must configure the following security roles and administrative users manually:

- `bnixadmin` role - This role and user are required for accessing the Gateway administrative interface.
- `idpadmin` role - This role and user are required for accessing the Identity Proxy administrative interface.

Use the following steps to add these roles and users.

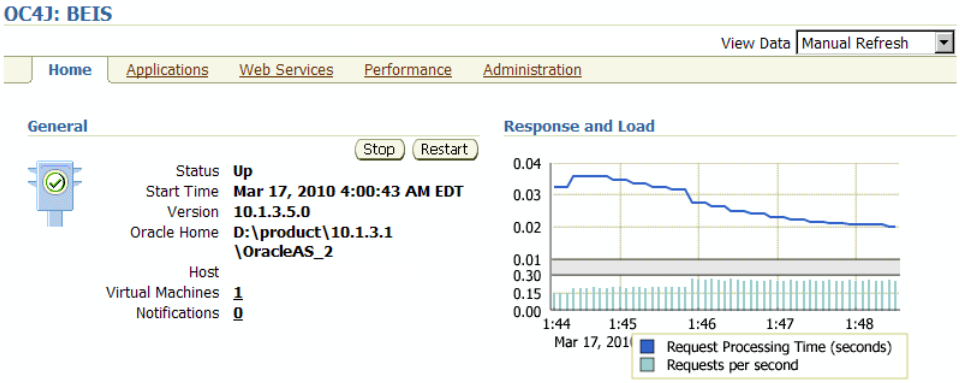
1. Connect to the Oracle Enterprise Manager:

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

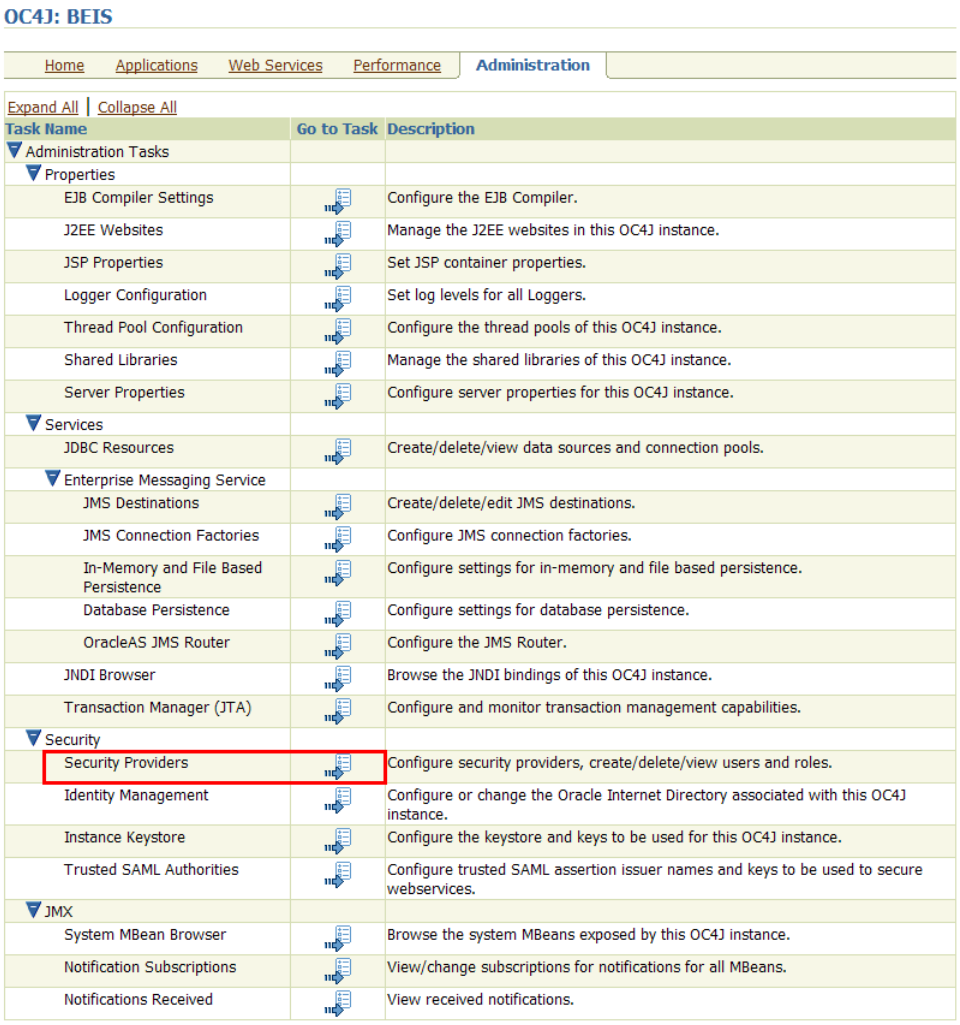
   The console is displayed.

2. Click the name of the OC4J instance where the Gateway and Identity Proxy are installed.

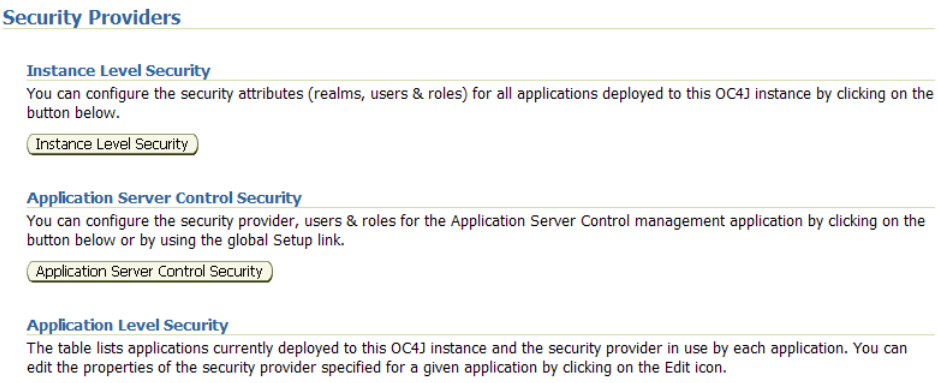
   The Home page for the selected instance is displayed.



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

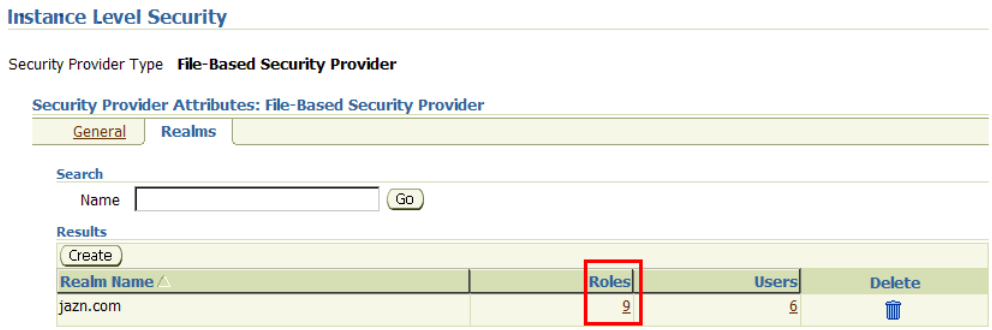


4. Select **Security Providers** in the Security section. The Security Providers page is displayed.



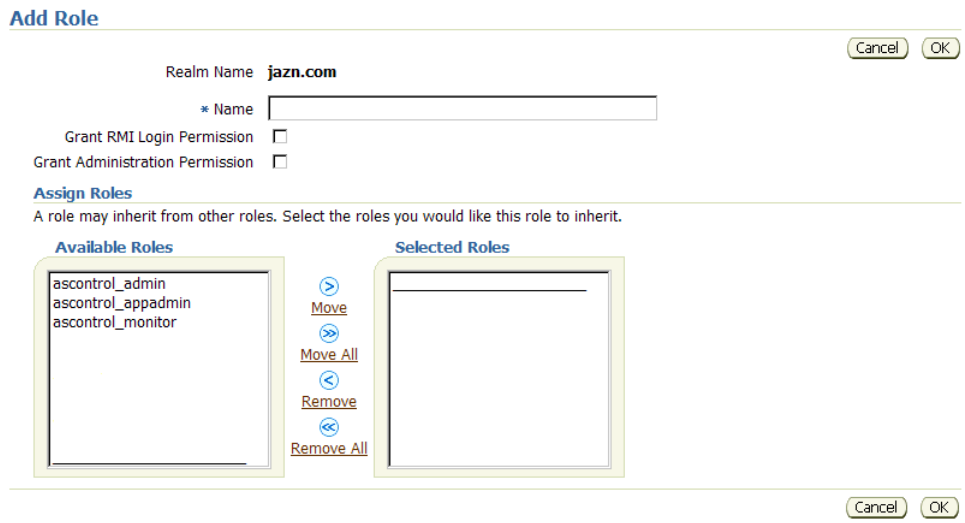
5. Click **Instance Level Security**. The Instance Level Security page is displayed.

6. Select the **Realms** tab.



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

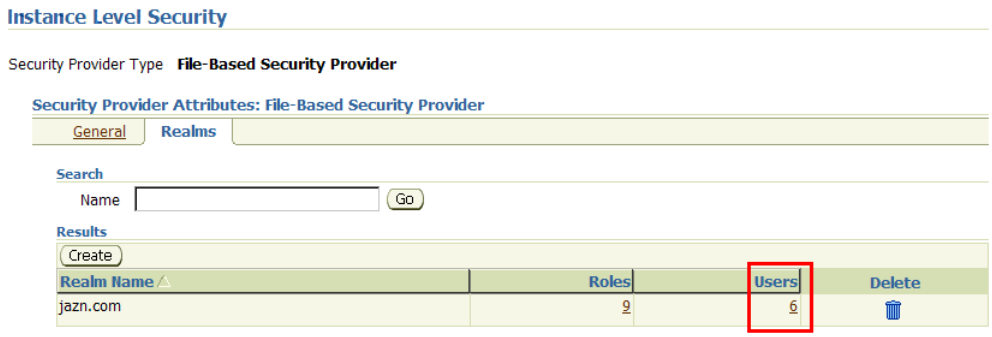
8. Click **Create**. The Add Role page is displayed.



9. Enter *bnixadmin* in the **Name** field.

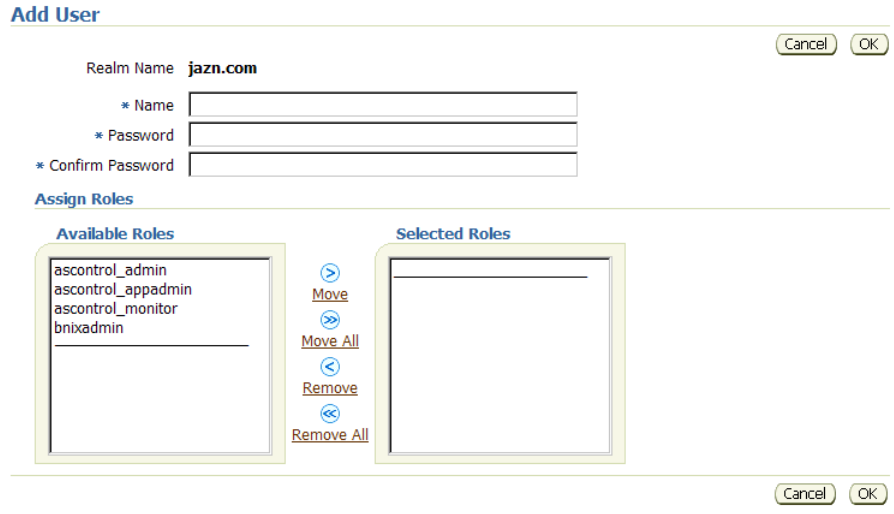
10. Click **OK**. The Roles page is redisplayed with the new role.

11. Return to the Instance Level Security page.



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

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



14. Enter the following information to create a user:

Name	*bnix* (This is an example. Enter the name of your choice.)
Password	Password used by the user to log in to the Gateway administrative interface
Confirm Password	Confirmation of the password

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

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

17. Return to the Instance Level Security page.

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

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

20. Enter *idpadmin* in the **Name** field.

21. Click **OK**. The Roles page is redisplayed with the new role.

22. Return to the Instance Level Security page.

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

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

25. Enter the following information to create a user:

Name	*idproxy* (This is an example. Enter the name of your choice.)
Password	Password used by the user to log in to the Identity Proxy administrative interface
Confirm Password	Confirmation of the password

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

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

Step 4 Configure logging

The Gateway and Identity Proxy use Apache's log4j to log the activities performed by the applications 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 file**. The default locations are `<OAS_HOME>/j2ee/home/bnig_application.log` and `<OAS_HOME>/j2ee/home/idp_application.log`. These locations should be changed to the OC4J instance where the applications are installed.
- **Logging level**. The default level is *INFO*, resulting in limited information (INFO, WARNING, ERROR, and FATAL level statements) being stored in log files. To provide detailed logging, you should modify the log4j configurations.

Note

The `Ant.installer.log` file does not display the whole log. ■

Logging for the Gateway

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

1. Navigate to `<OAS_HOME>/j2ee/<OC4J instance>/applications/bnig/bnigWeb/WEB-INF/classes`.
2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
`log4j.appender.out.File`	`bnig_application.log`	`../<OC4J instance>/log/ bnig_application.log`
`log4j.rootCategory`	`INFO,out`	`DEBUG,out`

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

Logging for the Identity Proxy

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

1. Navigate to `<OAS_HOME>/j2ee/<OC4J instance>/applications/IdProxy/IdProxyWeb/WEB-INF/classes`.

2. Edit `log4j.properties` as follows:

Property	Original Value	New Value
`log4j.appender.out.File`	`idp_application.log`	`../<OC4J instance>/log/ idp_application.log`
`log4j.rootCategory`	`INFO,out`	`DEBUG,out`

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

Installation on Oracle WebLogic Server 11g

`Deployables\Weblogic\beis_weblogic_installer.jar` is used for installation on Oracle WebLogic Server 11g.

Recommended configuration

The Gateway and the Identity Proxy must be installed together in an Oracle WebLogic Basic Domain Managed Server. They must not be installed in an Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual server for BEIS and other middle-tier components. This server would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server instance should consist of the default Admin Server and at least two Managed Servers:

- One Managed Server for the Gateway and the Identity Proxy, which must be installed together
- One Managed Server for the Identity Data Export Utilities and the SSO Manager

If a domain based on the Basic Domain template already exists for middle-tier applications, the BEIS components can be installed in two separate Managed Servers in that domain, based on the preceding recommendation.

Refer to the Oracle WebLogic Server Documentation Library for details on creating a new domain and a new Managed Server.

Installation steps

Use the following steps to use the automated installer to install the Gateway and the Identity Proxy on Oracle WebLogic Server 11g (version 10.3.2).

- Step 1, "Configure the domain security model"
- Step 2, "Configure the authentication provider"
- Step 3, "Configure and install the applications"

Step 1 Configure the domain security model

The Oracle WebLogic Server must be configured to use the *Advanced* security model instead of the default *DD only* option. This step pertains to the realm configuration. It applies to the entire domain. (Although you can create a totally new realm for the domain, only one realm can be active at a time for the entire domain.)

Note

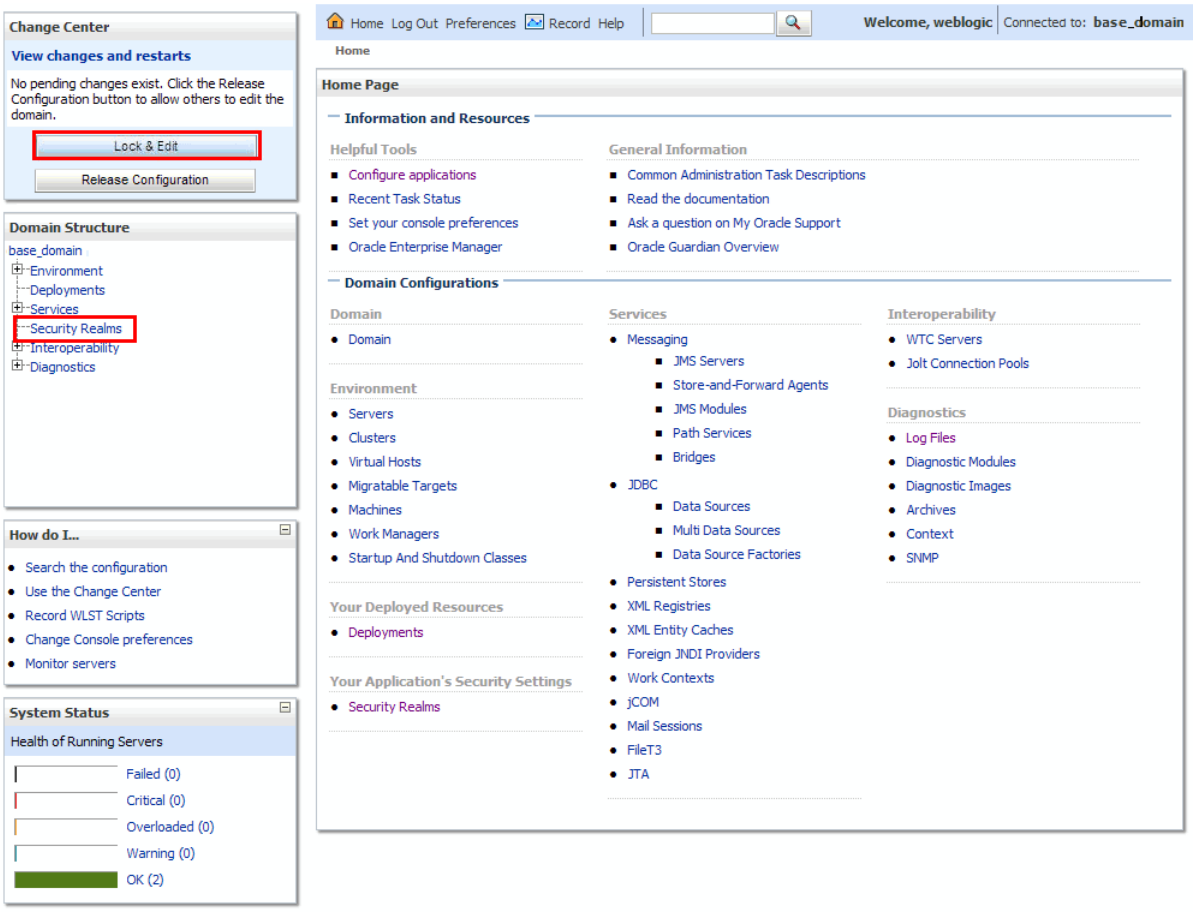
After you run the installer, you can set the security model back to the default *DD only* option.

Use the following steps to configure the domain security model.

1. Connect to the Oracle WebLogic Server Administration Console:

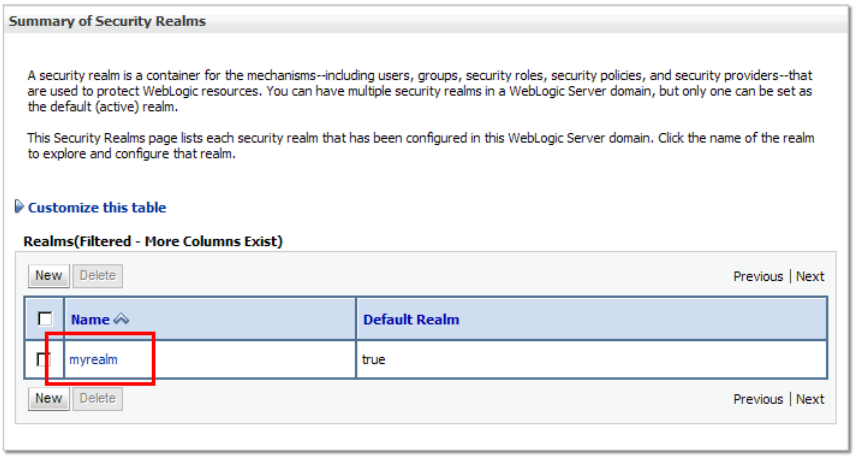
   ```
   http://<host>:<port>/console
   ```

The Home Page is displayed.

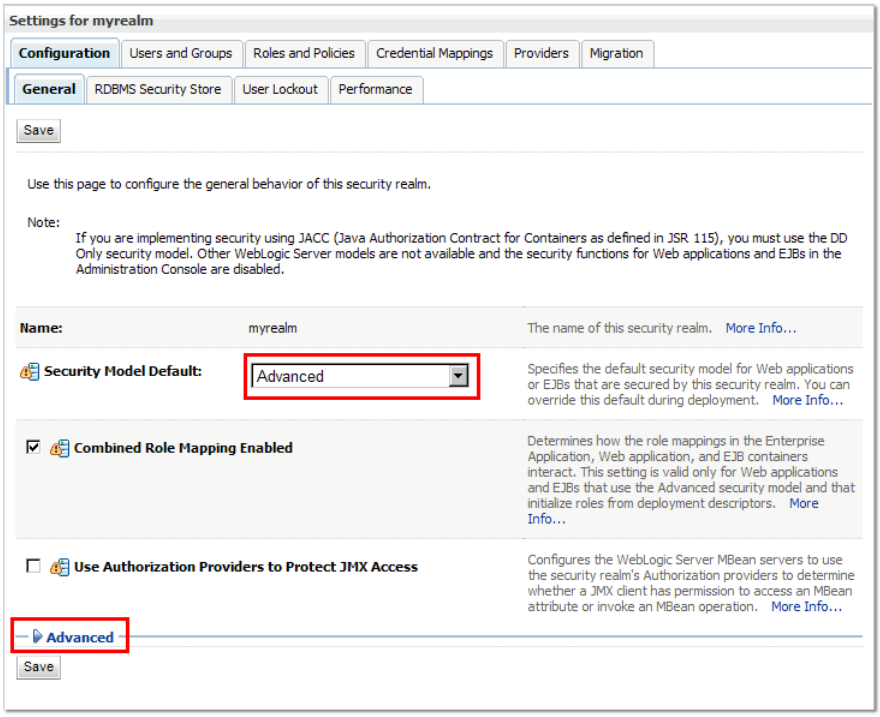


2. In the Change Center pane, click **Lock & Edit**.

3. In the Domain Structure pane, click **Security Realms**. The Summary of Security Realms page is displayed.

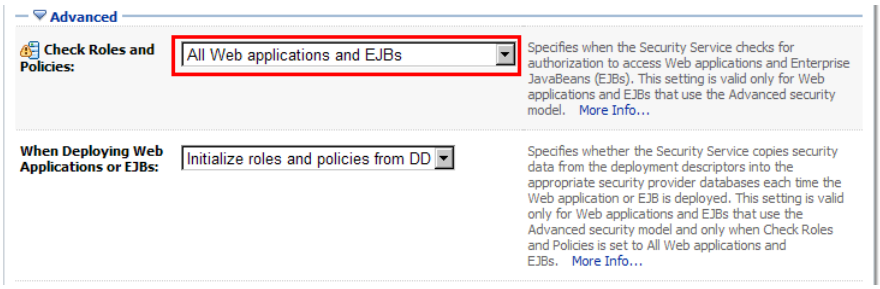


4. Click **myrealm**. The Settings page is displayed.



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

6. 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 the authentication provider

An authentication provider must be configured in Oracle WebLogic Server 11g to allow for basic authentication against the Web services that the components expose. The

authentication provider is set via a JAAS configuration file. The Oracle WebLogic Managed Server where the components will be deployed must be configured to load the configuration file on startup. Use the following steps to configure the authentication provider.

1. If a JAAS configuration file already exists for the Oracle WebLogic domain where the components will be deployed, skip to step 3.

   If a JAAS configuration file does not exist for the Oracle WebLogic domain where the components will be deployed, use a text editor to create the `jaas.config` file with the following content:

   ```
   myrealm {
      weblogic.security.auth.login.UsernamePasswordLoginModule
      REQUIRED;
   };
   ```

2. Save `jaas.config` in the following location:

   ```
   <WebLogic Home>/user_projects/domains/<your domain dir>/config
   /security
   ```

   where `<WebLogic Home>` is the base directory for the Oracle WebLogic software packages and configuration files, and `<your domain dir>` is the domain where the components will be deployed.

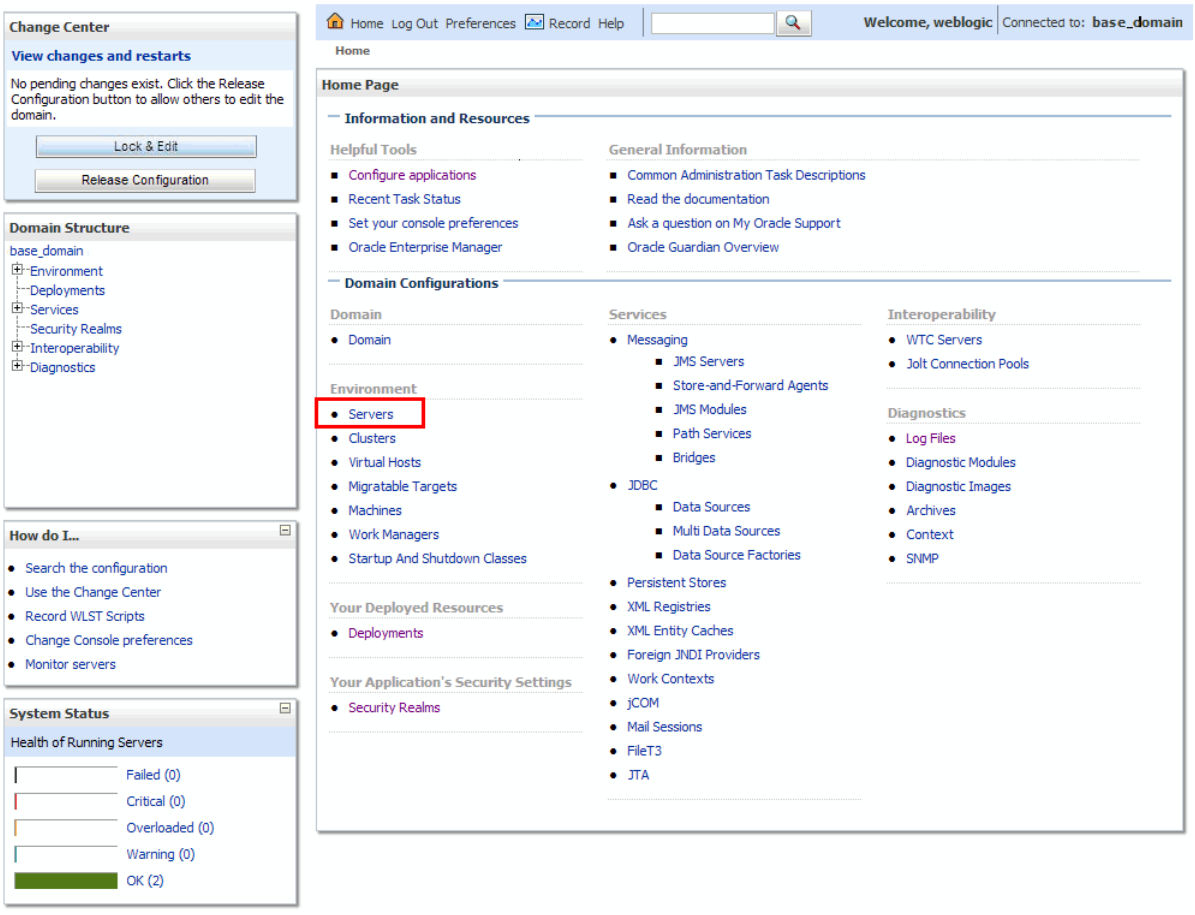
3. Configure the Managed Server to use the authentication provider.

   There are two ways to configure the Managed Server, depending on how you want to start the Managed Server. Use option 1 (page 9-22) if the Managed Server will be started by using the Oracle WebLogic Administration Console. Use option 2 (page 9-26) if the Managed Server will be started by running a script.

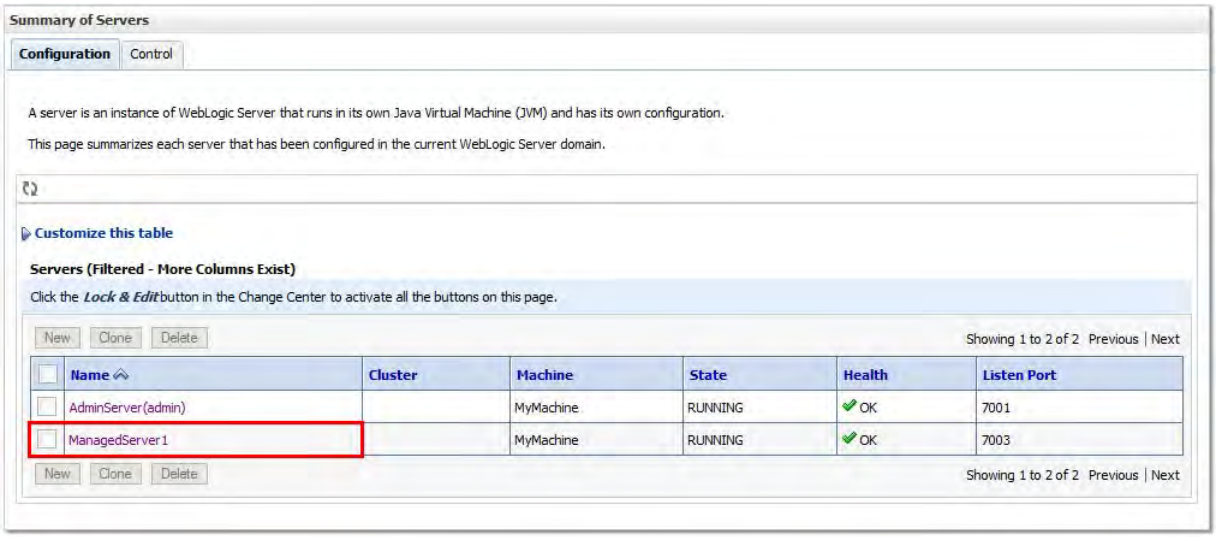
Option 1 - If you are using the administration console to start the Managed Server

With this option, the location of the JAAS configuration file is set as an argument on the Server Start tab of the specific Managed Server. The location of the JAAS configuration file applies only to that specific Managed Server.

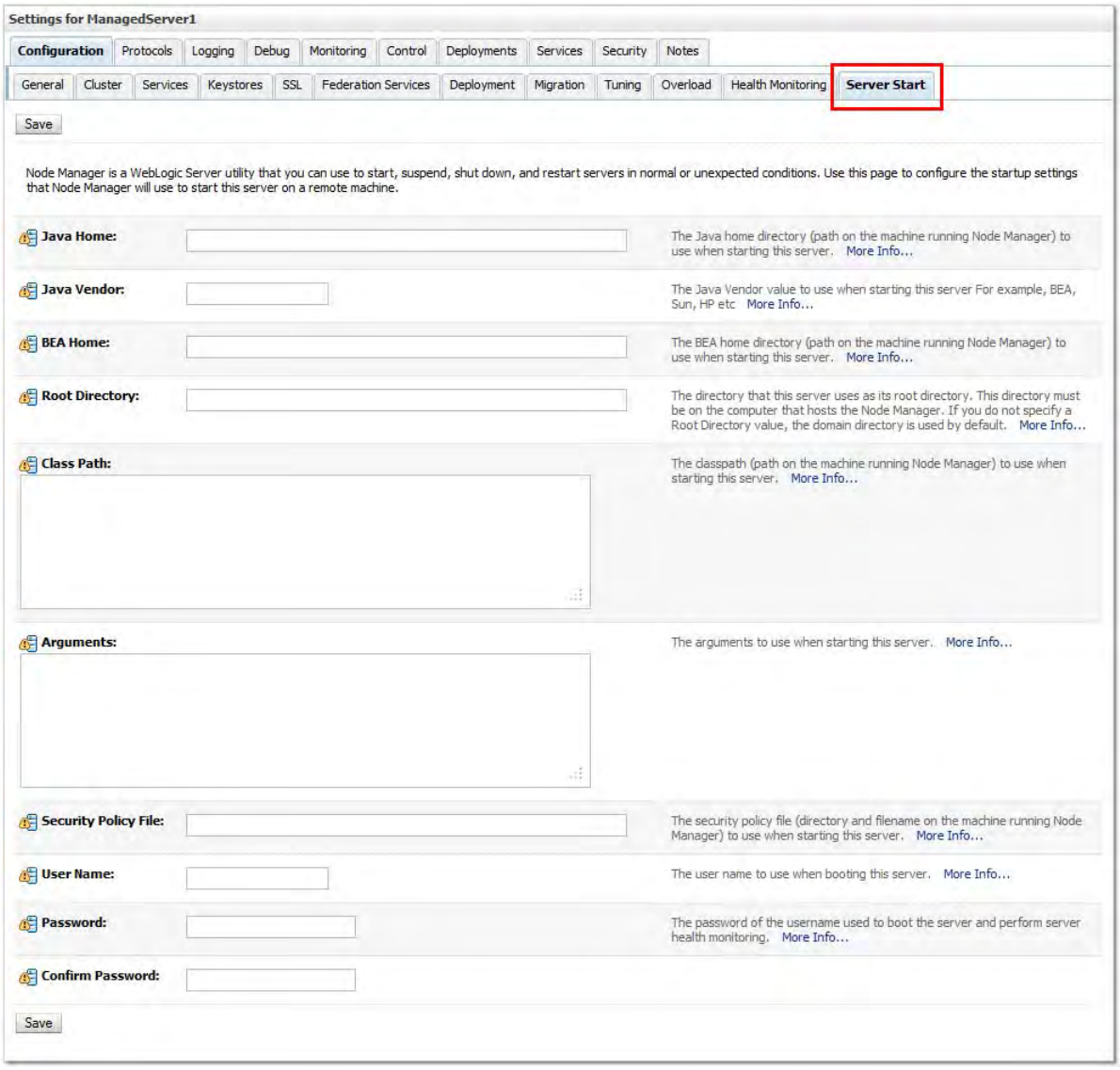
3.1. Connect to the Oracle WebLogic Server Administration Console for the domain where the components will be deployed. The Home Page is displayed.



3.2. Click **Servers**. The Summary of Servers page is displayed.



3.3. Click the name of the server where the components will be deployed. The Settings page is displayed.



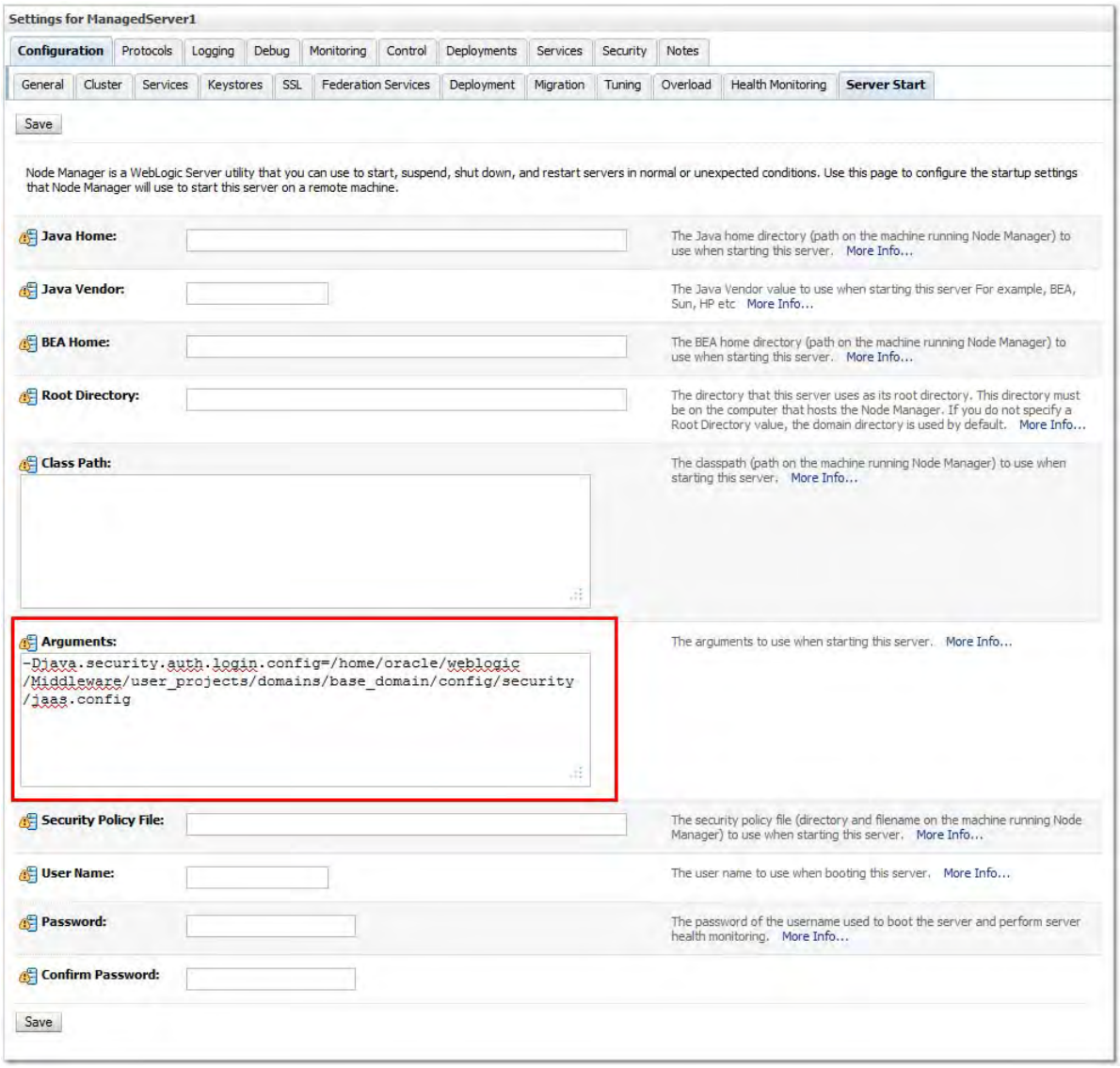
3.4. Click the **Server Start** tab.

3.5. Click **Lock & Edit** in the Change Center pane.

3.6. In the **Arguments** field enter the full path to the `jaas.config` file, including the file name:

```
-Djava.security.auth.login.config=<WebLogic Home>/
user_projects/domains/< domain dir>/config/security/
jaas.config
```

where `<WebLogic Home>` is the base directory for all Oracle WebLogic software packages and configuration files, and `<your domain dir>` is the domain where the components will be deployed.



3.7. Click **Save.**

3.8. Click **Activate Changes** in the Change Center pane.

Option 2 - If you are using a script to start the Managed Server

Use this option if the Managed Server will be started by running the `startManagedWebLogic.sh` (or `.cmd`) script. A `JAVA_OPTIONS` statement must be added to the `setDomainEnv.sh` (or `.cmd`) script. The location of the JAAS configuration file applies to the entire domain, including the Admin Server and all Managed Servers.

Use the following steps to update the script for Windows.

3.1. Open the `setDomainEnv.cmd` file located under `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

3.2. Search for the last occurrence of the following text:

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
```

3.3. Add the following in the line preceding the line identified in step 3.2.

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Djava.security.auth.login.config=
<domain home>\config\security\jaas.config
```

Use the following steps to update the script for Linux/Unix.

3.1. Open the `setDomainEnv.sh` file located under `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

3.2. Search for the last occurrence of the following text:

```
JAVA_OPTIONS="${JAVA_OPTIONS}"
```

3.3. Add the following in the line preceding the line identified in step 3.2.

```
JAVA_OPTIONS="${JAVA_OPTIONS}
-Djava.security.auth.login.config=
<domain home>/config/security/jaas.config"
```

Note

There is a space between the closing brace and the dash (that is, {JAVA_OPTIONS}**[space]**-Djava). ■

4. Restart the appropriate server(s).

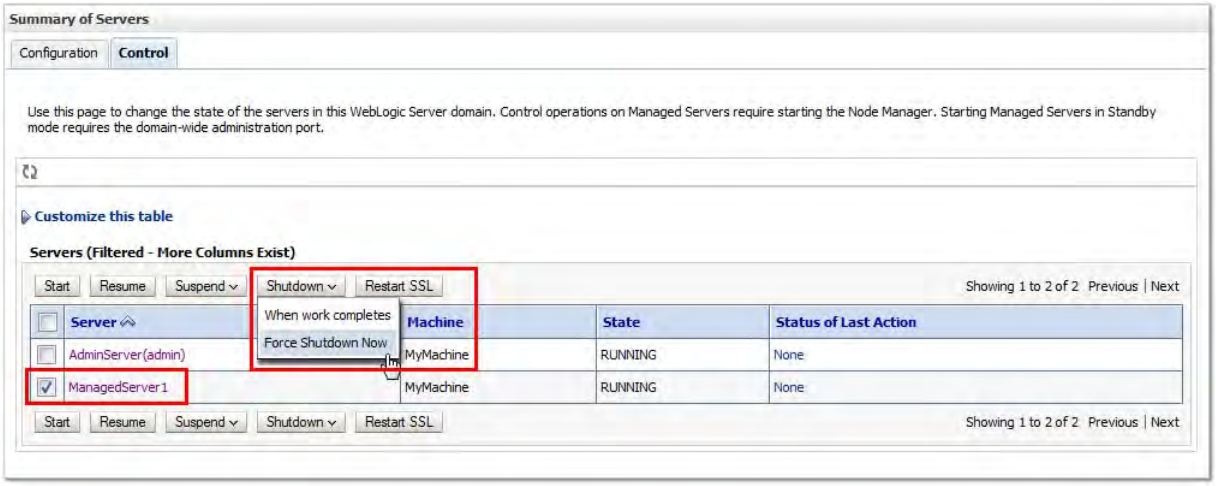
   There are two ways to restart the server(s). Use option 1 (page 9-27) if a single Managed Server was configured. Use option 2 (page 9-29) if all servers in the domain were configured.

Option 1 - If a single Managed Server was configured

Use this option if a single Managed Server was configured. Only that server needs to be restarted.

4.1. Navigate to the Summary of Servers page.

4.2. Click the **Control** tab.

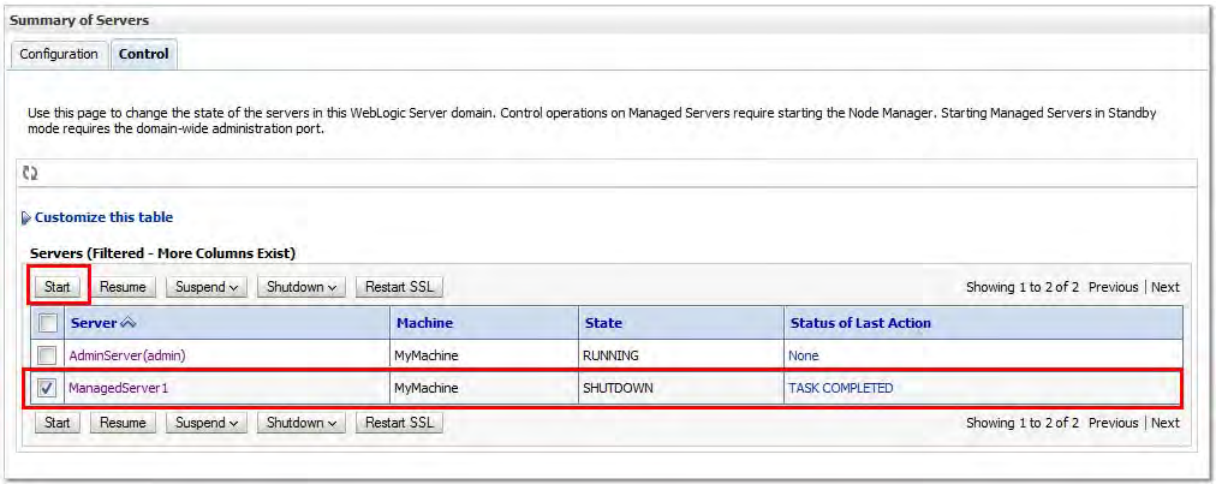


4.3. Select the Managed Server where the configuration changes were made.

4.4. Click **Shutdown** -> **Force Shutdown Now**.

4.5. Confirm the selection.

4.6. Wait for the server to enter a *SHUTDOWN* state.

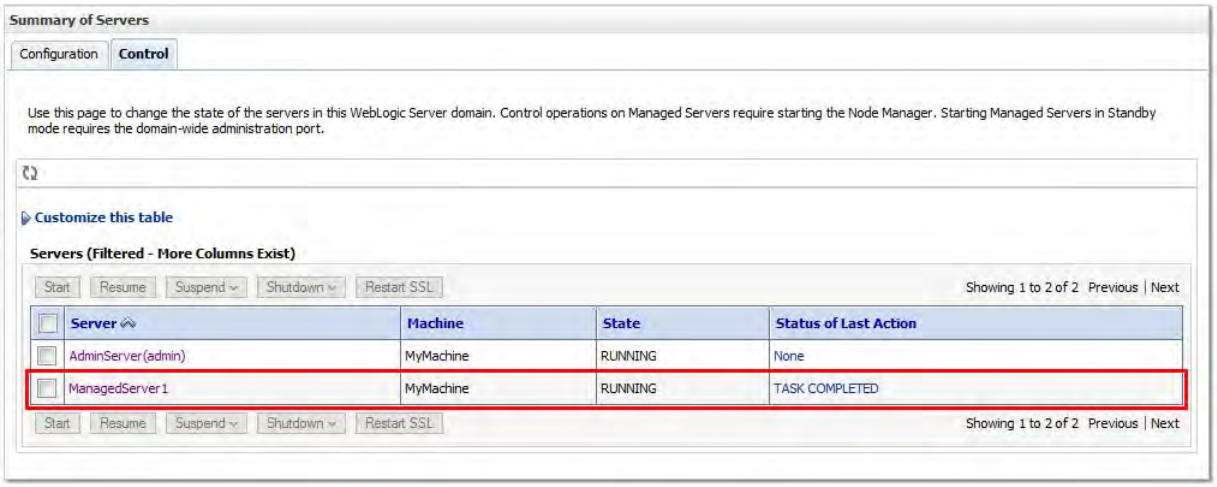


4.7. Select the same Managed Server.

4.8. Click **Start**.

4.9. Confirm the selection.

4.10. Wait for the server to enter a *RUNNING* state.



Option 2 - If all servers in the domain were configured

Use this option if all servers in the domain were configured. Only the Managed Server needs to be restarted.

Use the following steps to restart the server for Windows.

4.1. Navigate to `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

4.2. Stop the server by running the following script:

```
stopManagedWebLogic.cmd <ServerName>
```

Note

There is a space between the command and <ServerName>; that is, stopManagedWeblogic.cmd**[space]**<ServerName>. ■

Example:

```
stopManagedWebLogic.cmd ManagedServer1
```

4.3. Start the server by running the following script:

```
startManagedWebLogic.cmd <ServerName>
```

Use the following steps to restart the server for Linux/Unix.

4.1. Navigate to `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

4.2. Stop the server by running the following script:

```
./stopManagedWebLogic.sh <ServerName>
```

Note

There is a space between the command and <ServerName>; that is, ./stopManagedWeblogic.sh**[space]**<ServerName>. ■

Example:

```
./stopManagedWebLogic.sh ManagedServer1
```

4.3. Start the server by running the following script:

```
./startManagedWebLogic.sh <ServerName>
```

Step 3 Configure and install the applications

Use the following steps to install the Gateway and Identity Proxy with the automated installer.

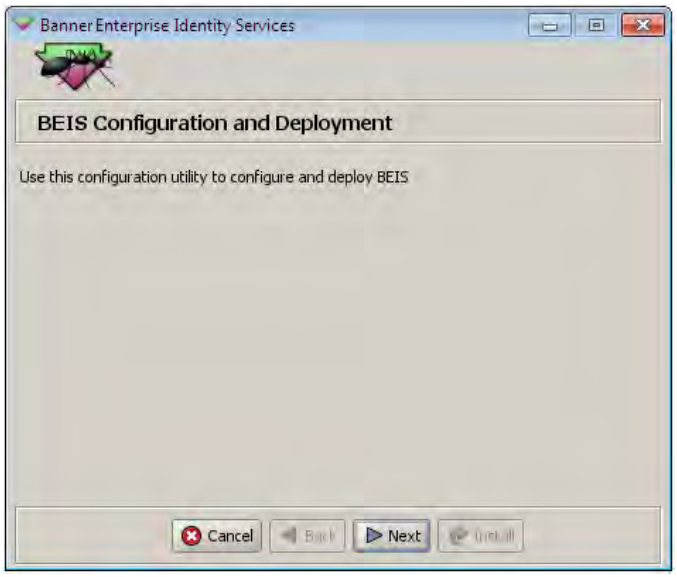
1. Open a command prompt and navigate to the directory where the installer jar file is located.

2. Choose one of the following:

   2.1. If you are using a windowing environment, run the following command:

   ```
   E:\AntInstaller>java -jar beis_weblogic_installer.jar
   ```

   The installer is extracted, and a splash screen is displayed.



   2.2. If you are using a non-windowing environment, run the following command:

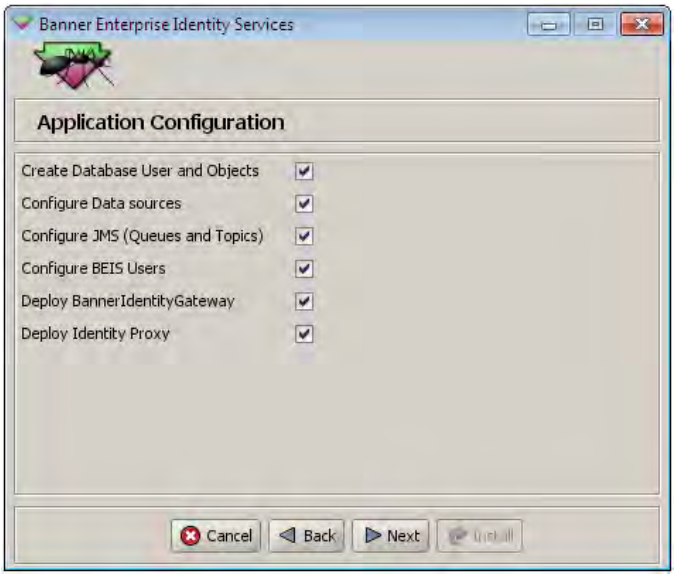
   ```
   [oracle@octopus Ant_Installer_Test]$ java -jar
   beis_weblogic_installer.jar
   ```

   The self extractor is loaded. If the environment has graphics, a splash screen is displayed. Otherwise, a command line interface is displayed.

Note

The remaining instructions are based on using a graphical user interface (GUI). The options displayed on GUI pages are identical to the options displayed in command line mode. In command line mode, each option is displayed with the default value in brackets. To accept the default value, press **Enter** on your keyboard. To enter a different value, enter the correct value and press **Enter** on your keyboard. ■

3. Click **Next**. The Application Configuration page is displayed.



4. Select each check box. (In command line mode, press **Enter** to accept the default value for a component, or enter **N** and press **Enter** to skip a component.)

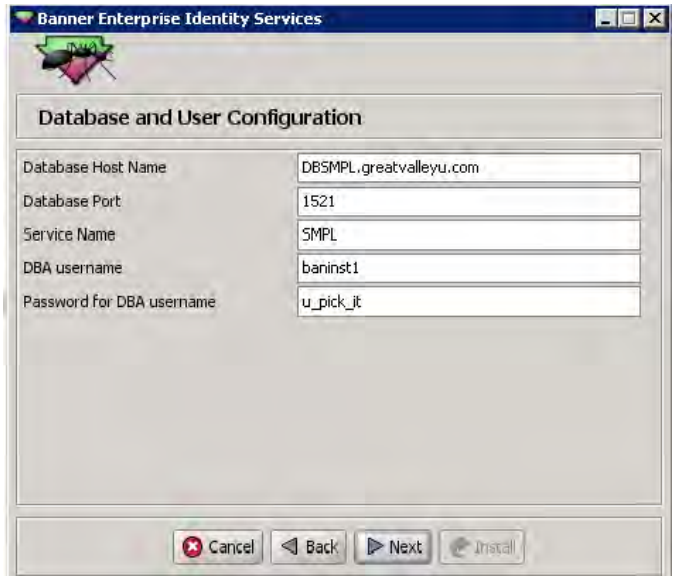
Note

Do not select **Configure Data Sources** if you previously created data sources manually. The installer cannot recreate data sources that were created manually.

5. Click **Next**. The Password for integmgr page is displayed.

6. Enter the password of the `integmgr` schema.

7. Click **Next**. The Database and User Configuration page is displayed.



8. Enter the following information to configure the database and user:

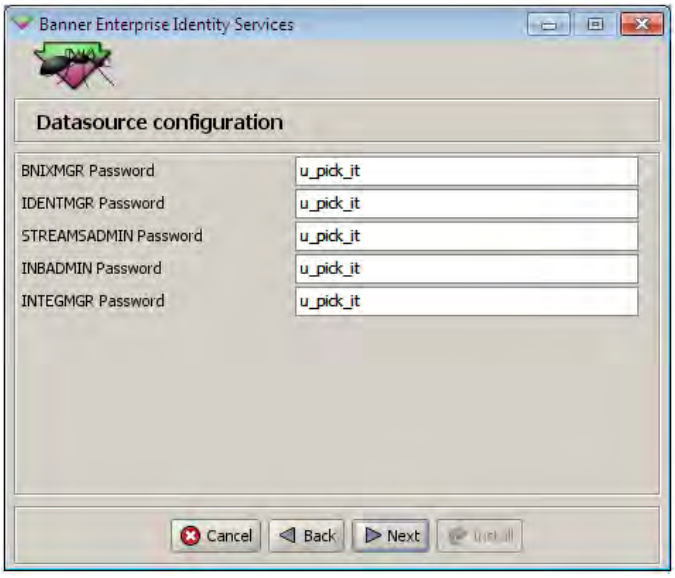
Database Host Name	IP address of the database server
Database Port	Port on the database server that is used to connect to the database
Service Name	Name of the database to which you are connecting
DBA username	*baninst1*
Password for DBA username	Password for the `baninst1` user

9. Click **Next**. The Application Server Details page is displayed.

10. Enter the following information to configure the application server:

Weblogic Server Home Directory	Installation directory path where the applications are to be installed
Weblogic Admin Username	User name used to log in to the Oracle WebLogic Server Administration Console
Weblogic Admin Password	Password used to log in to the Oracle WebLogic Server Administration Console
Weblogic Server Host	IP address or host name of the machine where the Oracle WebLogic server is installed
Weblogic Server Port	Port on which the WebLogic Admin Server is running
Weblogic Server Name	Name of the Managed Server where the data sources and JMS artifacts are created, and where the applications are deployed

11. Click **Next**. The Datasource Configuration page is displayed.



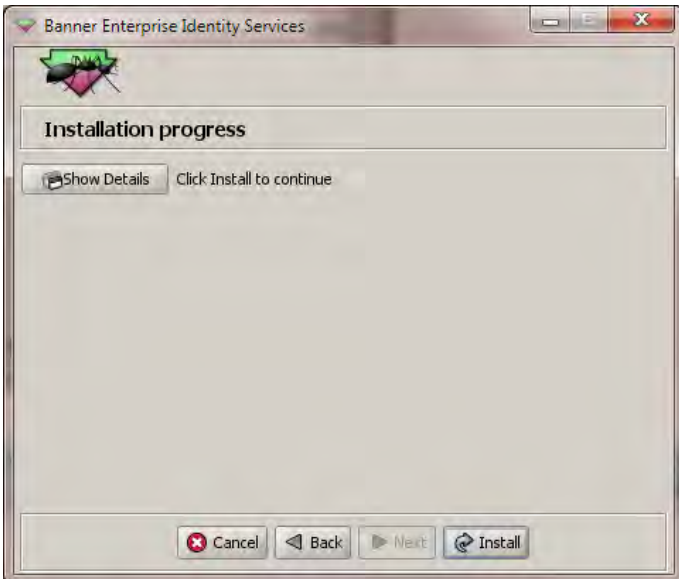
12. Enter the database user passwords that will be used to create data sources for the Gateway and Identity Proxy.

13. Click **Next**. The BEIS Security Configuration page is displayed.

14. Enter the following information to identify the administrative user:

BNIG User Name	User name used to log in to the Banner Identity Gateway administrative interface
Password for BNIG User	Password used to log in to the Banner Identity Gateway administrative interface
Identity Proxy User Name	User name used to log in to the Enterprise Identity Proxy Services administrative interface
Password for Identity Proxy User	Password used to log in to the Enterprise Identity Proxy Services administrative interface

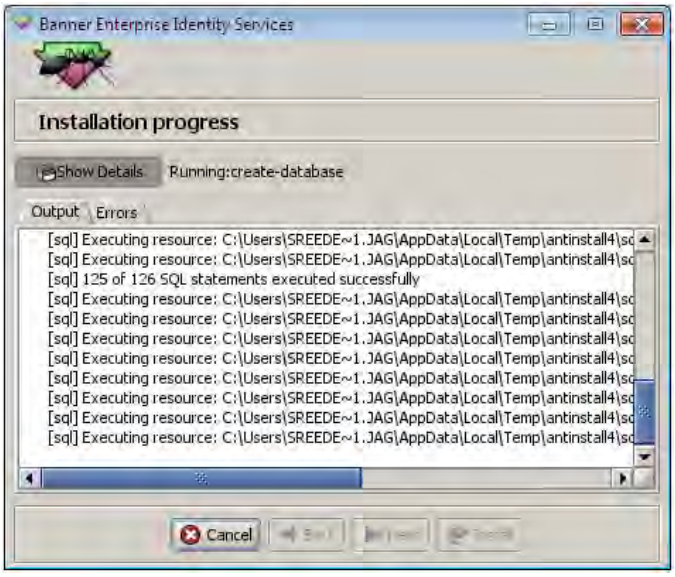
15. Click **Next**. The Installation Progress page is displayed.



16. Click **Show Details**.

17. Click **Install**.

The installation starts. Installation details are displayed as the installation progresses.



10 SPML LDAP Adapter

The SPML LDAP Adapter creates user accounts in an LDAP V3 compliant directory server. UDCIdentity element data are mapped to LDAP attributes by specifying XPath entries to define the mapping. The adapter can be used with the SPML Publisher (see Chapter 6, “Identity Data Export Utilities”) as one way to initially load accounts from Banner® to an LDAP directory. A secondary use of the adapter is realtime account provisioning from Banner.

The adapter is packaged in two zip files:

- `Deployables\OC4J\ldap_spml_psp_full_release.zip` is used for installation on Oracle Application Server 10.1.3.4/5.
- `Deployables\Weblogic\ldap_spml_psp_full_release.zip` is used for installation on Oracle WebLogic Server 11g.

This chapter gives instructions for installing the adapter on both servers.

Installation on Oracle Application Server

`Deployables\OC4J\ldap_spml_psp_full_release.zip` is used for installation on Oracle Application Server 10.1.3.4/5. This zip file contains an archive file named `ldap-spml-psp.ear`.

The SPML LDAP Adapter can be installed on an existing Oracle Application Server. The adapter can be deployed into a new or existing OC4J instance. The adapter can be deployed into the same OC4J instance as other Banner Enterprise Identity Services (BEIS) applications or into its own OC4J instance. In either case, BEIS applications should be deployed separately from other applications so they can be managed independently.

Use the following steps to install the SPML LDAP Adapter on OAS 10.1.3.4/5.

- Step 1, “Extract the ear file”
- Step 2, “Configure the SPML LDAP Adapter”
- Step 3, “Rebuild the ear file”
- Step 4, “Install the SPML LDAP Adapter”

Step 1 Extract the ear file

Use the following steps to extract the `ldap-spml-psp.ear` file.

1. Copy the `ldap-spml-psp.ear` file to a temporary location. This location is referred to as `<EAR_HOME>`.

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

   ```
   jar xvf ldap-spml-psp.ear
   ```

   The extract contains an archive named `ldap-spml-psp.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>/ldap-spml-psp.war
   ```

   The extract contains a directory named `WEB-INF`.

Step 2 Configure the SPML LDAP Adapter

Use the following properties files to configure the SPML LDAP Adapter. These files are located in the `war_home/WEB-INF/classes` directory.

```
ldap.properties
ldap-namespace.properties
ldap-user-profile.properties
```

ldap.properties

The properties in `ldap.properties` must be changed to reflect your environment.

Note

Use a space to separate the property name from the property value. ■

Property Name	Property Value	Example
`ldap_basedn`	ldap base dn for the directory	`dc=sungardhe,dc=com`
`ldap_username`	Privileged ldap user that can create user and group entries	`cn=Manager, dc=sungardhe, dc=com`
`ldap_password`	Password associated with `ldap_username`	`root`
`ldap_url`	URL to connect to the directory	`ldap://localhost:389`
`ldap_factory`	Java class that is used to connect to the directory	`com.sun.jndi.ldap. LdapCtxFactory`
`ldap_authentication`	Type of authentication	`simple`
`ldap_person_dir`	ldap dn where person data is stored relative to the basdn. Must be present in the LDAP directory server.	`ou=users`
`ldap_group_provision_ enabled`	Set to *true* to enable provisioning of group entries	`false`
`ldap_group_dir`	ldap dn where group entries are stored relative to the basdn. Must be present in the LDAP directory server.	`ou=roles`
`ldap_groupObject Classes`	ldap object class hierarchy for the groups directory	`groupOfNames`
`ldap.objectclasses`	ldap object class hierarchy for your directory	`=top,person, organizationalPerson, inetOrgPerson`

Property Name	Property Value	Example
`ldap.distinguished_name`	Unique identifier of the person in the LDAP user store. This property accepts any LDAP attribute as value. The value is mapped in `ldap-user-profile.properties`. If the UDCIdentifier needs to be configured as the distinguished name, then modify the property as follows: `=cn` The value is provided as `cn` based on the sample mappings in `ldap-user-profile.properties`.	`=uid` In this example, the `uid` LDAP attribute is configured as the distinguished name for the LDAP user store. Based on the sample mappings, the Principal of the person uniquely identifies the person in the LDAP user store.
`ldap_maximum_user_per_request`	Maximum number of UDCIdentity messages that can be processed in a single SOAP request message. (The SPML toolkit can process multiple identities in a single SOAP request.)	`=100`

Note

Group provisioning into a directory whose object class is not `groupOfNames` is not supported.

ldap-namespace.properties

All namespaces that are used in the UDCIdentity.xml document are mentioned as part of the namespace configuration in `ldap-namespace.properties`.

Example

```
ns1=urn:sungardhe:enterprise:domain:identity:1.0
```

`ns1` is a sample only. However, there is a strict dependency between the prefix/key `ns1` that is defined in the `ldap-namespace.properties` file and the XPath queries that are defined in the mapping file `ldap-user-profile.properties`. Mapping

an element (with a namespace prefix) inside the UDCIdentity XML structure against an LDAP attribute is described in the following section.

ldap-user-profile.properties

The `ldap-user-profile.properties` file has the mappings for the LDAP attributes and the UDCIdentity values. The mappings are prefixed with the respective namespaces as defined in `ldap-namespace.properties`. This file maps each LDAP attribute to the XPath expression that points to the UDCIdentity value in the UDCIdentity XML structure.

Example

```
cn = /ns1:UDCIdentity/ns1:UDCIdentifier
```

The `cn` attribute of LDAP is mapped to the UDCIdentifier value under the UDCIdentity tag in the UDCIdentity XML structure. `ns1` is the prefix for the namespace for a particular element in the UDCIdentity XML structure and is defined in `ldap-namespace.properties`.

The following table shows the default mappings provided with the application.

LDAP Attribute	XPath Expression	Description
`cn`	`ns1:UDCIdentity/ ns1:UDCIdentifier`	UDCIdentifier of the person
`sn`	`ns1:UDCIdentity/ ns1:PersonIdentity/ ns1:PersonName/ ns1:FormattedName`	Formatted name of the person
`uid`	`ns1:UDCIdentity/ ns1:Extension/ ns1:Attribute[ns1:name = 'PRINCIPAL']/ ns1:value`	LDAP username used to authenticate to the LDAP server
`userPassword`	`ns1:UDCIdentity/ ns1:Extension/ ns1:Attribute[ns1:name = 'CREDENTIAL']/ ns1:value`	Credential of the person
`userRoles`	`/ns1:UDCIdentity/ ns1:InstitutionRoles/ ns1:institutionrole/ ns1:role`	Group to which the person belongs

The XPath expression can be modified for any LDAP attribute. For more details on the usage of XPath, refer to http://www.w3schools.com/xpath/default.asp.

Step 3 Rebuild the ear file

Use the following steps to rebuild the ear file with the modified configuration properties.

1. Execute the following command from the `war_home` directory:

   ```
   jar cvf <EAR_HOME>/ldap-spml-psp.war WEB-INF/* index*
   ```

   The war file is rebuilt with the modified configurations.

2. Execute the following command from `<EAR_HOME>`:

   ```
   jar cvf ldap-spml-psp.ear * .war META-INF/*
   ```

   The ear file is rebuilt with the modified configurations. The rebuilt ear file is used for installation.

Step 4 Install the SPML LDAP Adapter

Use the following steps to install the SPML LDAP Adapter to the Oracle Application Server.

1. Connect to the Oracle Enterprise Manager:

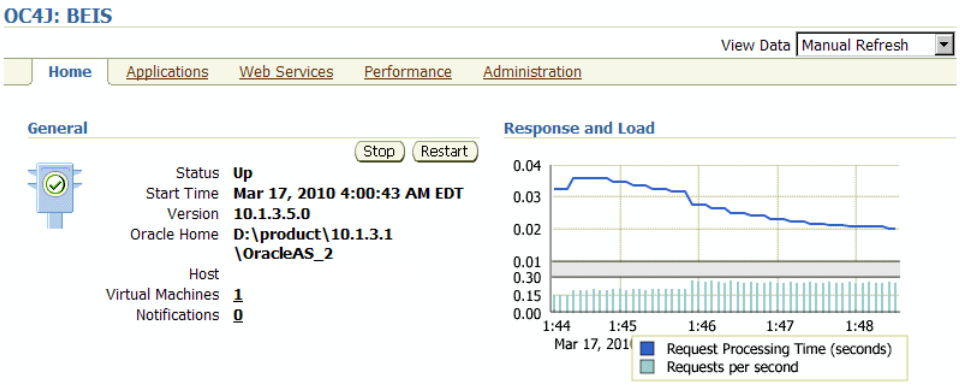
   ```
   http://<host>:<port>/em
   ```

   The console is displayed.

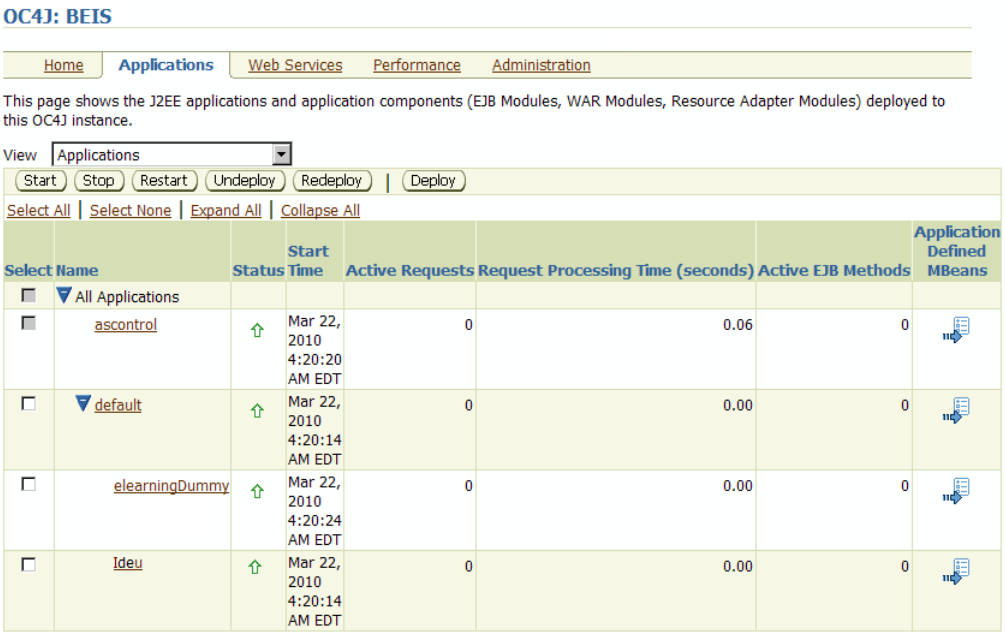
2. Click the name of the OC4J instance that will host the SPML LDAP Adapter.

   The adapter can be installed in the same instance with other BEIS applications or in its own instance. In either case, BEIS applications must be installed separately from other applications so they can be independently managed.

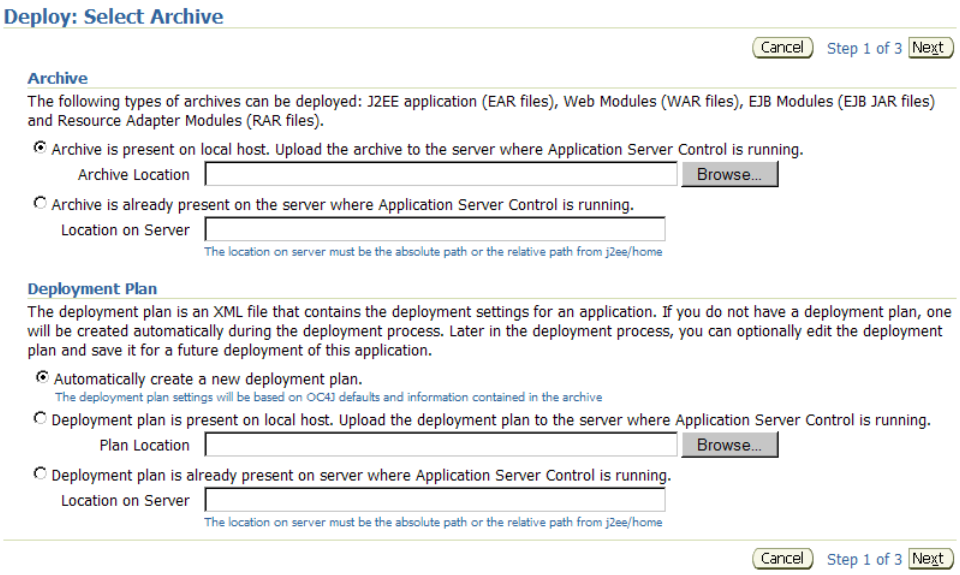
   The Home page for the selected instance is displayed.



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



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



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 navigate to the `ldap-spml-psp.ear` file.

   5.3. Select the file and click **Open**.

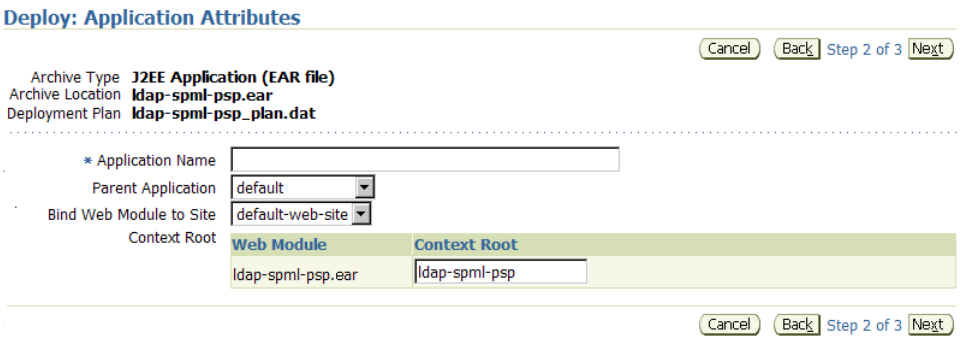
6. Select the deployment plan for the application:

   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 `ldap-spml-psp_plan.dat` file.

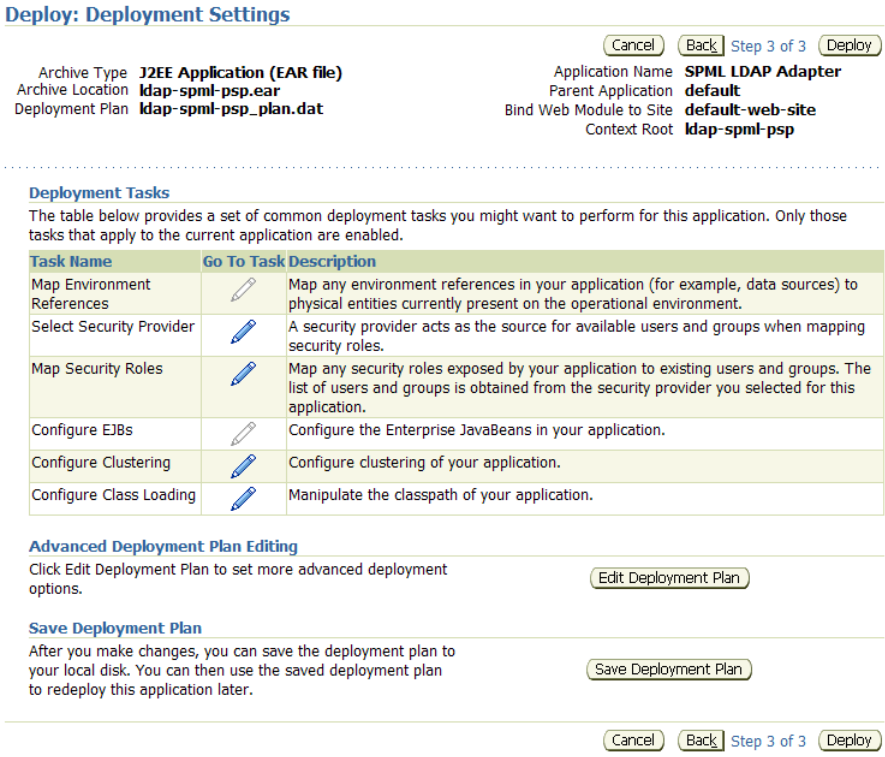
   6.3. Select the file and click **Open**.

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



8. Enter a name for the application (for example, *SPML LDAP Adapter*) in the **Application Name** field.

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



10. Click **Deploy** to accept the values and install the SPML LDAP Adapter. A deployment confirmation page is displayed.

11. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

The SPML endpoint can be accessed with the following URL:

```
http://<host>:<port>/ldap-spml-psp/xfire/psp/DocumentLiteral
```

Installation on Oracle WebLogic Server 11g

`Deployables\Weblogic\ldap_spml_psp_full_release.zip` is used for installation on Oracle WebLogic Server 11g. This zip file contains an archive file named `ldap-spml-psp.ear`.

Recommended configuration

BEIS components must be installed together in an Oracle WebLogic Basic Domain. They must not be installed in an Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual server for BEIS and other middle-tier components. This server would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server instance should consist of the default Admin Server and at least two Managed Servers:

- One Managed Server for the Banner Identity Gateway and the Enterprise Identity Proxy Services, which must be installed together
- One Managed Server for the Identity Data Export Utilities and the SSO Manager

The SPML LDAP Adapter can be installed in either of these Managed Servers or in a separate Managed Server.

If a domain based on the Basic Domain template already exists for middle-tier applications, the BEIS components can be installed in separate Managed Servers in that domain, based on the preceding recommendation.

Refer to the Oracle WebLogic Server Documentation Library for details on creating a new domain and a new Managed Server.

Installation steps

Use the following steps to install the SPML LDAP Adapter on Oracle WebLogic Server 11g (version 10.3.2).

- Step 1, “Extract the ear file”
- Step 2, “Configure the SPML LDAP Adapter”
- Step 3, “Rebuild the ear file”
- Step 4, “Install the SPML LDAP Adapter”

Step 1 Extract the ear file

Use the following steps to extract the `ldap-spml-psp.ear` file.

1. Copy the `ldap-spml-psp.ear` file to a temporary location. This location is referred to as `<EAR_HOME>`.
2. Navigate to `<EAR_HOME>` and execute the following command:

   ```
   jar xvf ldap-spml-psp.ear
   ```

   The extract contains an archive named `ldap-spml-psp.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>/ldap-spml-psp.war
   ```

   The extract contains a directory named `WEB-INF`.

Step 2 Configure the SPML LDAP Adapter

Use the following properties files to configure the SPML LDAP Adapter. These files are located in the `war_home/WEB-INF/classes` directory.

```
ldap.properties
ldap-namespace.properties
ldap-user-profile.properties
```

Refer to “Configure the SPML LDAP Adapter” on page 10-2 for details about configuring each properties file.

Step 3 Rebuild the ear file

Use the following steps to rebuild the ear file with the modified configuration properties.

1. Execute the following command at from the `war_home` directory:

   ```
   jar cvf <EAR_HOME>/ldap-spml-psp.war WEB-INF/* index*
   ```

   The war file is rebuilt with the modified configurations.

2. Execute the following command from `<EAR_HOME>`:

   ```
   jar cvf ldap-spml-psp.ear *.war META-INF/*
   ```

   The ear file is rebuilt with the modified configurations.The rebuilt ear file is used for installation.

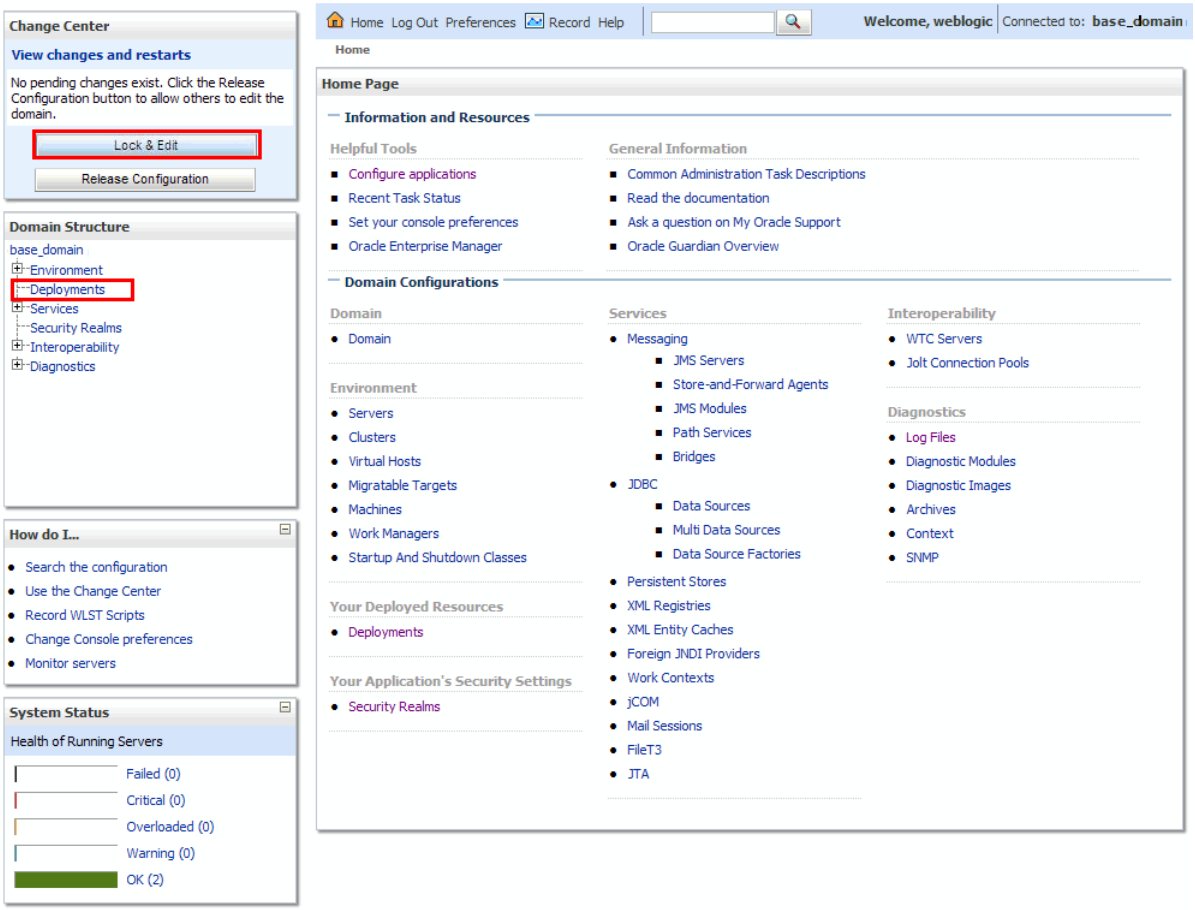
Step 4 Install the SPML LDAP Adapter

Use the following steps to install the SPML LDAP Adapter to the Oracle WebLogic Server.

1. Connect to the Oracle WebLogic Server Administration Console:

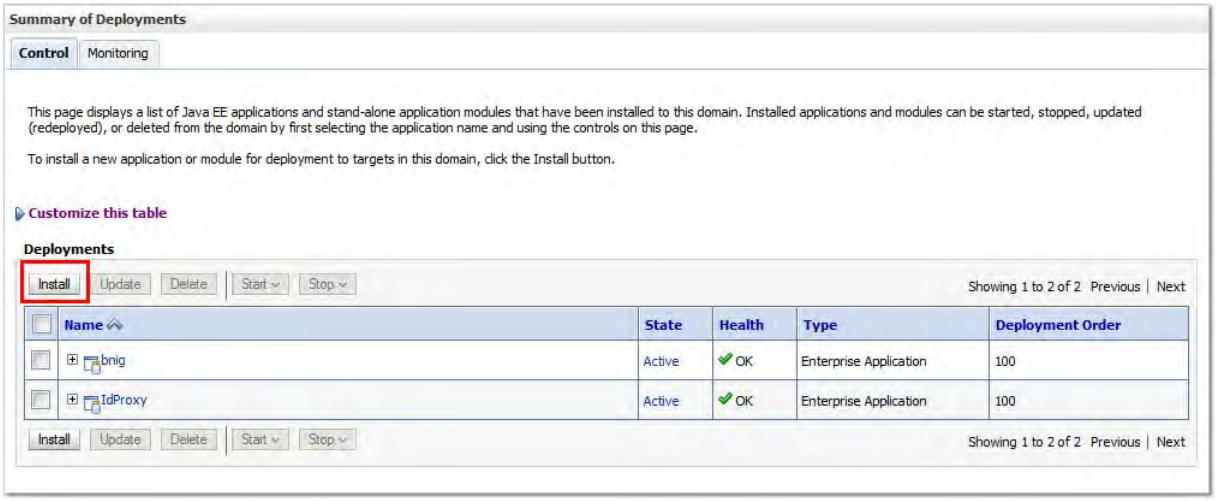
   ```
   http://<host>:<port>/console
   ```

   The Home Page is displayed.

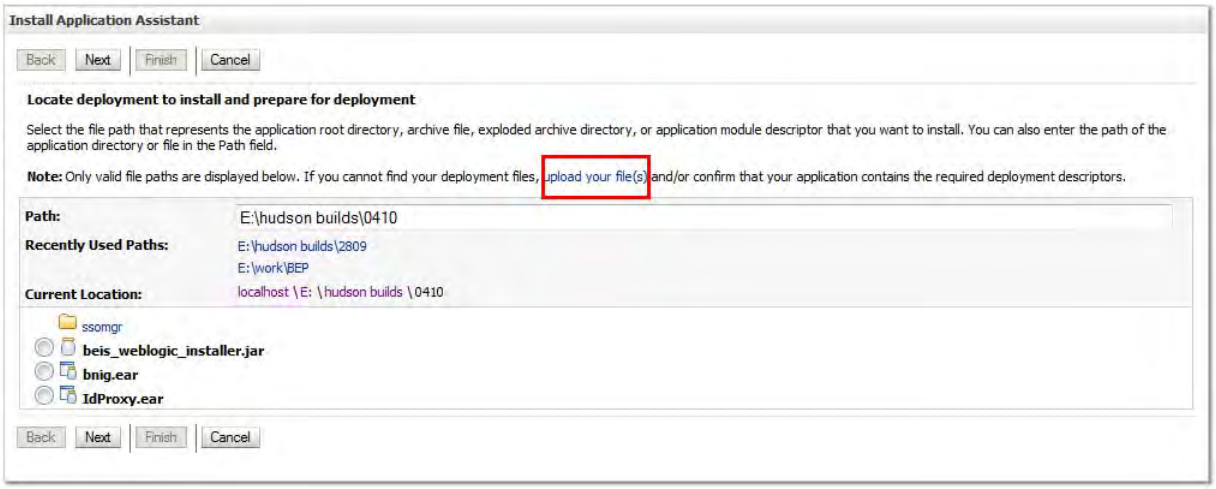


2. In the Change Center pane, click **Lock & Edit**.

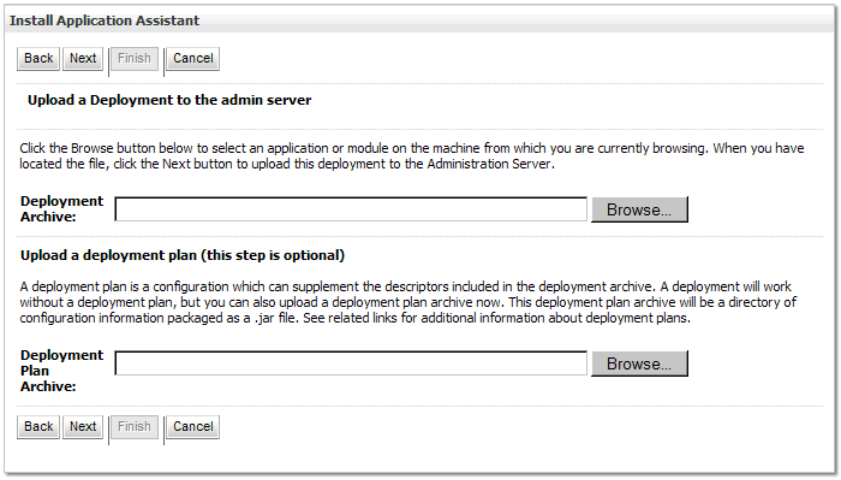
3. In the Domain Structure pane, click **Deployments**. The Summary of Deployments page is displayed.



4. Click **Install**. The Install Application Assistant page is displayed.



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

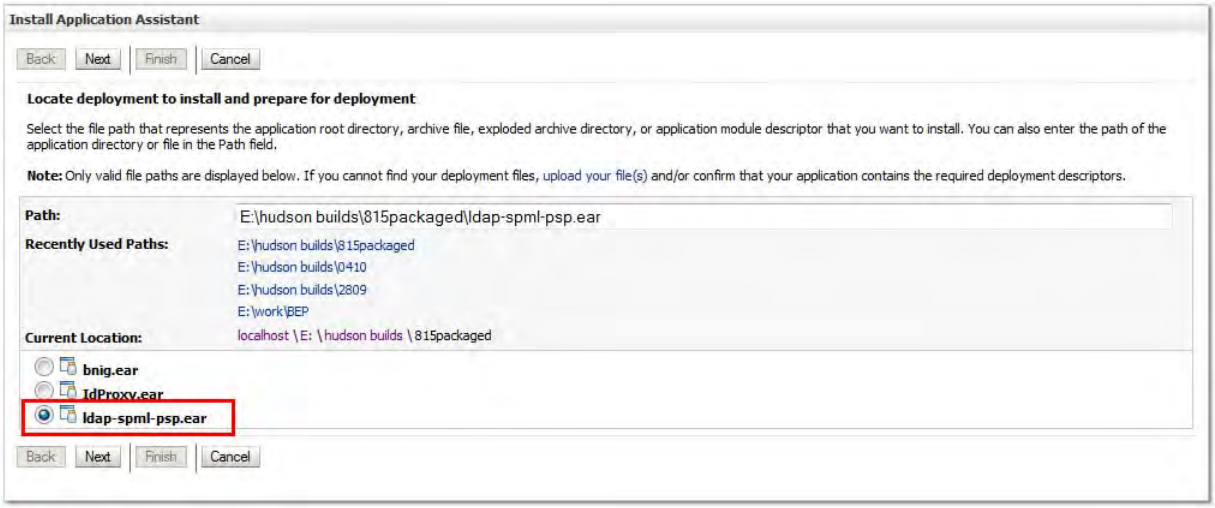


6. Select the file to be uploaded:

   6.1. In the **Deployment Archive** field, click **Browse** and navigate to the `ldap-spml-psp.ear` file.

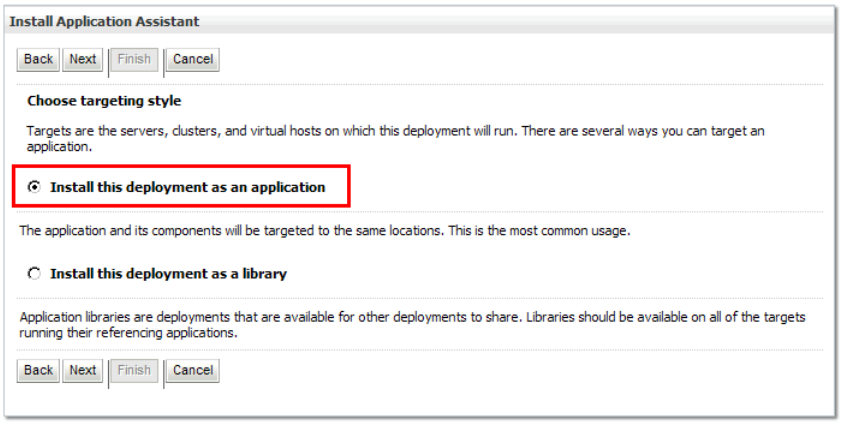
   6.2. Select the file and click **Open**.

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



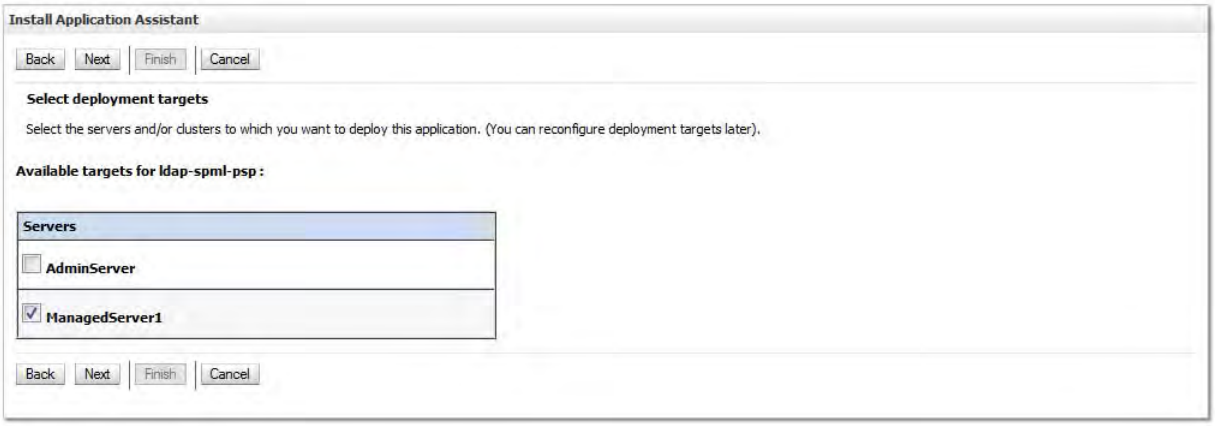
8. Select the `ldap-spml-psp.ear` file from the list.

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.

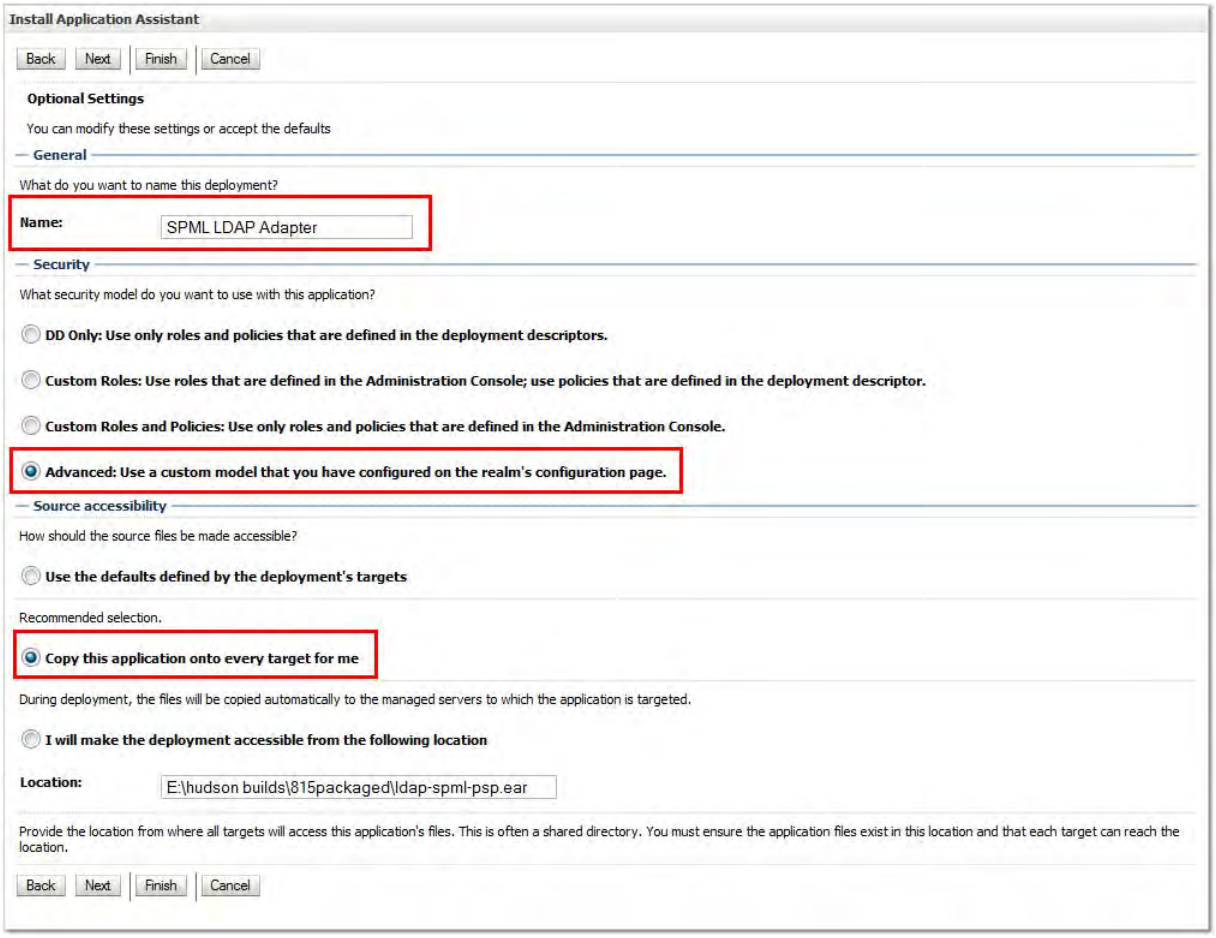


12. Select the server where the application should be deployed. (The application can be installed on an existing server.)

Note

SunGard Higher Education recommends deploying applications to a WebLogic Managed Server and not to the Administration Server. If you do not see the preceding page, you should check your WebLogic Server configuration to ensure that a Managed Server is available for deployment of applications. If a Managed Server is not available, the application will be deployed to the Administration Server, which is not a recommended configuration. For more information, consult the Oracle WebLogic Server Documentation Library. ■

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

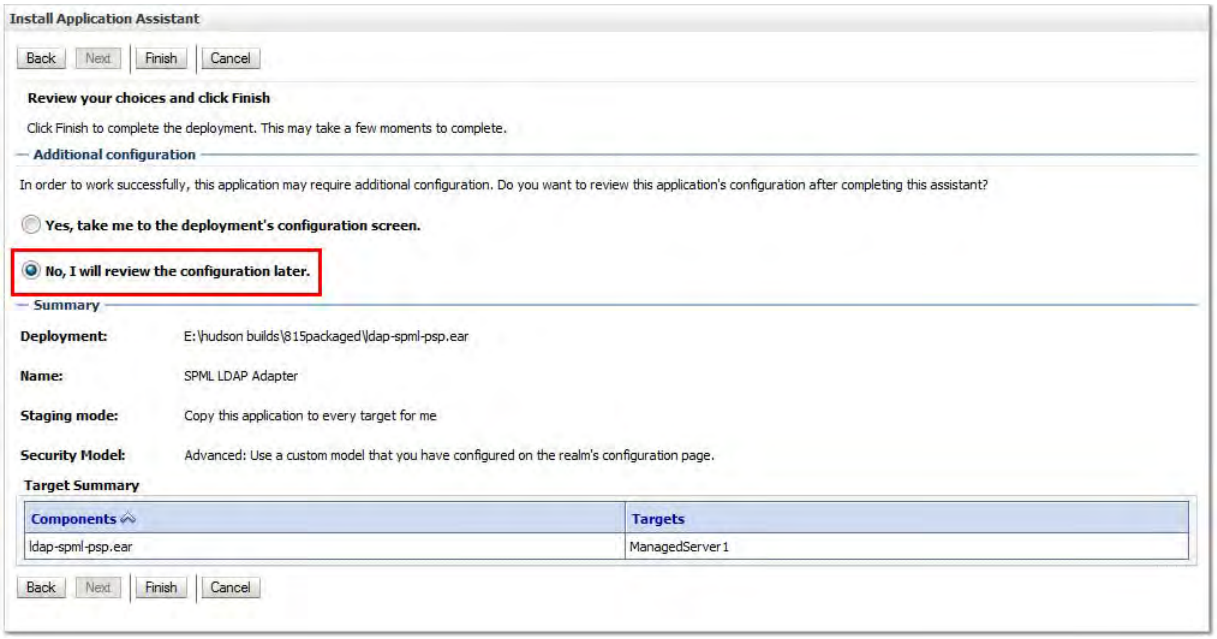


14. Enter a name for the application (for example, *SPML LDAP Adapter*) in the **Name** field.

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

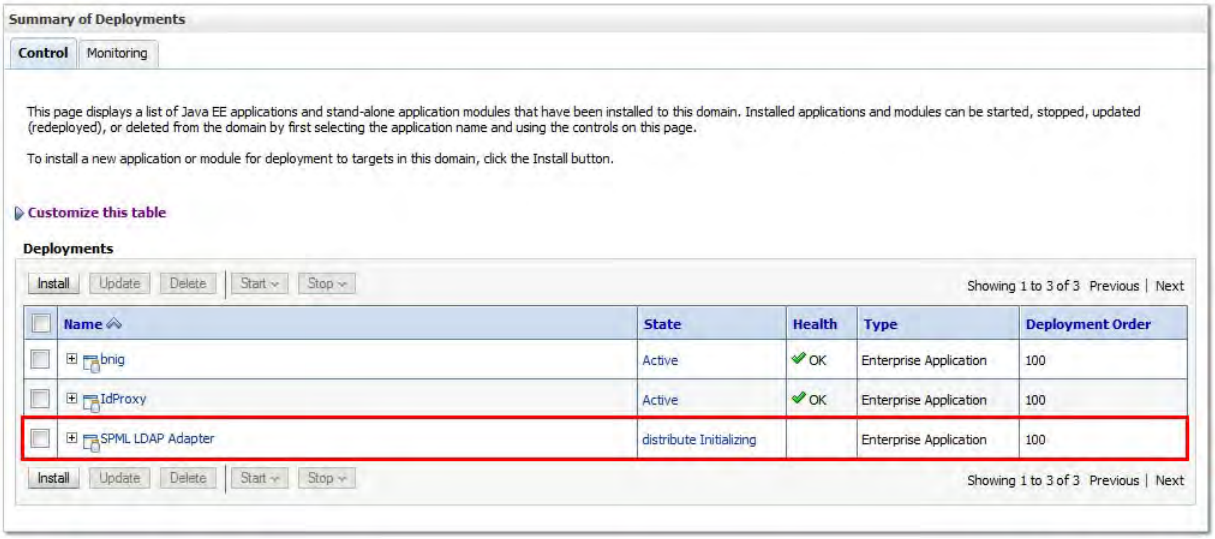
16. Select **Copy this application onto every target for me**.

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



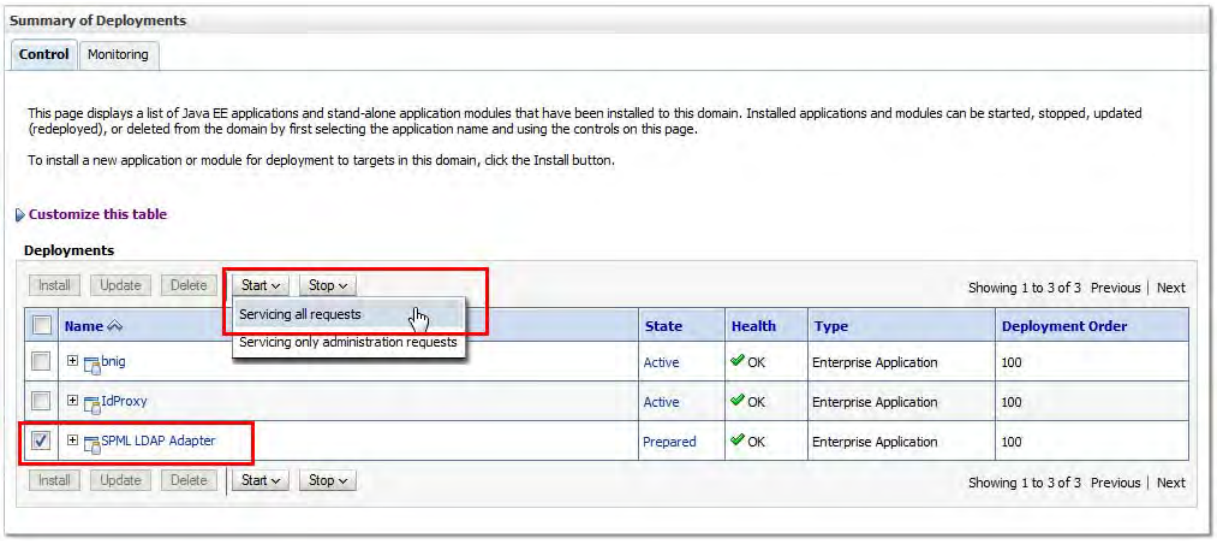
18. Select **No, I will review the configuration later**.

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



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

21. Start the newly deployed application as follows:



21.1. Select the newly deployed application.

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

21.3. Click **Yes**.

The SPML endpoint can be accessed with the following URL:

```
http://<host>:<port>/spml/xfire/psp/DocumentLiteral
```

11 SSO Manager

The SSO Manager acts as a single sign on gateway for Internet-native Banner® (INB) and Self-Service Banner (SSB), allowing these applications to participate in a claims-based authentication environment. The SSO Manager also provides services that other SunGard® Higher Education applications can use to facilitate claims-based authentication based on the UDCIdentifier.

This chapter gives instructions for implementing the SSO Manager and using it for development purposes.

Definitions

The following terms are important for understanding the SSO Manager:

- **Single sign on** (SSO) centralizes the process of authenticating the digital identity of a user. Once authenticated, a user can access multiple software systems without having to sign on to each system.
- **Claims-based authentication** authenticates a user based on claims, contained in a trusted token, about the user's identity. The token is issued and signed by a trusted entity that authenticates the user.

How the SSO Manager facilitates SSO

The following sections explain how the SSO Manager facilitates SSO for Self-Service Banner (SSB) and Internet-native Banner (INB).

SSO for Self-Service Banner (SSB)

The SSO Manager acts as a front-end to SSB, bypassing native SSB authentication. A central access manager handles authentication, protecting the SSB access URLs that are exposed by the SSO Manager. Once the central access manager authenticates the user, the SSO Manager collaborates with Banner Web Tailor to provide access to the user.

When using the SSO Manager, SSB is accessed through one of the following URLs:

CAS	`http(s)://<host>:<port>/ssomanager/c/SSB`
Third-party access manager	`http(s)://<host>:<port>/ssomanager/c/auth/SSB`

Note

You can still access Banner with the standard SSB URL. In this case, native SSB authentication (SPRIDEN ID and PIN), rather than SSO, is used to access Banner. ■

SSO components

The following components collaborate to accomplish SSO to Self-Service Banner:

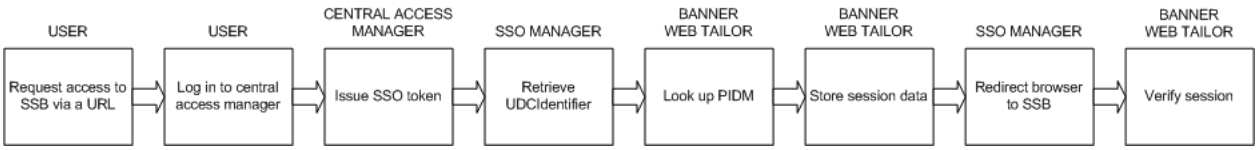
Component	Description
Central access manager	The central access manager for SSO can be the JA-SIG Central Authentication Service (CAS) or a third-party access manager. If CAS is used as the central access manager, the CAS attribute assertion features facilitate single sign on. Attributes that identify the user are retrieved via a proprietary validation service, /bannerValidate, or via the CAS /samlValidate service.
SSO Manager	This Web application acts as the SSO gateway for Banner, facilitating the following processes when starting the SSO session: • Retrieval of the user's unique identifier (UDCIdentifer) from the identity vault via /bannerValidate or /samlValidate • Proxy of SSO requests in an SSO environment that is administered by BEIS
Banner Web Tailor	Banner Web Tailor accepts the identity assertion from the SSO Manager, determines the user based on the assertion, and creates a session for the user.

Processing flow

The following processing occurs when using the SSO Manager for single sign on to Self-Service Banner.

Note

This processing flow does not address implementation-specific differences between CAS and third-party single sign on. ■



1. The user opens a Web browser and requests access to SSB through a protected SSB URL:

CAS	`http(s)://<host>:<port>/ssomanager/c/SSB`
Third-party access manager	`http(s)://<host>:<port>/ssomanager/c/auth/SSB`

2. Because the URL is protected by a central access manager, the user is redirected to the central access manager login page.
3. The user authenticates to the central access manager.
4. The central access manager performs the following:
   - **4.1.** Issues an SSO token to the user.
   - **4.2.** Forwards the request to the SSO Manager.
5. The SSO Manager performs the following:
   - **5.1.** Uses the SSO token to retrieve the required user information (UDCIdentifier).
   - **5.2.** Stores the UDCIdentifier in a cookie that it creates in the browser session.
   - **5.3.** Calls Banner Web Tailor to create a session for the user.
6. Banner Web Tailor performs the following:
   - **6.1.** Uses the UDCIdentifier to look up the associated Banner PIDM.
   - **6.2.** Inserts or updates a row in the Web Tailor Web SessionID Table (TWGBWSES), indicating to Banner Web Tailor security that a session was started and properly authenticated.
   - **6.3.** Returns control to the SSO Manager.
7. The SSO Manager redirects the user's browser to SSB.

8. Banner Web Tailor verifies the following:

   - A row exists in the Web Tailor SessionID Table (TWGBWSES) for the PIDM associated with the UDCIdentifier that is stored in the cookie created by the SSO Manager.
   - The last access date is valid.

   These verifications ensure that the session was started after authentication and prevent hackers from setting cookies with a non-authenticated UDCIdentifier.

SSO for Internet-native Banner (INB)

The SSO Manager acts as a front-end to INB, bypassing native Banner authentication. A central access manager handles authentication, protecting the INB access URLs that are exposed by the SSO Manager. Once the central access manager authenticates the user, the SSO Manager collaborates with Oracle Forms runtime components to provide access to Banner.

When using the SSO Manager, INB is accessed through one of the following URLs:

CAS	`http(s)://<host>:<port>/ssomanager/c/INB`
Third-party access manager	`http(s)://<host>:<port>/ssomanager/c/auth/INB`

Note

You can still access Banner with the standard INB URL. In this case, native INB authentication (Oracle Forms user ID and password), rather than SSO, is used to access Banner.

SSO components

The following components collaborate to accomplish SSO to Internet-native Banner:

Component	Description
Central access manager	The central access manager for SSO can be the JA-SIG Central Authentication Service (CAS) or a third-party access manager. If CAS is used as the central access manager, the CAS attribute assertion features facilitate single sign on. Attributes that identify the user are retrieved via a proprietary validation service, /bannerValidate, or via the CAS /samlValidate service.

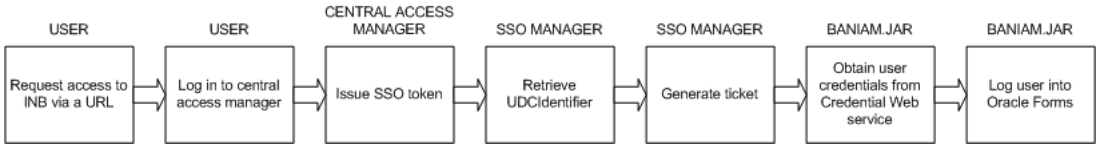
Component	Description
SSO Manager	This Web application acts as the SSO gateway for Banner, facilitating the following processes when starting an SSO session: • Retrieval of the user's unique identifier (UDCIdentifer) from the identity vault • Creation of the ticket used to retrieve the user's Oracle credentials from the Credential Web service • Exposure of a Credential Web service for storing and retrieving application-specific credentials
baniam.jar	Internet-native Banner uses this jar file to obtain the user's Oracle credentials from the Credential Web service. Refer to "Configure the Oracle Forms server for Internet-native Banner SSO" on page 11-74 and "Configure baniam.jar for Internet-native Banner SSO" on page 11-82 for more information.

Processing flow

The following processing occurs when using the SSO Manager for single sign on to Internet-native Banner.

Note

This processing flow does not address implementation-specific differences between CAS and third-party single sign on. ■



1. The user opens a Web browser and requests access to INB through a protected INB URL:

CAS	`http(s)://<host>:<port>/ssomanager/c/INB`
Third-party access manager	`http(s)://<host>:<port>/ssomanager/c/auth/INB`

2. Because the URL is protected by a central access manager, the user is redirected to the central access manager login page.

3. The user authenticates to the central access manager.

4. The central access manager performs the following:

    4.1. Issues an SSO token to the user.

    4.2. Forwards the request to the SSO Manager.

5. The SSO Manager performs the following:

    5.1. Uses the SSO token to obtain the required user information (UDCIdentifier).

    5.2. Creates a request scope INB ticket.

    5.3. Forwards the user request to the Oracle Forms application (baniam.jar).

6. baniam.jar communicates with the Credential Web service, exposed by the SSO Manager, to obtain the user credentials that are required to log the user into Oracle Forms and start the user session.

Note

The SSO Manager can be configured to automatically generate a password if the Credential Web service does not know the user credentials.

The INB ticket is never reused. The ticket that the SSO Manager forwards is destroyed as soon as the Oracle Forms application (baniam.jar) uses it to request credentials from the Credential Web service.

The INB ticket is created with an expiration time. If the Oracle Forms application never uses the INB ticket, the ticket is destroyed when it expires.

Implementation of the SSO Manager

The SSO Manager can be installed on an existing Oracle Application Server or on an existing Oracle WebLogic Server, noting the following restrictions:

- On an Oracle Application Server, the SSO Manager can be deployed into a new or existing OC4J instance on an existing Oracle Application Server. It should not be deployed with any of the following BEIS components: Identity Data Export Utilities, Banner Identity Gateway, or Enterprise Identity Proxy Services.
- On an Oracle WebLogic Server, the SSO Manager can be deployed on a new or existing Managed Server in a Basic Domain. It should not be deployed to a server that was configured using the Classic Domain template that supports Oracle Forms and Reports.

The recommended configuration is to deploy SSO Manager in a location (OC4J instance or Managed Server) that is separate from where the Banner Identity Gateway and

Enterprise Identity Proxy Services are deployed. This allows for separate administration of SSO and provisioning

This section describes the prerequisites and the tasks that are used to implement the SSO Manager:

- "Establish the environment" on page 11-7
- "Configure the domain security model" on page 11-8
- "Configure the authentication provider" on page 11-11
- "Run the configuration utility" on page 11-19
- "Complete the installation on Oracle Application Server" on page 11-31
  -or-
  "Complete the installation on Oracle WebLogic Server 11g" on page 11-48
- "Migrate credentials (optional)" on page 11-73
- "Configure the supporting components" on page 11-73
- "Verify the configuration" on page 11-91

Prerequisites

Implementation of the SSO Manager has the following prerequisites:

- JDK 1.6 must be installed.
- The environment variable `'JAVA_HOME'` must be set.
- A database user account that has privileges to create users must be identified or created. This account is needed when you run the SSO Manager installation utility.

Establish the environment

Use the following steps to establish the environment for the SSO Manager.

1. Configure the central access manager that authenticates users for SSO. The central access manager can be the JA-SIG Central Authentication Service (CAS) or a third-party access manager:
   - For JA-SIG CAS, use the instructions in Appendix D, "CAS Installation and Configuration" to configure the CAS server.
   - For a third-party access manager, consult your vendor's documentation to configure the following:
     - SSO Manager URLs for SSB and INB as protected resources in the third-party access manager

- Retrieval of the UDCIdentifier from the third-party access manager's identity vault
- Inclusion of the UDCIdentifier in redirects to the SSO Manager via a cookie, header, or parameter

2. (Optional) If you plan to deploy the SSO Manager into a separate application server container, use the application server administration console to create a new OC4J instance (Oracle Application Server) or a new Managed Server (Oracle WebLogic Server). Refer to Oracle installation and configuration documentation for details.

3. If you are using CAS, register the SSO Manager deployment with the CAS server the same way that you register any other CAS application. See your CAS server documentation for more details.

Configure the domain security model

Note

This step only applies to implementation on Oracle WebLogic Server 11g. Skip this step if you are implementing the SSO Manager on Oracle Application Server. ■

The Oracle WebLogic Server must be configured to use the *Advanced* security model instead of the default *DD only* option. This step pertains to the realm configuration. It applies to the entire domain. (Although you can create a totally new realm for the domain, only one realm can be active at a time for the entire domain.)

Note

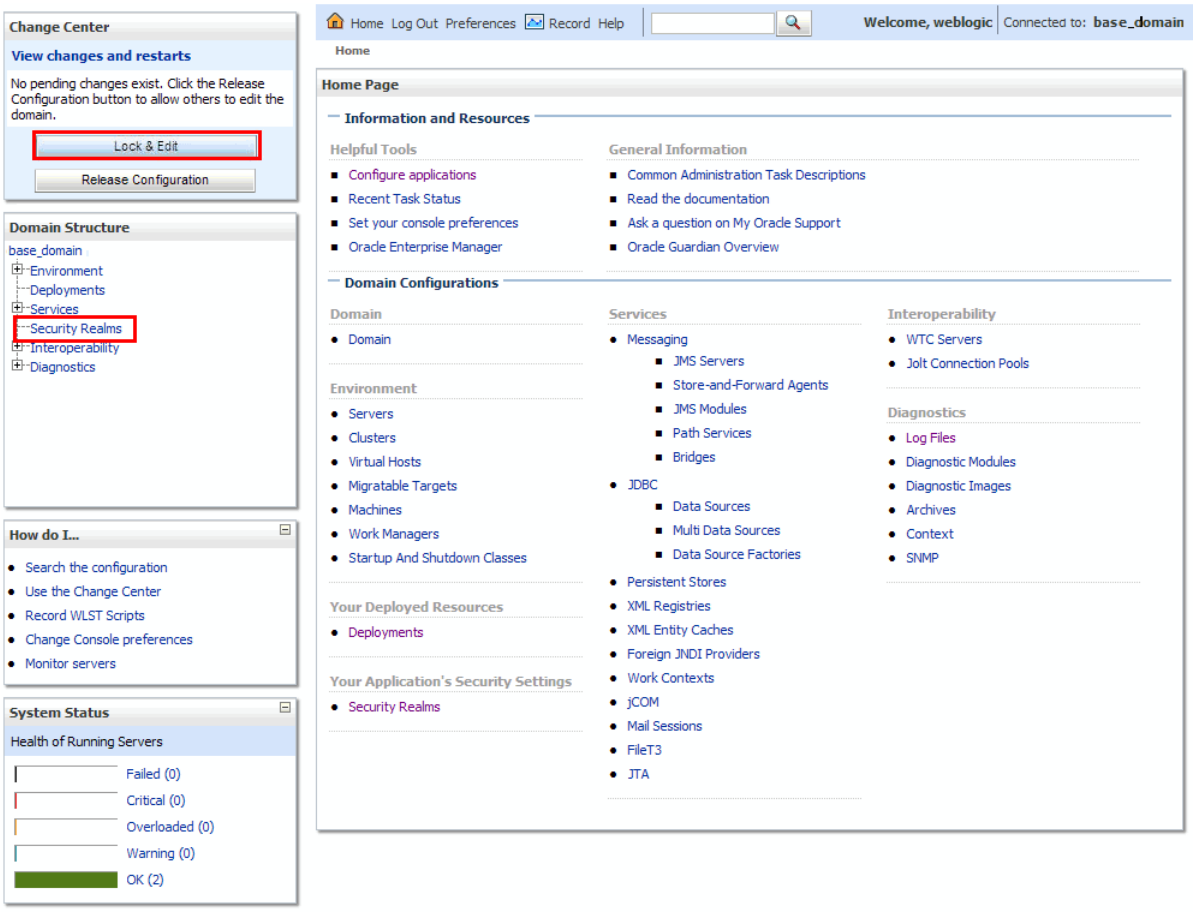
After you run the configuration utility, you can set the security model back to the default *DD only* option. ■

Use the following steps to configure the domain security model.

1. Connect to the Oracle WebLogic Server Administration Console:

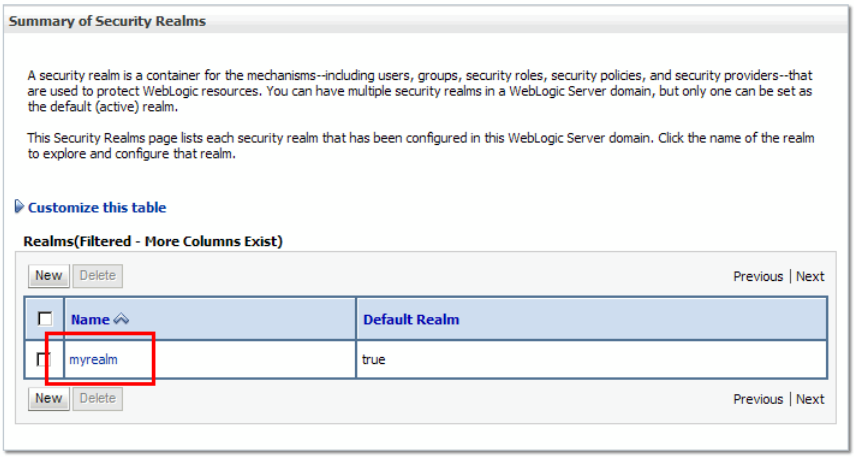
   ```
   http://<host>:<port>/console
   ```

The Home Page is displayed.

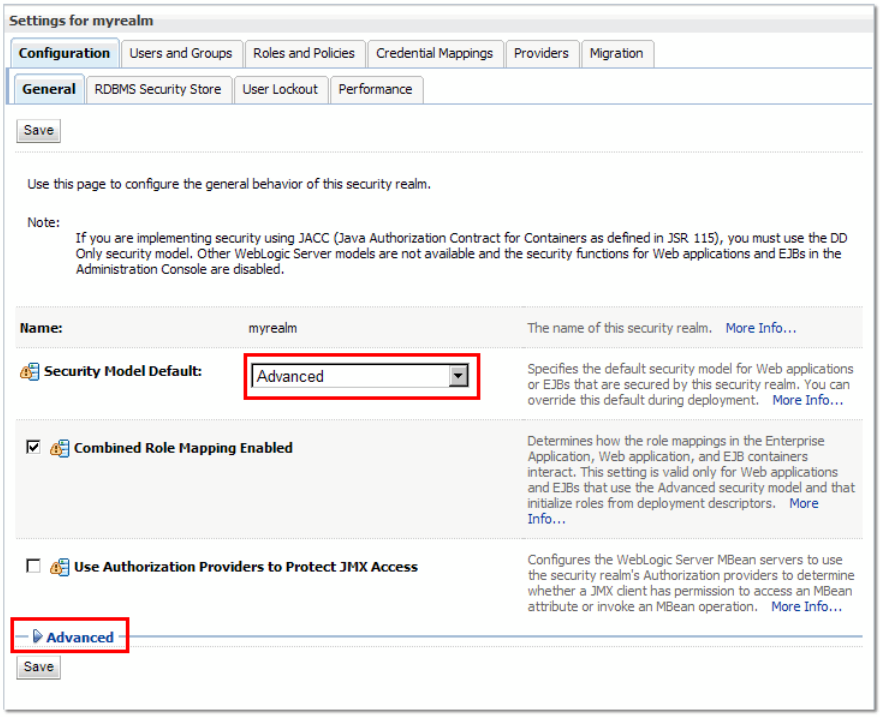


2. In the Change Center pane, click **Lock & Edit**.

3. In the Domain Structure pane, click **Security Realms**. The Summary of Security Realms page is displayed.

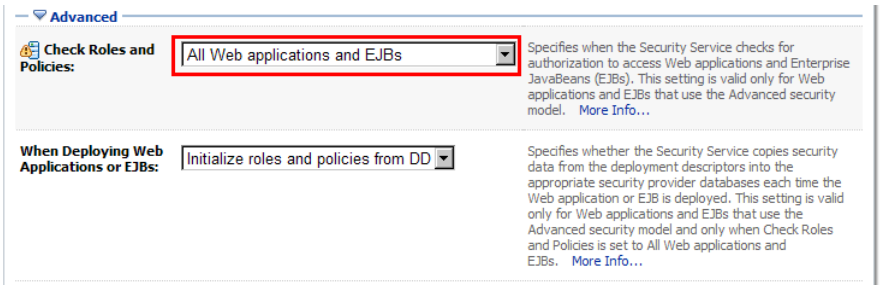


4. Click **myrealm**. The Settings page is displayed.



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

6. 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.

Configure the authentication provider

Note

This step only applies to implementation on Oracle WebLogic Server 11g. Skip this step if you are implementing the SSO Manager on Oracle Application Server.

An authentication provider must be configured in Oracle WebLogic Server 11g to allow for basic authentication against the Web services that the SSO Manager exposes. The authentication provider is set via a JAAS configuration file. The Oracle WebLogic Managed Server where the SSO Manager will be deployed must be configured to load the configuration file on startup. Use the following steps to configure the authentication provider.

1. If a JAAS configuration file already exists for the Oracle WebLogic domain where the SSO Manager will be deployed, skip to step 3.

   If a JAAS configuration file does not exist for the Oracle WebLogic domain where the SSO Manager will be deployed, use a text editor to create the `jaas.config` file with the following content:

   ```
   myrealm {
      weblogic.security.auth.login.UsernamePasswordLoginModule
      REQUIRED;
   };
   ```

2. Save `jaas.config` in the following location:

   ```
   <WebLogic Home>/user_projects/domains/<your domain dir>/config
   /security
   ```

   where `<WebLogic Home>` is the base directory for the Oracle WebLogic software packages and configuration files, and `<your domain dir>` is the domain where the SSO Manager will be deployed.

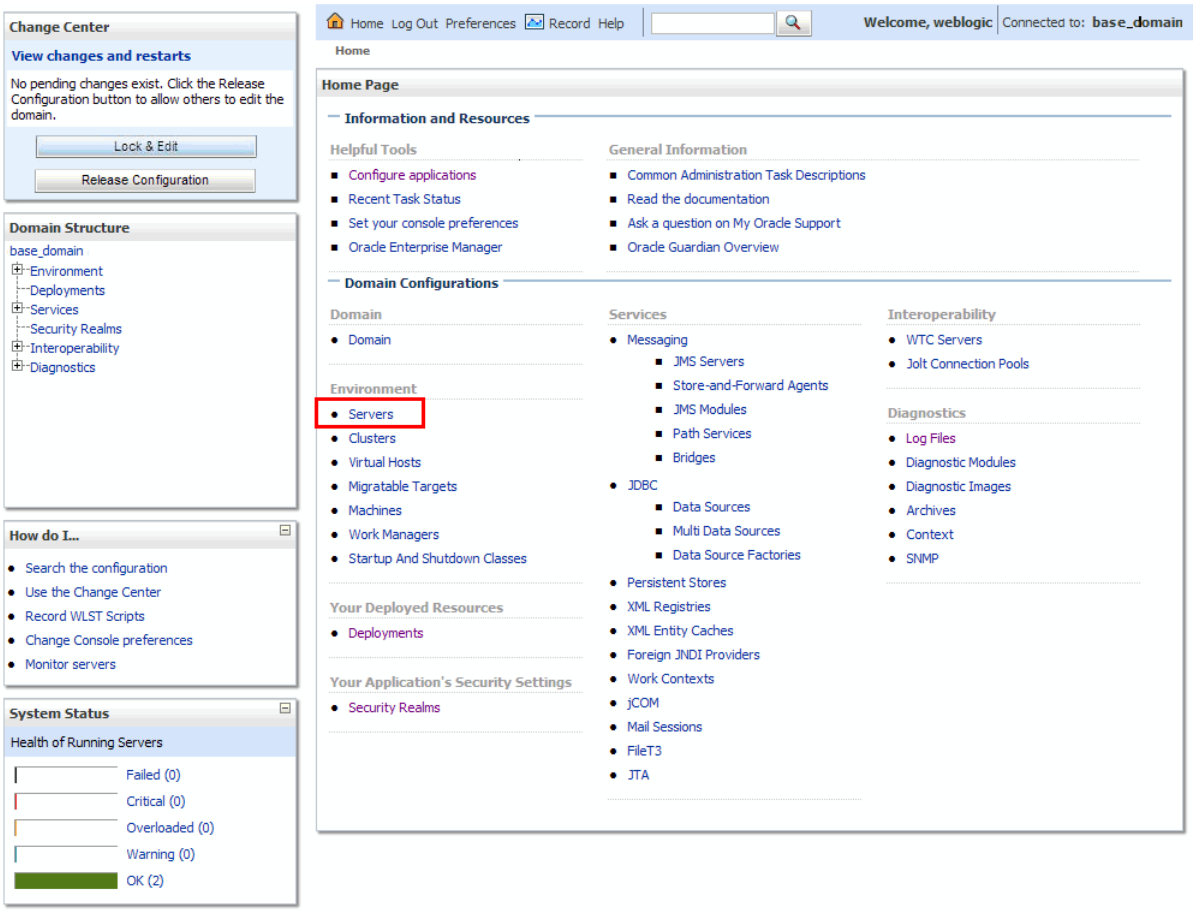
3. Configure the Managed Server to use the authentication provider.

   There are two ways to configure the Managed Server, depending on how you want to start the Managed Server. Use option 1 (page 11-12) if the Managed Server will be started by using the Oracle WebLogic Server Administration Console. Use option 2 (page 11-16) if the Managed Server will be started by running a script.

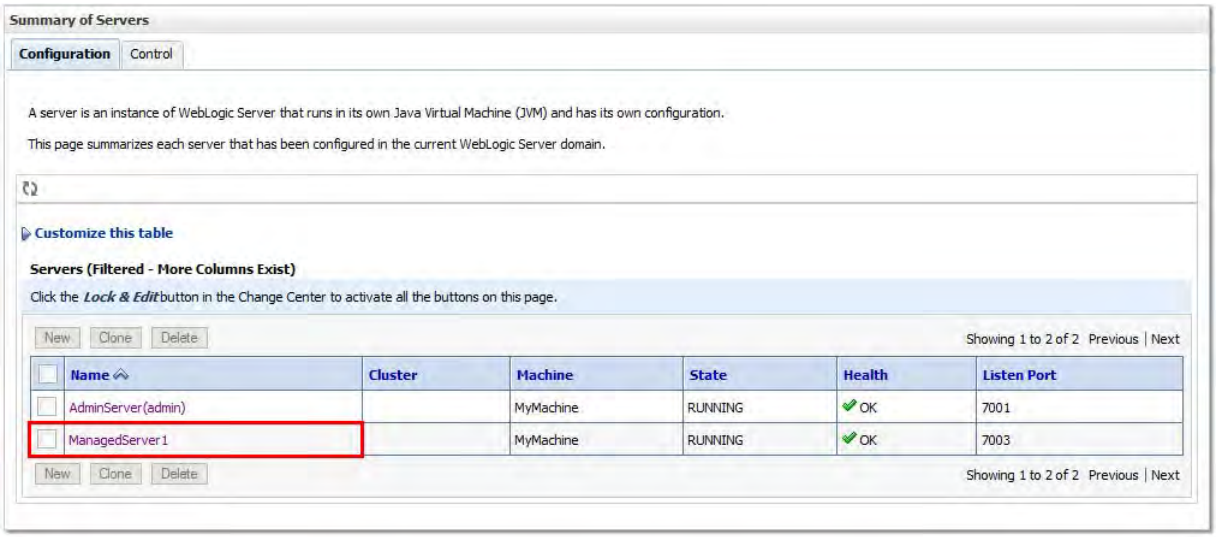
Option 1 - If you are using the administration console to start the Managed Server

With this option, the location of the JAAS configuration file is set as an argument on the Server Start tab of the specific Managed Server. The location of the JAAS configuration file applies only to that specific Managed Server.

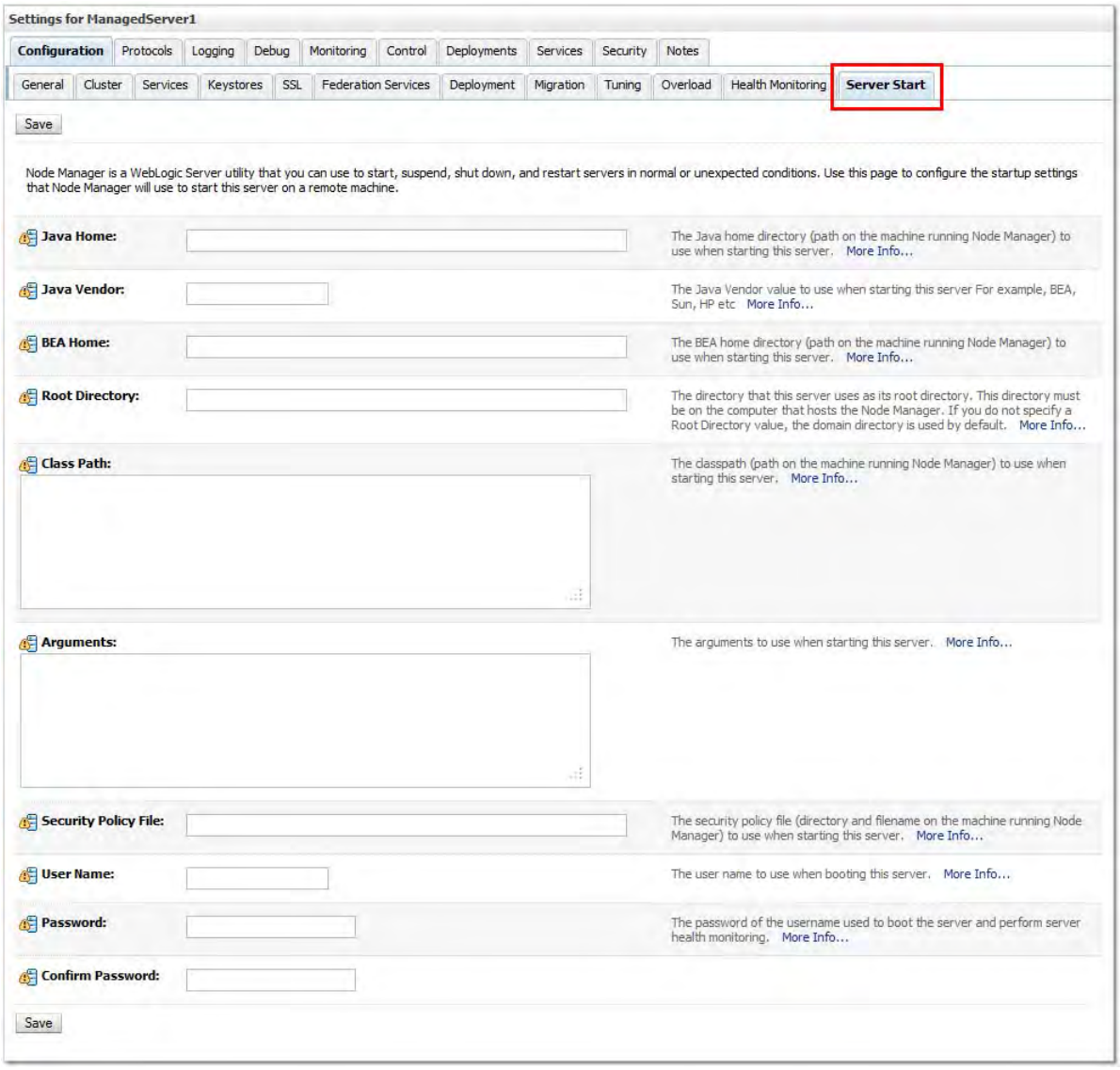
3.1. Connect to the Oracle WebLogic Server Administration Console for the domain where the SSO Manager will be deployed. The Home Page is displayed.



3.2. Click **Servers**. The Summary of Servers page is displayed.



3.3. Click the name of the server where the SSO Manager will be deployed. The Settings page is displayed.



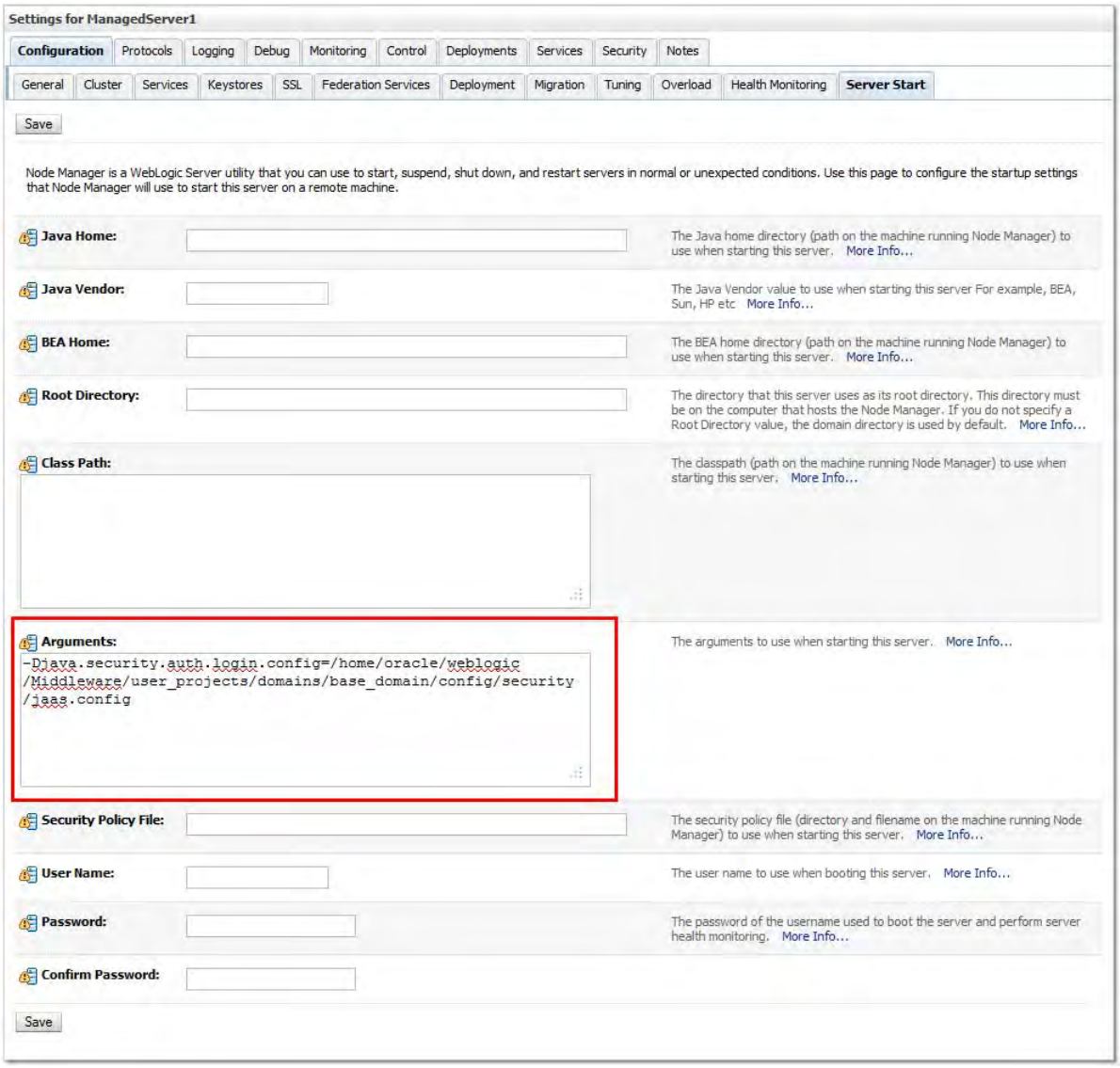
3.4. Click the **Server Start** tab.

3.5. Click **Lock & Edit** in the Change Center pane.

3.6. In the **Arguments** field, enter the full path to the `jaas.config` file, including the file name:

```
-Djava.security.auth.login.config=<WebLogic Home>/
user_projects/domains/<your domain dir>/config/security/
jaas.config
```

where `<WebLogic Home>` is the base directory for all Oracle WebLogic software packages and configuration files, and `<your domain dir>` is the name of the domain where the SSO Manager will be deployed.



3.7. Click **Save.**

3.8. Click **Activate Changes** in the Change Center pane.

Option 2 - If you are using a script to start the Managed Server

Use this option if the Managed Server will be started by running the `startManagedWebLogic.sh` (or `.cmd`) script. A `JAVA_OPTIONS` statement must be added to the `setDomainEnv.sh` (or `.cmd`) script. The location of the JAAS configuration file applies to the entire domain, including the Admin Server and all Managed Servers.

Use the following steps to update the script for Windows.

3.1. Open the `setDomainEnv.cmd` file located under `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

3.2. Search for the last occurrence of the following text:

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
```

3.3. Add the following in the line preceding the line identified in step 3.2.

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Djava.security.auth.login.config=
<domain home>\config\security\jaas.config
```

Use the following steps to update the script for Linux/Unix.

3.1. Open the `setDomainEnv.sh` file located under `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

3.2. Search for the last occurrence of the following text:

```
JAVA_OPTIONS="${JAVA_OPTIONS}"
```

3.3. Add the following in the line preceding the line identified in step 3.2.

```
JAVA_OPTIONS="${JAVA_OPTIONS}
-Djava.security.auth.login.config=
<domain home>/config/security/jaas.config"
```

Note

There is a space between the closing brace and the dash (that is, {JAVA_OPTIONS}**[space]**-Djava). ■

4. Restart the appropriate server(s).

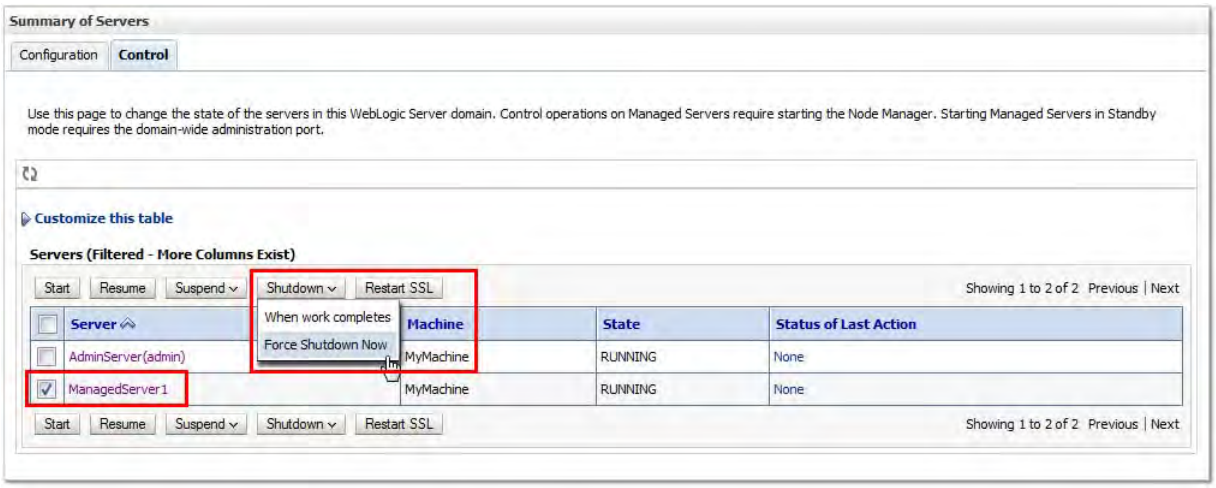
   There are two ways to restart the server(s). Use option 1 (page 11-17) if a single Managed Server was configured. Use option 2 (page 11-19) if all servers in the domain were configured.

Option 1 - If a single Managed Server was configured

Use this option if a single Managed Server was configured. Only that server needs to be restarted.

4.1. Navigate to the Summary of Servers page.

4.2. Click the **Control** tab.

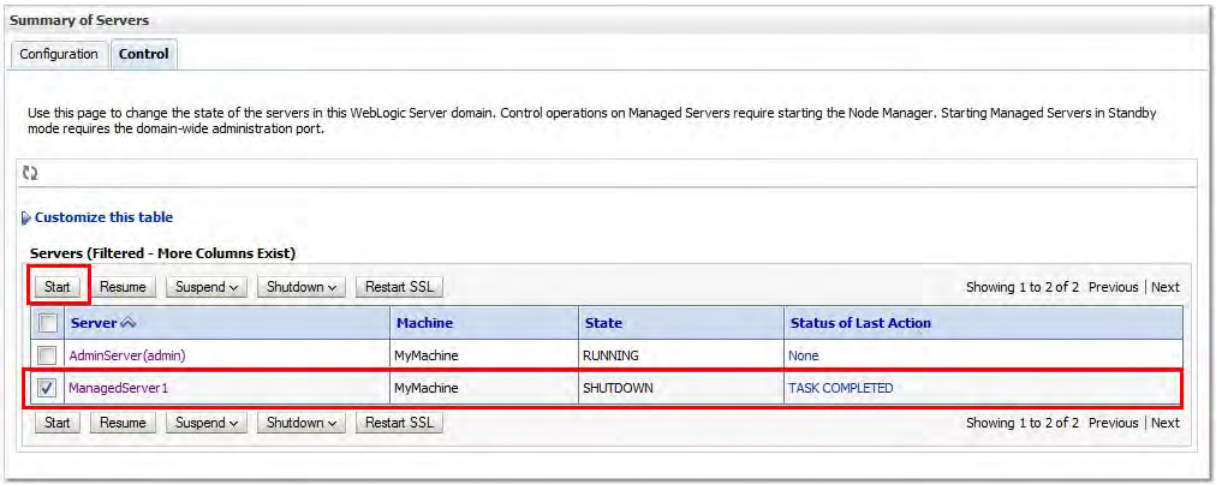


4.3. Select the Managed Server where the configuration changes were made.

4.4. Click **Shutdown** -> **Force Shutdown Now**.

4.5. Confirm the selection.

4.6. Wait for the server to enter a *SHUTDOWN* state.

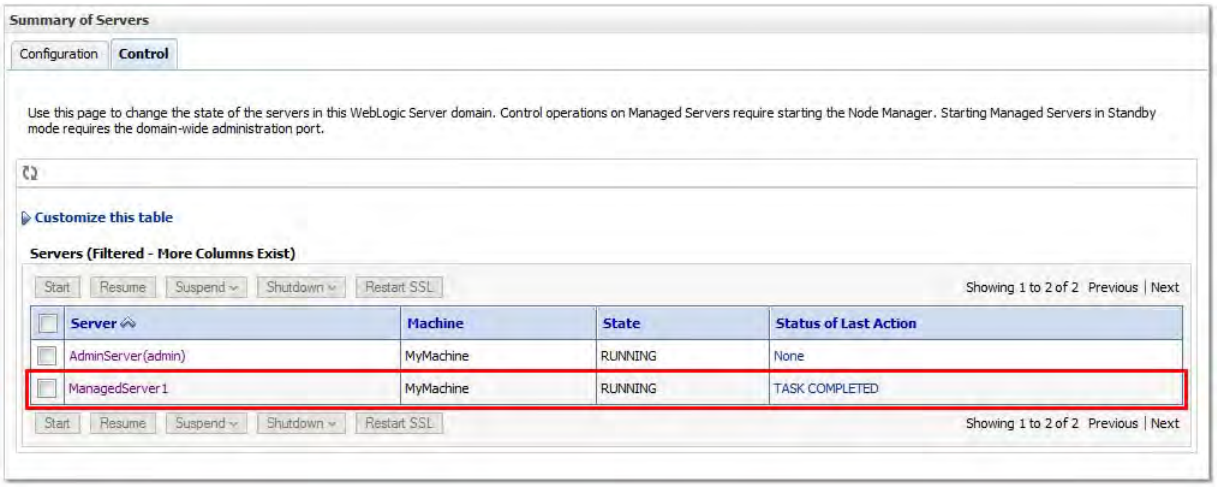


4.7. Select the same Managed Server.

4.8. Click **Start**.

4.9. Confirm the selection.

4.10. Wait for the server to enter a *RUNNING* state.



Option 2 - If all servers in the domain were configured

Use this option if all servers in the domain were configured. Only the Managed Server needs to be restarted.

Use the following steps to restart the server for Windows.

4.1. Navigate to `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

4.2. Stop the server by running the following script:

```
stopManagedWebLogic.cmd <ServerName>
```

Note

There is a space between the command and <ServerName>; that is, stopManagedWeblogic.cmd**[space]**<ServerName>.

Example:

```
stopManagedWebLogic.cmd ManagedServer1
```

4.3. Start the server by running the following script:

```
startManagedWebLogic.cmd <ServerName>
```

Use the following steps to restart the server for Linux/Unix.

4.1. Navigate to `<WebLogic Home>/user_projects/domains/<your domain dir>/bin`.

4.2. Stop the server by running the following script:

```
./stopManagedWebLogic.sh <ServerName>
```

Note

There is a space between the command and <ServerName>; that is, ./stopManagedWeblogic.sh**[space]**<ServerName>.

Example:

```
./stopManagedWebLogic.sh ManagedServer1
```

4.3. Start the server by running the following script:

```
./startManagedWebLogic.sh <ServerName>
```

Run the configuration utility

The configuration utility must be run to configure the SSO Manager ear file. In addition, the utility can optionally automate the following tasks:

- Create the database schema and database objects.
- Create the data sources.
- Create security users for logging into the application (Oracle WebLogic 11g only).
- Deploy the SSO Manager into the application server.

Any components that are not created with the utility can be created manually, if needed. See "Complete the installation on Oracle Application Server" on page 11-31 or "Complete the installation on Oracle WebLogic Server 11g" on page 11-48 for the manual steps.

The utility simplifies installation and is useful for deployments where servers or the database are *not* clustered. If you plan to deploy the SSO Manager in a cluster of Managed Servers or against an Oracle RAC environment, the corresponding steps in the installation process (creating data sources and deploying the application) should be done manually, *not* with the utility.

When the utility is run on a server with a windowing environment, a graphical user interface (GUI) is presented that groups related options on pages. When the utility is run on a server without a windowing environment, each option is displayed in command-line mode with a default value. You can accept the default value or enter another value.

The following archive files are provided for running the configuration utility:

- `sso-manager-oc4j-installer.jar` is used for installation on Oracle Application Server 10.1.3.4/5.
- `sso-manager-weblogic-installer.jar` is used for installation on Oracle WebLogic Server 11g.

Use the following steps to run the configuration script. These instructions are based on the GUI mode. Configuration options are identical if you are using the command-line mode.

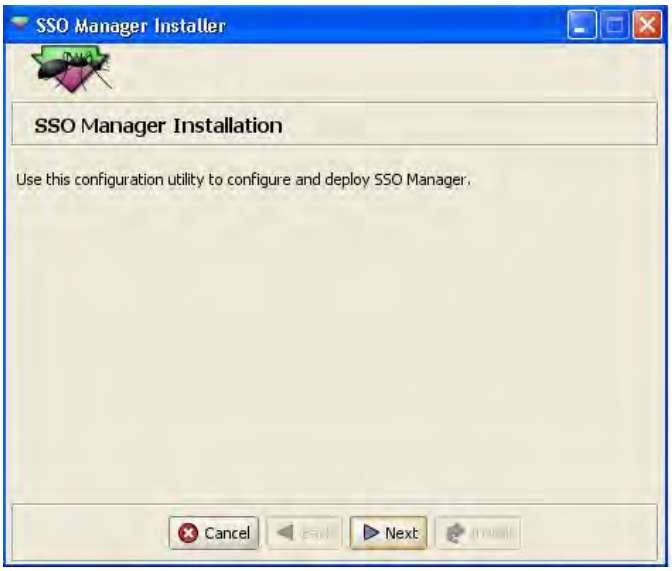
1. Open a command prompt and navigate to the appropriate archive file:

   `sso-manager-oc4j-installer.jar`
   or
   `sso-manager-weblogic-installer.jar` archive file.

2. Run the appropriate command:

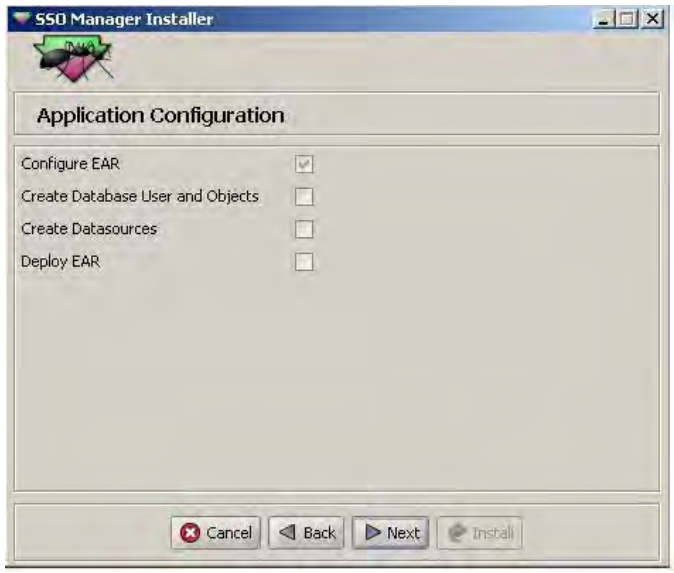
   ```
   java -jar sso-manager-oc4j-installer.jar
   or
   java -jar sso-manager-weblogic-installer.jar
   ```

The SSO Manager Installation window is displayed.



3. Click **Next**. The Application Configuration window is displayed.

OC4J version:



Oracle WebLogic 11g version:

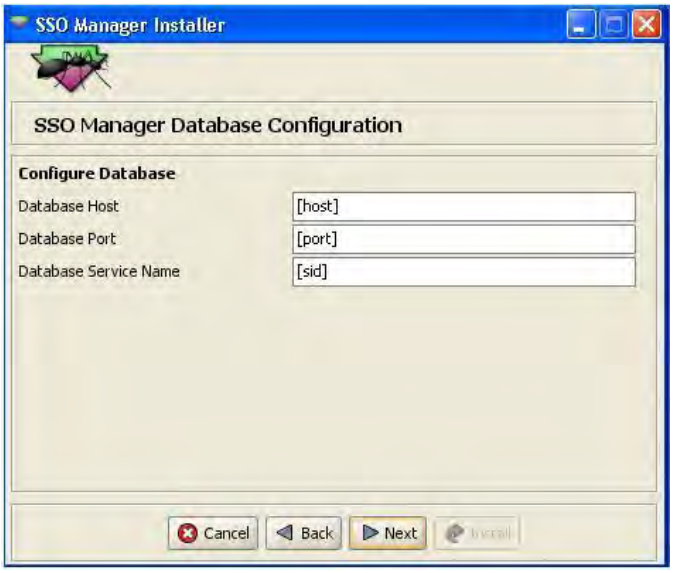
4. Select the components that you wish to configure:

Configure EAR	This check box is always selected and cannot be changed. The utility always configures the SSO Manager ear file.
Create Database User and Objects	The SSO Manager uses a separate schema to store configuration details, credentials, and ticket-related details. Select this check box to create the database user and objects (tables) associated with the SSO Manager schema. If the SSO Manager schema already exists, it will be dropped and re-created.
Create Datasources	Select this check box to create data source definitions on the application server. These data source definitions provide connectivity to the database(s) used by the SSO Manager. The SSO Manager needs a connection to the database where the configuration details and the credential vault are stored. The SSO Manager also needs connection to the Banner database for runtime purposes. These two databases can be the same or different. **Note:** Do not select **Create Datasources** if you previously created data sources manually. The installer cannot recreate data sources that were created manually.
Create Security Users	Select this check box to create a user who is authorized to access the SSO Manager user interface. **Note:** This option is available on Oracle WebLogic 11g only. For OC4J, you cannot use the utility to create security users; you must create them manually.
Deploy EAR	Select this check box to deploy the SSO Manager ear file to the application server.

The **Configure EAR** check box is always selected and cannot be changed. You can optionally choose to configure any other component(s). Those components that are not created with the utility can be created manually, if needed. See "Complete the installation on Oracle Application Server" on page 11-31 or "Complete the installation on Oracle WebLogic Server 11g" on page 11-48 for the manual configuration steps.

If you select a check box on this window, the associated configuration window will be displayed in the following steps. If you do not select a check box, the associated configuration window will be skipped.

5. Click **Next**. The SSO Manager Database Configuration window is displayed.



6. Enter the following information:

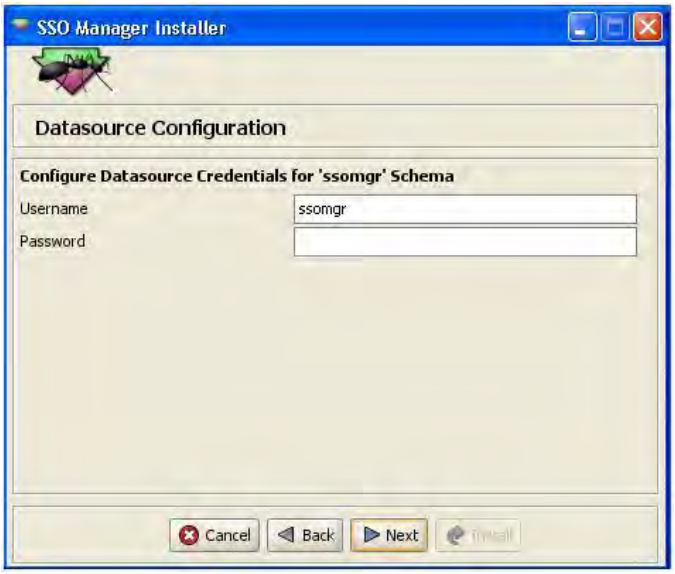
Database Host	Database host name
Database Port	Database port number
Database Service Name	Service name that identifies the database

7. Click **Next**. The SSO Manager Database and User Configuration window is displayed.

8. Enter the following information:

Admin Username	Oracle account that has privileges to create user accounts. The SSO Manager installer uses this Oracle account to create the database account and schema that the SSO Manager needs to function. No script is provided with BEIS to create an Oracle account that has privileges to create users. You must create an account or use an existing account.
Admin Password	Password for the administration user account
Tablespace Name	Tablespace name for the database schema (for example, *ssomgr_tbls*)
Data Filename	Name of the data file with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0./dbs/ssomgr.dbf*)
Database Username	Name of the database schema for the SSO Manager tables (for example, *ssomgr*). The installer will create this schema and user account in the database.
Database Password	Password for the database schema

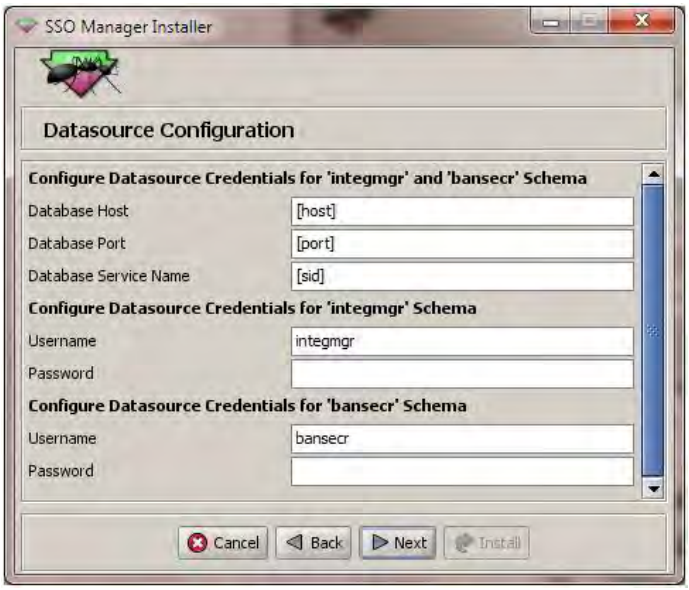
9. Click **Next**. The following Datasource Configuration window is displayed.



10. Enter the following information to create a data source, on the application server, that the SSO Manager will use to connect to the SSO Manager schema:

Username	Name of the user account associated with the data source that is used to connect to the SSO Manager schema. This should be the same as the schema name (for example, *ssomgr*).
Password	Password for the account. This should be the same as the database schema password.

11. Click **Next**. The following Datasource Configuration window is displayed.



12. Enter the following information to create data sources, on the application server, that the SSO Manager will use to connect to the Banner database schemas. These schemas must exist on the same database.

Database Host	Database host name
Database Port	Database port number
Database Service Name	Service name that identifies the database
integmgr Username	Normally *integmgr*

integmgr Password	Password for the `integmgr` account
bansecr Username	Normally *bansecr*
bansecr Password	Password for the `bansecr` account

13. Click **Next**. The Central Authentication Service Configuration window is displayed.

14. If you are installing the SSO Manager for use with a third-party EIMS, proceed to step 15. (Do not enter anything on this window.)

    If you are installing the SSO Manager for use with CAS, enter the following information:

CAS Server Protocol	Protocol used by the CAS server (*https* or *http*)
CAS Server Host	CAS server host name
CAS Server Port	Port number where the CAS server is running
CAS Server Context	Context under which CAS is deployed and accessible via a browser (for example, *http(s)://<host>:<port>/<context>*).
SSO Manager Server Protocol	Protocol used by the SSO Manager (*https* or *http*)
SSO Manager Host	SSO Manager host name
SSO Manager Port	Port number where the SSO Manager is running

15. If you are running the utility in Oracle WebLogic 11g, proceed to step 16 to create security users.

    If you are running the utility in OC4J, skip to step 18. You cannot use the utility to create security users. They must be created manually. (See "Create a security user" on page 11-41.)

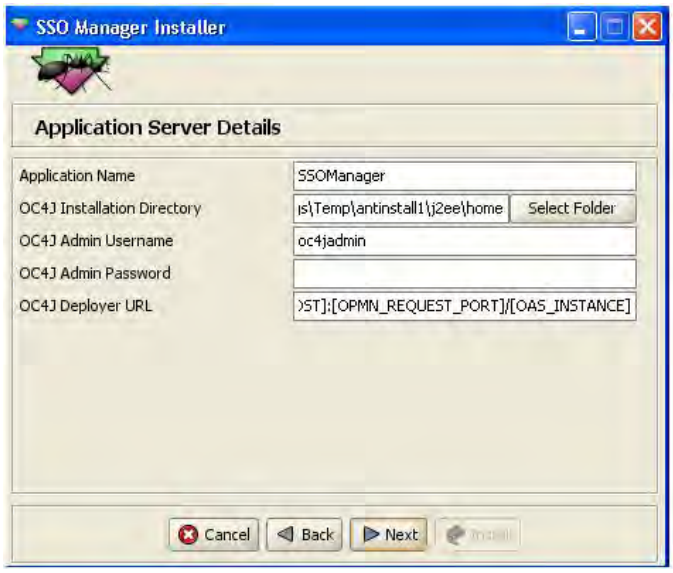
16. Click **Next**. The Create Security Users window is displayed. (This window is displayed for Oracle WebLogic 11g only.)

17. Enter the following information:

Username	User name for the SSO Manager application
Password	Password for the SSO Manager application

18. Click **Next**. The Application Server Details window is displayed.

OC4J version:



Oracle WebLogic 11g version:

19. Enter the following information:

19.1. If you are running the utility on OC4J:

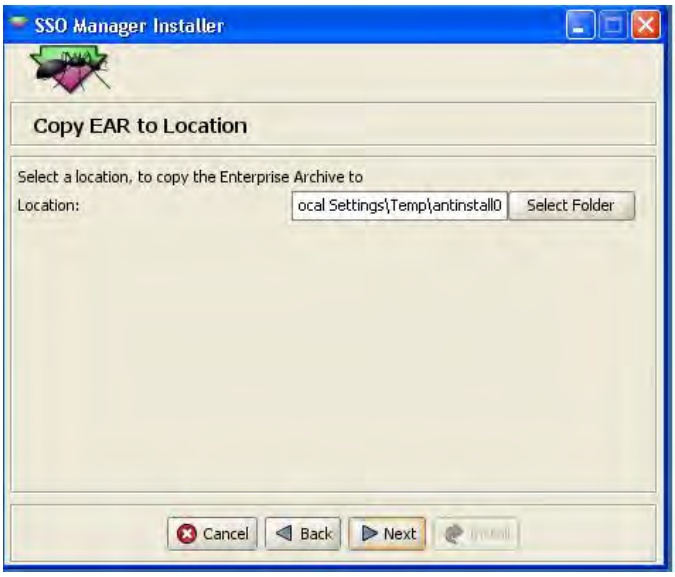
Application name	*SSOManager*
OC4J Installation Directory	*$OC4J_HOME\j2ee*<OC4J instance>
OC4J Admin Username	User name for the application server administrator
OC4J Admin Password	Password for the application server administrator

OC4J Deployer URL	**For standalone OC4J server:**
	`deployer:oc4j:localhost:23791`
	Port 23791 is the rmi port.
	For OAS SOA Suite:
	`deployer:oc4j:<ormis:>opmn://<opmn host>:<opmn port>/<oc4j instance name>`
	Example:
	`deployer:oc4j:opmn://m038071:6004/Ant_Installer_Test`
	Port 6004 is the opmn request port.

19.2. If you are running the utility on Oracle WebLogic 11g:

Application Name	*SSOManager*
Weblogic Server Home Directory	$WEBLOGIC_HOME\ Example: *C:\Oracle\Middleware\wlserver_10.3*
Weblogic Admin Username	User name for the application server administrator
Weblogic Admin Password	Password for the application server administrator
Weblogic Deployer URL	*t3://*<host>*:*<port>, where <port> is the WebLogic Admin Server port
Weblogic Managed Server Name	Name of the Oracle WebLogic Managed Server in the domain

20. Click **Next**. The Copy EAR to Location window is displayed.



21. Click **Select Folder** and browse to the location where you want to copy the ear file.

22. Click **Next**.

23. Click **Install**.

Complete the installation on Oracle Application Server

Note

The manual steps in this section apply to implementation of the SSO Manager on Oracle Application Server 10.1.3.4/5. If you are implementing the SSO Manager on Oracle WebLogic 11g, refer to "Complete the installation on Oracle WebLogic Server 11g" on page 11-48. ■

The configuration utility must be run to configure the SSO Manager ear file (see "Run the configuration utility" on page 11-19). In addition, the utility can optionally automate the following tasks:

- Create the database schema and database objects.
- Create the data sources.
- Deploy the SSO Manager into the application server.

Any components that were not created with the utility can be created manually, if needed. A manual configuration provides more flexibility and is recommended if you want to fine-tune specific parameters, use a RAC-based database, or deploy into a server cluster for high availability, scalability, and load balancing.

On OC4J, you *must* configure the security users manually (step 5). You *cannot* use the utility to configure the security users. All other components can be configured with the utility or with the following manual steps.

- Step 1, "Create a database user and objects"
- Step 2, "Define the integmgr data source"
- Step 3, "Define the data source for the Banner security administrator"
- Step 4, "Define the data source for connecting to the SSO Manager schema"
- Step 5, "Create a security user"
- Step 6, "Deploy the ear file"

You can skip any steps that were already performed with the configuration script.

A manual installation uses the application server's installation tools to deploy the SSO Manager into the target application server. This is typically accomplished by using the browser to connect to the Enterprise Manager console of the target application server and creating all resources that are required by the SSO Manager.

Step 1 Create a database user and objects

If the Create Database User and Objects option was not selected when running the configuration utility, a database user and objects must be created manually. The `sso-manager-oc4j-installer.jar` file contains SQL scripts that create a database user, tables, and packages for the SSO Manager. These scripts must be extracted and executed manually. The following steps provide one way to accomplish this task.

1. Navigate to the directory where `sso-manager-oc4j-installer.jar` is located.
2. Extract the sql subdirectory from the jar file:

   ```
   jar xvf sso-manager-oc4j-installer.jar sql
   ```

   This command creates an sql subdirectory in the current directory.
3. Navigate to the sql subdirectory:

   ```
   cd sql
   ```
4. Run SQL*Plus and connect as DBA.
5. Execute the `user_create_manual.sql` script:

   ```
   sqlplus> @user_create_manual.sql
   ```

6. When prompted, enter the following information:
   - Schema name for the SSO Manager schema (*ssomgr*)
   - Tablespace name for the SSO Manager schema (for example, *ssomgr_tbls*)
   - Name of the datafile with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0/dbs/ssomgr.dbf*)
   - Password for the SSO Manager schema (for example, *u_pick_it*)
7. Exit SQL*Plus:

   ```
   sqlplus> exit
   ```
8. Run SQL*Plus and connect as the user created in step 5 (for example, *ssomgr*).
9. Execute the `db_create.sql` script to create the tables:

   ```
   sqlplus> @db_create.sql
   ```
10. Exit SQL *Plus.:

    ```
    sqlplus> exit
    ```

Step 2 Define the integmgr data source

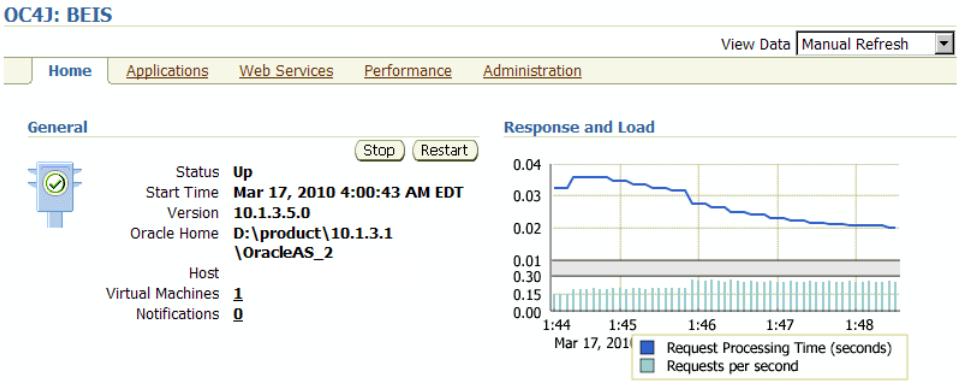
If the Create Datasource option was not selected when running the configuration utility, several datasources must be created manually for the SSO Manager. Use the following steps to define the data source for connecting to the `integmgr` database schema in Banner.

1. Connect to the Oracle Enterprise Manager:

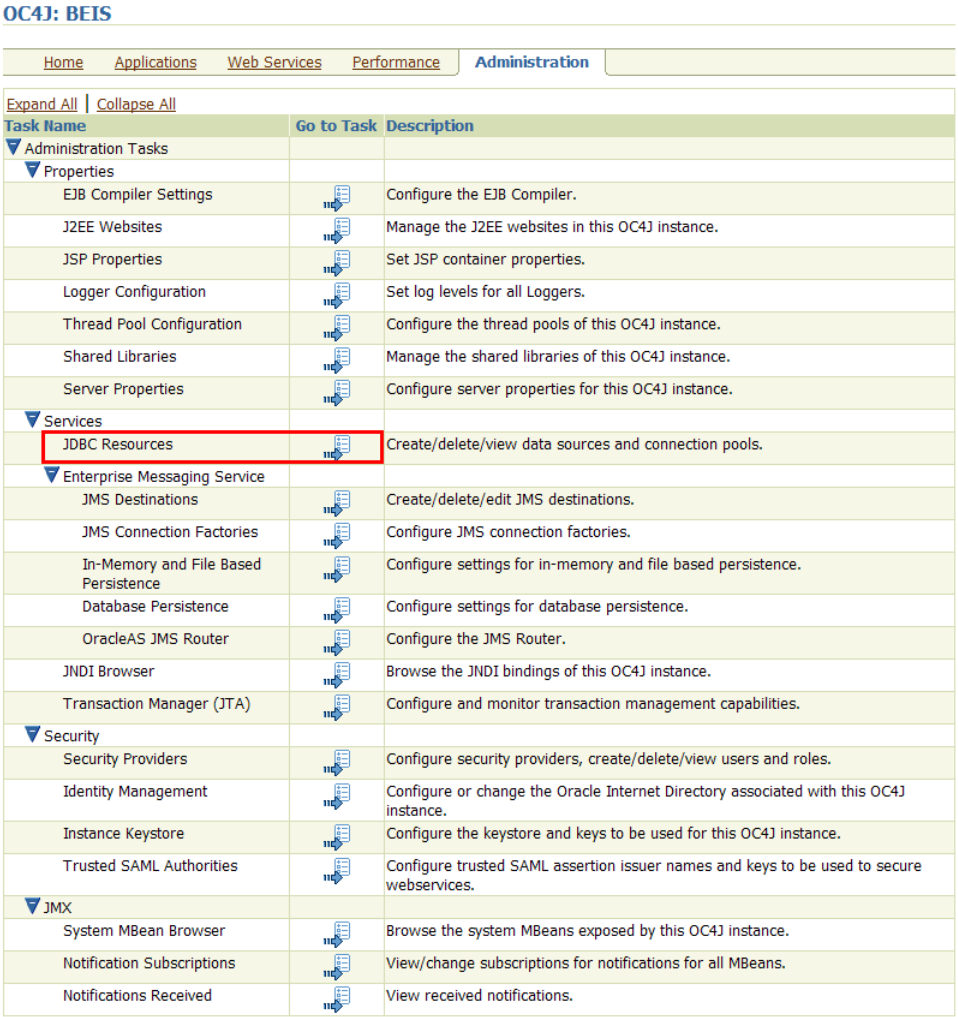
   ```
   http://<host>:<port>/em
   ```

   The console is displayed.

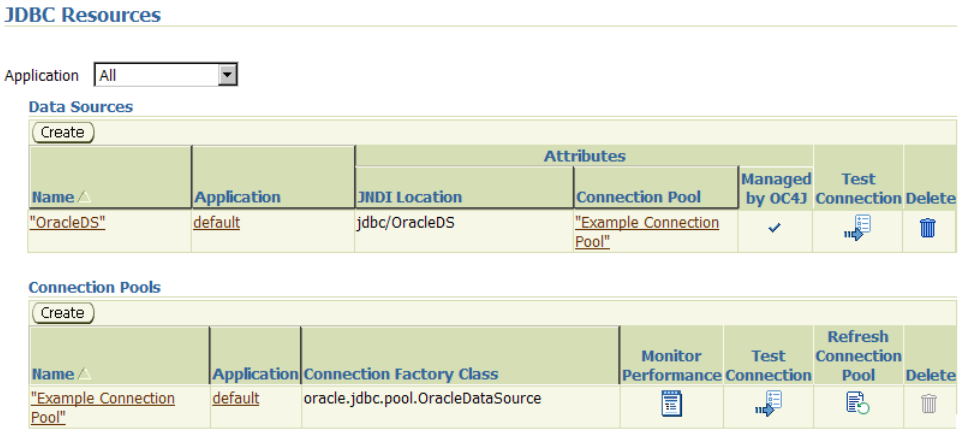
2. Click the name of the OC4J instance that will host the SSO Manager. The Home page for the selected instance is displayed.



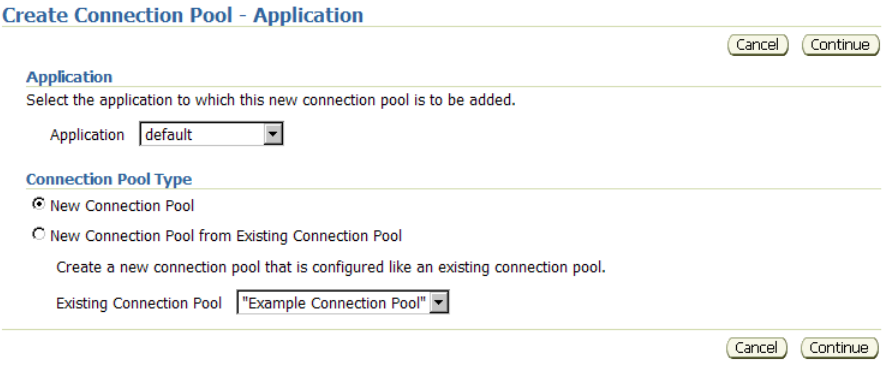
3. Select the **Administration** tab. A list of tasks is displayed.



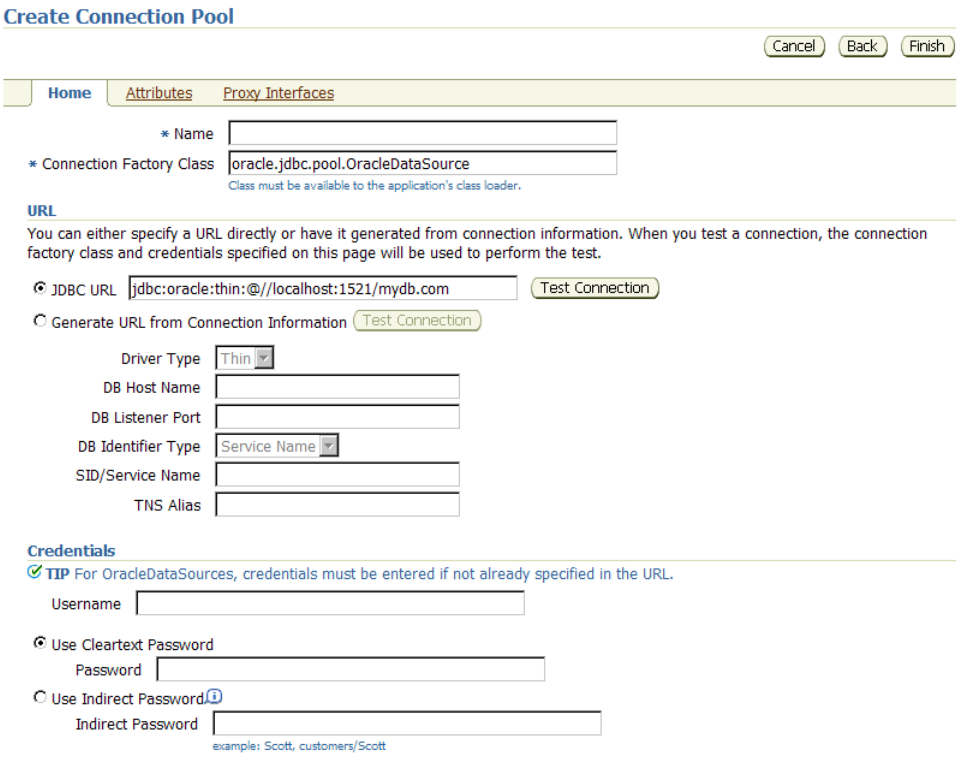
4. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.



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



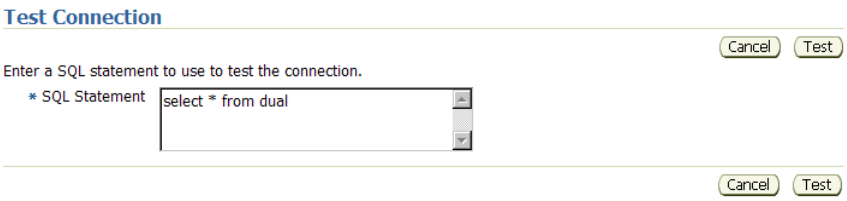
6. Click **Continue**. The Create Connection Pool page is displayed.



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

Name	*SSOIntegMgr_Pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*integmgr*
Use Cleartext Password	Select **Use Cleartext Password** and enter a password for the `integmgr` schema.

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



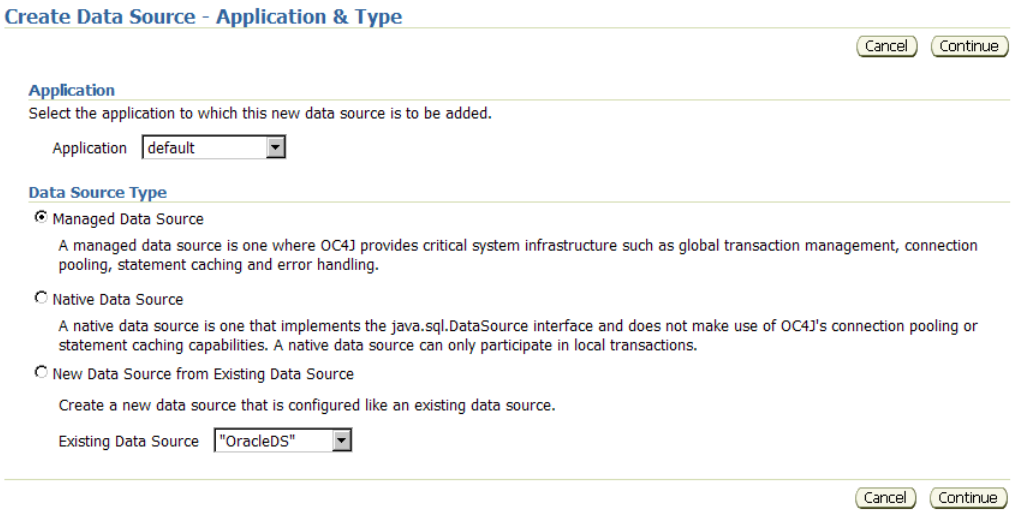
9. Click **Test** to test the connection pool for the `integmgr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

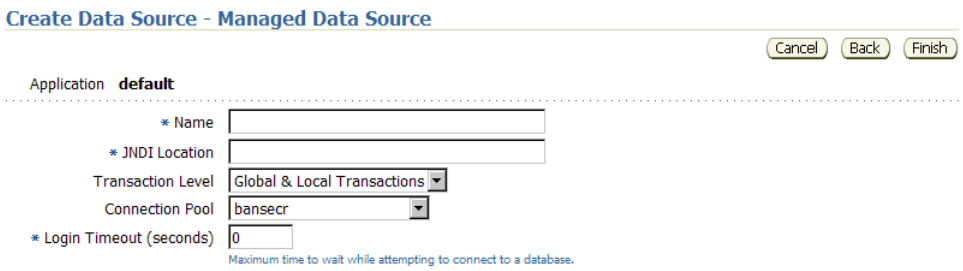
   9.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

10. Click **Finish**.

11. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.



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



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

Name	*SSOIntegMgrDataSource*
JNDI Location	*jdbc/ssomgr_integmgr_banner*
Connection Pool	*SSOIntegMgr_Pool*

14. Click **Finish**.

Step 3 Define the data source for the Banner security administrator

If the Create Datasource option was not selected when running the configuration utility, several datasources must be created manually for the SSO Manager. Use the following steps to define the data source for the Banner security manager.

1. Select the **Administration** tab. A list of tasks is displayed.
2. Select **JDBC Resources** in the Services section. The JDBC Resources page is displayed.
3. Click **Create** in the Connection Pools section. The Create Connection Pool - Application page is displayed.
4. Click **Continue**. The Create Connection Pool page is displayed.
5. Enter the following information to set up the connection pool for the `bansecr` schema:

Name	*SSOINBAdmin_Pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*bansecr*
Use Cleartext Password	Select **Use Cleartext Password** and provide a password for the `bansecr` schema.

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

7. Click **Test** to test the connection pool for the `bansecr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

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

8. Click **Finish**.

9. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

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

11. Enter the following information to set up the `inbadmin` data source:

Name	*SSOINBAdminDataSource*
JNDI Location	*jdbc/ssomgr_inbadmin*
Connection Pool	*SSOINBAdmin_Pool*

12. Click **Finish**.

Step 4 Define the data source for connecting to the SSO Manager schema

If the Create Datasource option was not selected when running the configuration utility, several datasources must be created manually for the SSO Manager. Use the following steps to define the data source for connecting to the SSO Manager schema.

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

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

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

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

5. Enter the following information to set up the connection pool for the `ssomgr` schema:

Name	*SSOManager_Pool*
Connection Factory Class	*oracle.jdbc.pool.OracleDataSource*
JDBC URL	*jdbc:oracle:thin:@host:port:SID* where *host* = database host *port* = database listener port (usually *1521*) *SID* = database instance
Username	*ssomgr*
Use Cleartext Password	Select **Use Cleartext Password** and provide a password for the `ssomgr` schema.

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

7. Click **Test** to test the connection pool for the `bansecr` schema. The Create Connection Pool page is redisplayed with a success or failure message.

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

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

8. Click **Finish**.

9. Click **Create** in the Data Sources section on the JDBC Resources page. The Create Data Source - Application & Type page is displayed.

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

11. Enter the following information to set up the `ssomgr` data source:

Name	*SSOManagerDataSource*
JNDI Location	*jdbc/ssomgr*
Connection Pool	*SSOManager_Pool*

12. Click **Finish**.

Step 5 Create a security user

Security users and groups cannot be created by the configuration utility on Oracle Application Server 10.1.3.4/5. Use the following steps to configure the security user and group for the SSO Manager.

Warning

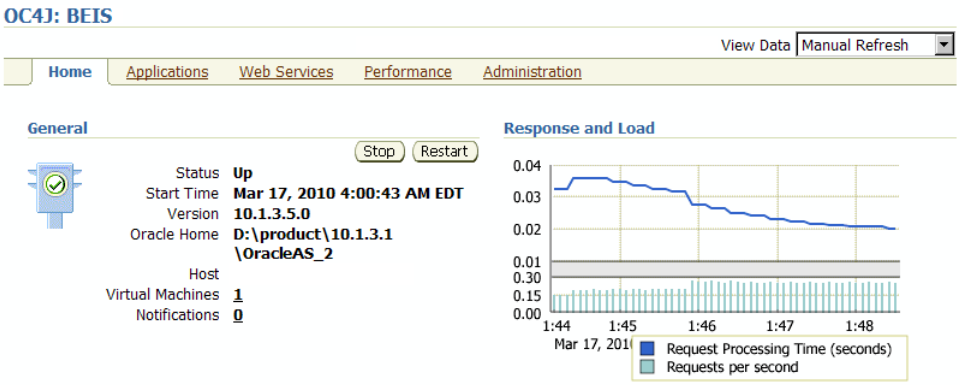
Although other manual steps are optional, this manual step is required. ■

1. Connect to the Oracle Enterprise Manager:

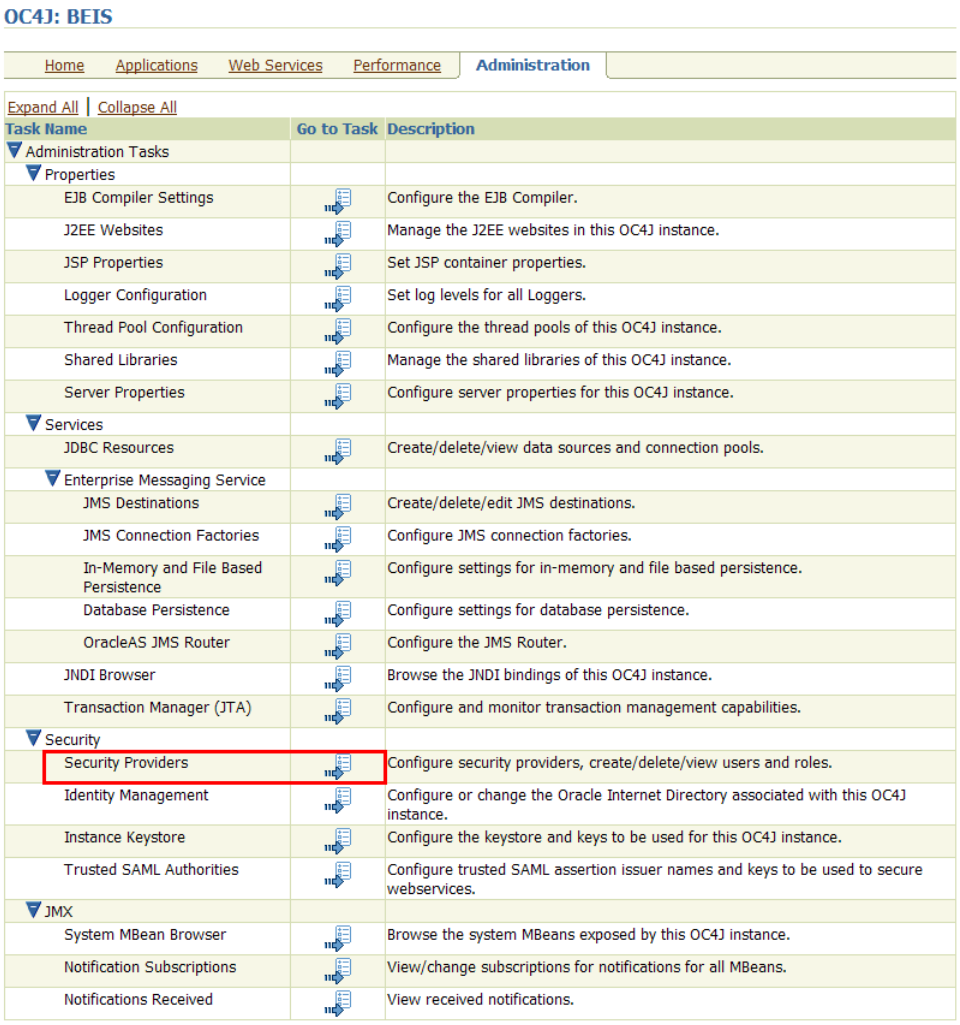
   `http://<host>:<port>/em`

   The console is displayed.

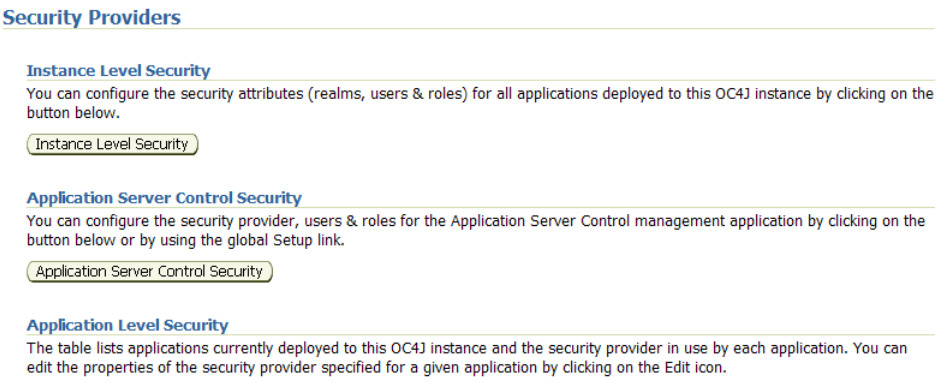
2. Click the name of the OC4J instance that will host the SSO Manager. The Home page for the selected instance is displayed.



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

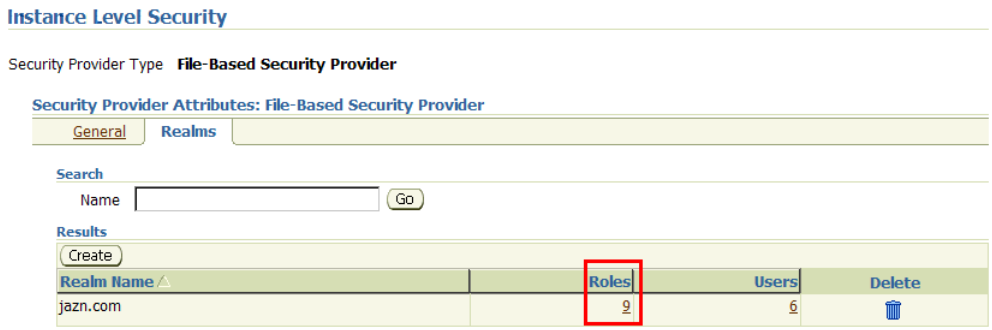


4. Select **Security Providers** in the Security section. The Security Providers page is displayed.



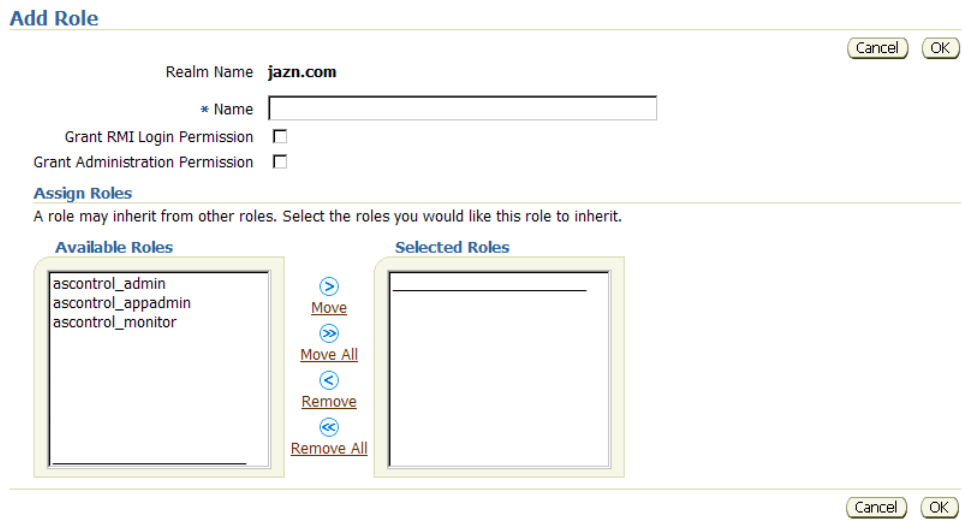
5. Click **Instance Level Security**. The Instance Level Security page is displayed.

6. Select the **Realms** tab.



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

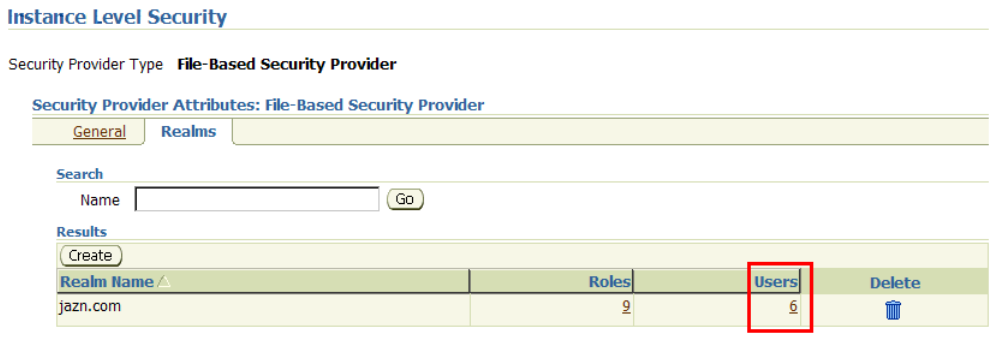
8. Click **Create**. The Add Role page is displayed.



9. Enter *ssomgr-role* in the **Name** field.

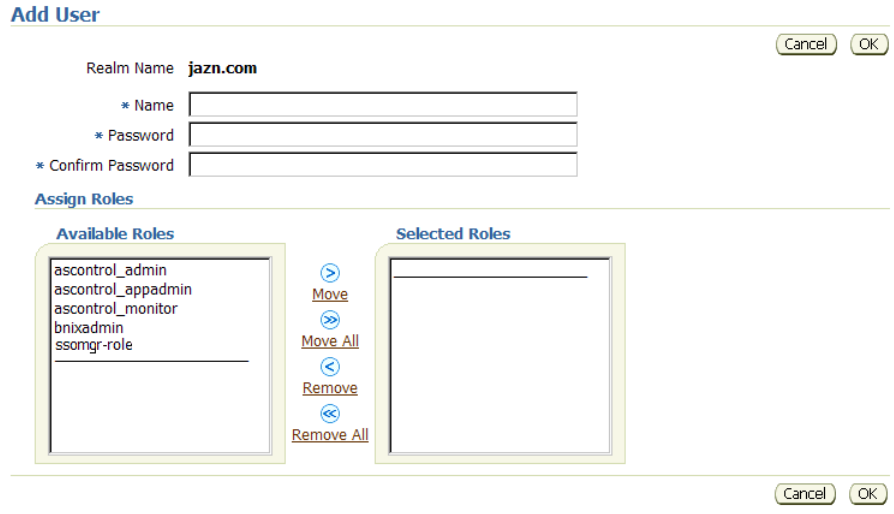
10. Click **OK**. The Roles page is redisplayed with the new role.

11. Return to the Instance Level Security page.



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

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



14. Enter the following information to create a user:

Name	*ssomgr* (This is an example. Enter the name of your choice.)
Password	Password used to log in to the SSO Manager administrative interface
Confirm Password	Confirmation of the password

15. In the Assign Roles section, select the *ssomgr-role* role in the **Available Roles** list and move it to the **Selected Roles** list.

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

Step 6 Deploy the ear file

If the Deploy EAR option was not selected when running the configuration utility, the SSO Manager ear file must be deployed manually. Use the following steps, on Oracle Application Server 101.3.4/5, to deploy the ear file.

Note

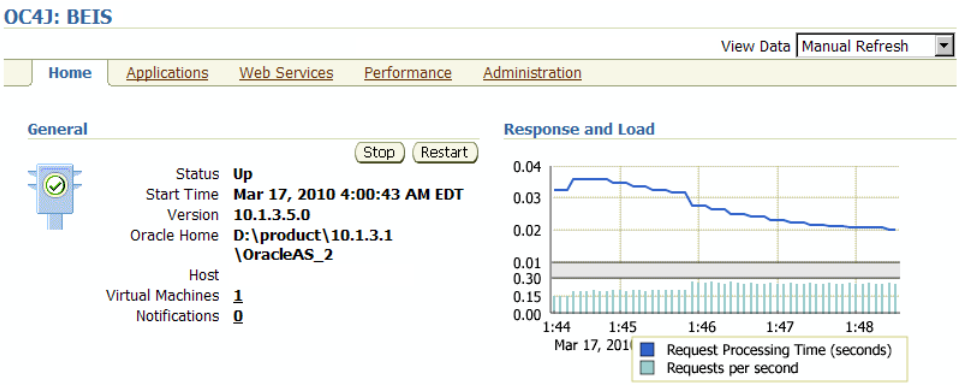
You must know the location of the SSO Manager ear file that you configured with the configuration utility, as described on page 11-31. ■

1. Connect to the Oracle Enterprise Manager:

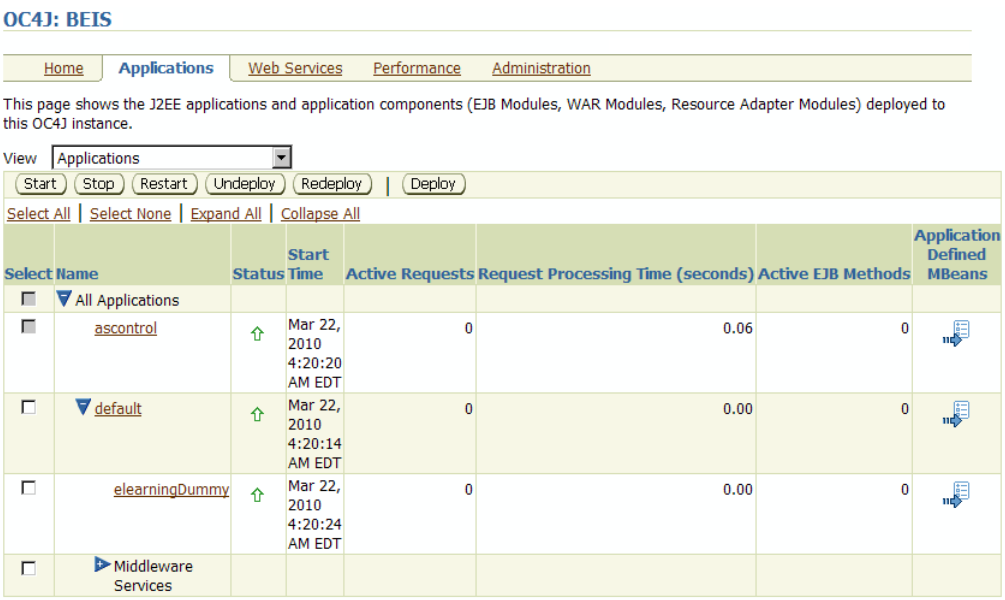
   `http://<host>:<port>/em`

   The console is displayed.

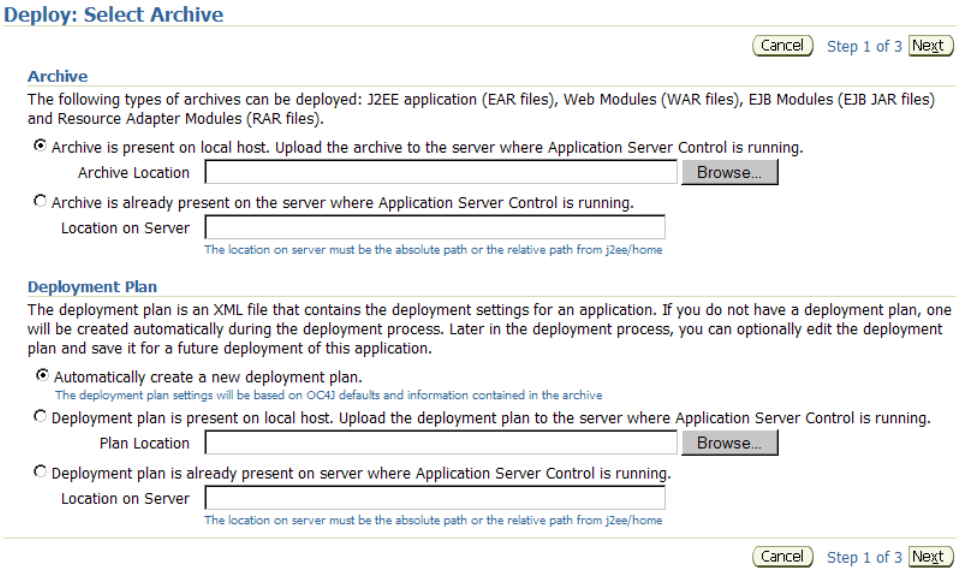
2. Click the name of the OC4J instance that will host the SSO Manager. The Home page for the selected instance is displayed.



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



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



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 navigate to the `sso-manager.ear` file. This file was copied to a specified location when the

configuration utility was run. (See the steps are on page 11-31.)

5.3. 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.**

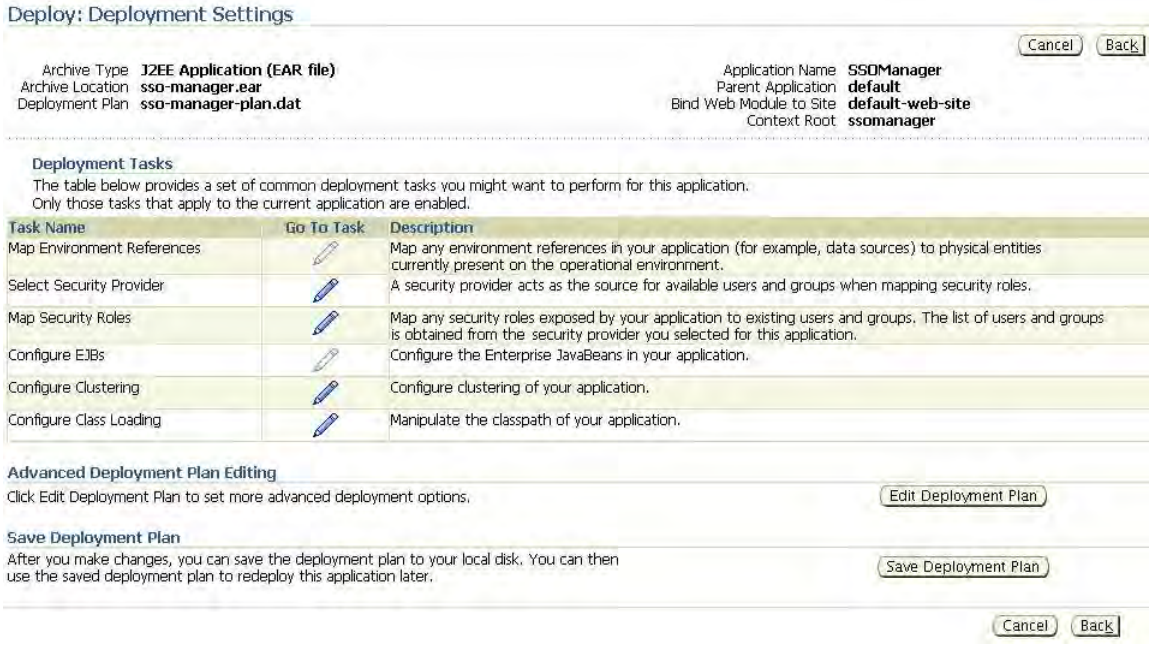
6.2. In the **Plan Location** field, click **Browse** and navigate to the `sso-manager-plan.dat` file. (This file was copied to the same location where the ear file was copied when the configuration utility was run.)

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.

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

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



10. Click **Deploy** to accept the values and install the SSO Manager. A deployment confirmation page is displayed.

11. Click **Return** to continue. The **Applications** tab is displayed with the deployed application.

Complete the installation on Oracle WebLogic Server 11g

Note

The manual steps in this section apply to implementation of the SSO Manager on Oracle WebLogic 11g (version 10.3.2). If you are implementing the SSO Manager on Oracle Application Server 10.1.3.4/5, refer to "Complete the installation on Oracle Application Server" on page 11-31.

The configuration utility must be run to configure the SSO Manager ear file (see "Run the configuration utility" on page 11-19). In addition, the utility can optionally automate the following tasks:

- Create the database schema and database objects.
- Create the data sources.
- Create security users for logging into the application.
- Deploy the SSO Manager into the application server.

Any components that were not created with the utility can be created manually, if needed. A manual configuration provides more flexibility and is recommended if you want to fine-tune specific parameters, use a RAC-based database, or deploy into a server cluster for high availability, scalability, and load balancing.

All components can be configured with the utility or with the following manual steps.

- Step 1, “Create a database user and objects”
- Step 2, “Define the integmgr data source”
- Step 3, “Define the data source for the Banner security administrator”
- Step 4, “Define the data source for connecting to the SSO Manager schema”
- Step 5, “Create a security user”
- Step 6, “Deploy the ear file”

You can skip any steps that were already performed with the configuration script.

A manual installation uses the application server’s installation tools to deploy the SSO Manager into the target application server. This is typically accomplished by using the browser to connect to the Enterprise Manager console of the target application server and creating all resources that are required by the SSO Manager.

Step 1 Create a database user and objects

If the Create Database User and Objects option was not selected when running the configuration utility, a database user and objects must be created manually. The `sso-manager-weblogic-installer.jar` file contains SQL scripts that create a database user, tables, and packages for the SSO Manager. These scripts must be extracted and executed manually. The following steps provide one way to accomplish this task.

1. Navigate to the directory where `sso-manager-weblogic-installer.jar` is located.
2. Extract the sql subdirectory from the jar file:

   ```
   jar xvf sso-manager-weblogic-installer.jar sql
   ```

   This command creates an sql subdirectory in the current directory.
3. Navigate to the sql subdirectory:

   ```
   cd sql
   ```
4. Run SQL*Plus and connect as DBA.
5. Execute the `user_create_manual.sql` script:

   ```
   sqlplus> @user_create_manual.sql
   ```

6. When prompted, enter the following information:
   - Schema name for the SSO Manager schema (*ssomgr*)
   - Tablespace name for the SSO Manager schema (for example, *ssomgr_tbls*)
   - Name of the datafile with the complete path (for example, */u01/app/oracle/ORDBMS/10.2.0/dbs/ssomgr.dbf*)
   - Password for the SSO Manager schema (for example, *u_pick_it*)
7. Exit SQL*Plus:

   ```
   sqlplus> exit
   ```

8. Run SQL*Plus and connect as the user created in step 5 (for example, *ssomgr*).
9. Execute the `db_create.sql` script to create the tables:

   ```
   sqlplus> @db_create.sql
   ```

10. Exit SQL *Plus.:

    ```
    sqlplus> exit
    ```

Step 2 Define the integmgr data source

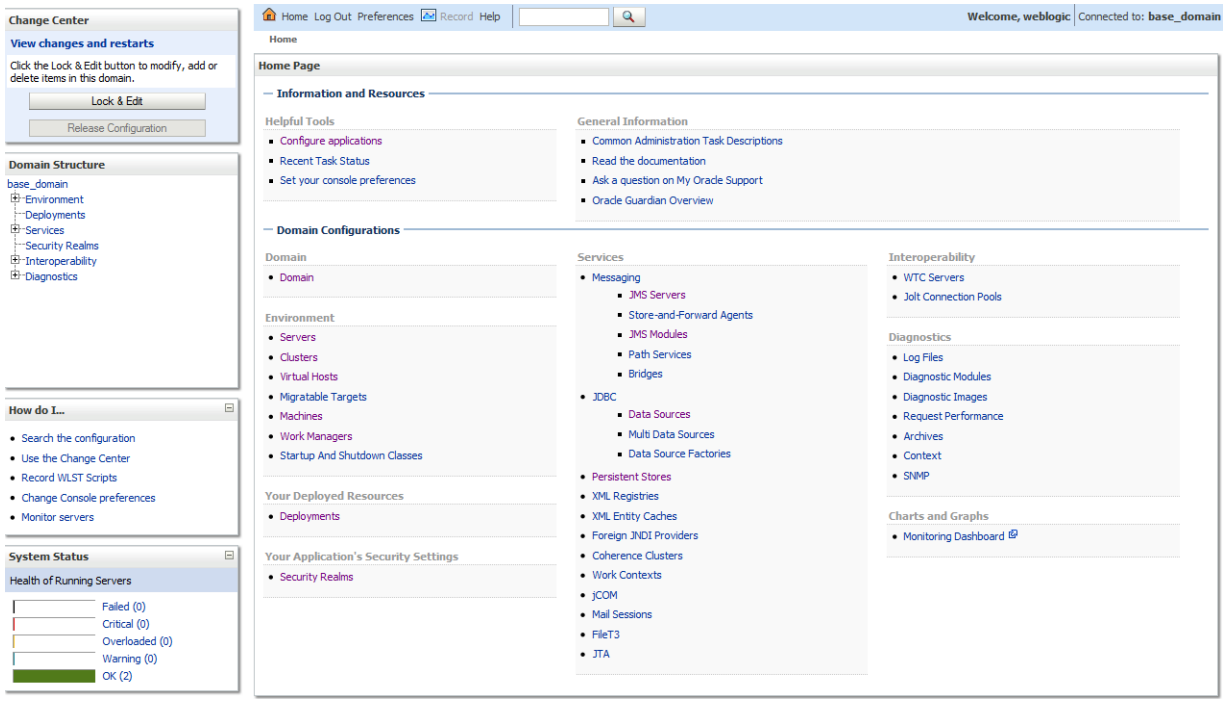
If the Create Datasource option was not selected when running the configuration utility, several datasources must be created manually for the SSO Manager. Use the following steps to define the data source for connecting to the `integmgr` database schema in Banner.

Use the following steps, on Oracle WebLogic 11g (version 10.3.2), to define the data source for connecting to the `integmgr` database schema in Banner.

1. Connect to the Oracle WebLogic Server Administration Console:

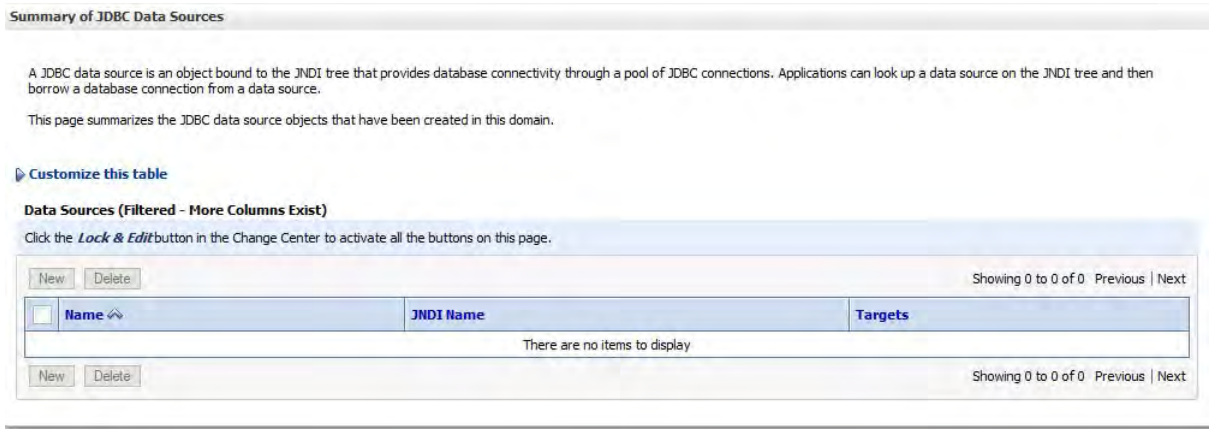
   ```
   http://<host>:<port>/console
   ```

The Home Page is displayed.



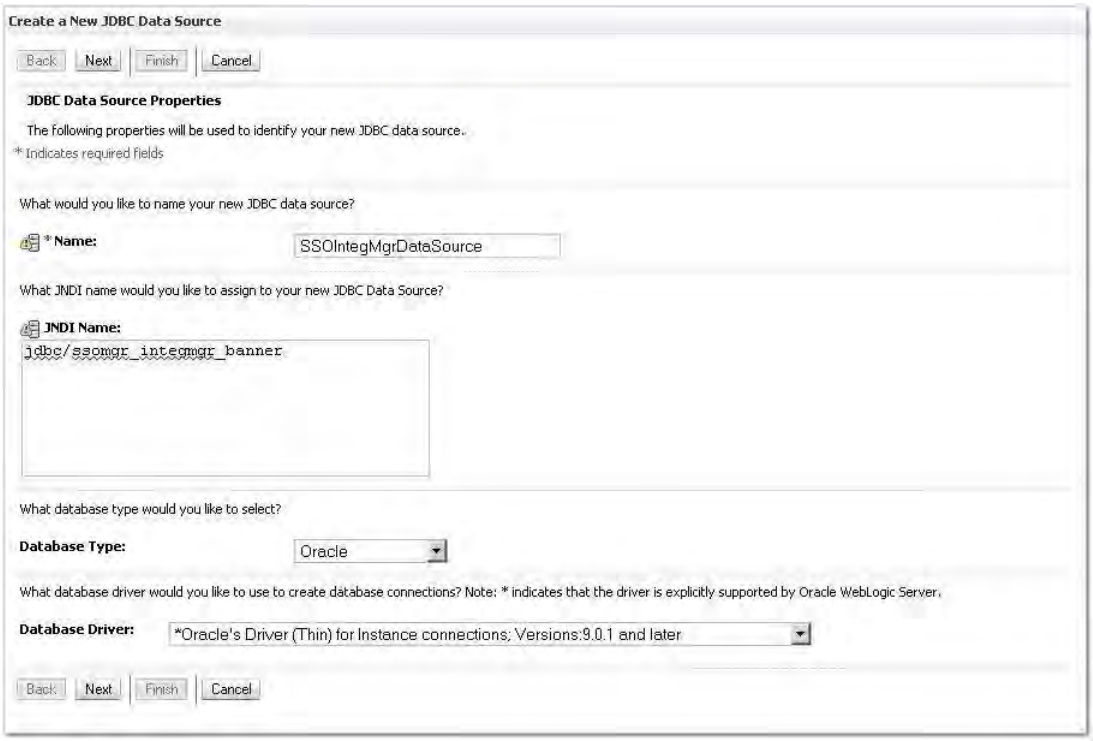
2. In the Domain Structure pane, expand and click **Services -> JDBC -> Data Sources**.

The Summary of JDBC Data Sources page is displayed.



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

4. On the Summary of JDBC Data Sources page, click **New**. The Create a New JDBC Data Source page is displayed.

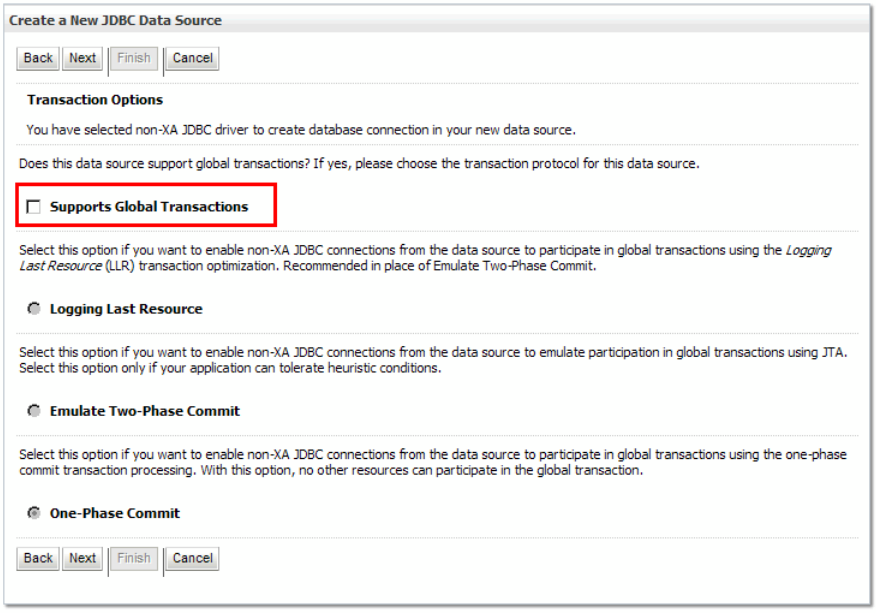


5. Enter the following data source properties:

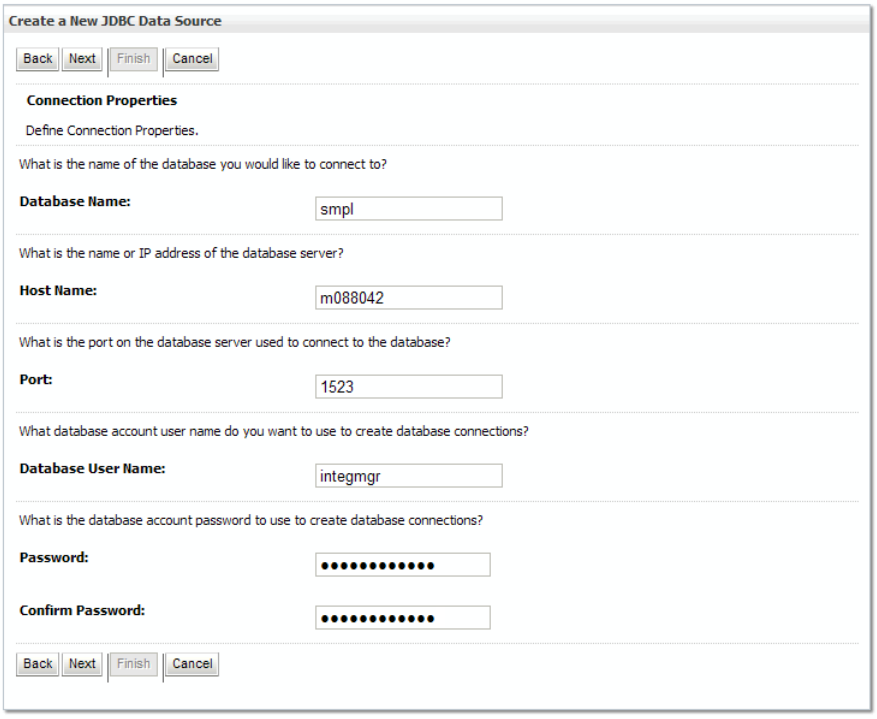
Name	*SSOIntegMgrDataSource*
JNDI Name	*jdbc/ssomgr_integmgr_banner*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,and later*.

6. 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 8. If the following page is not displayed, skip to step 9.



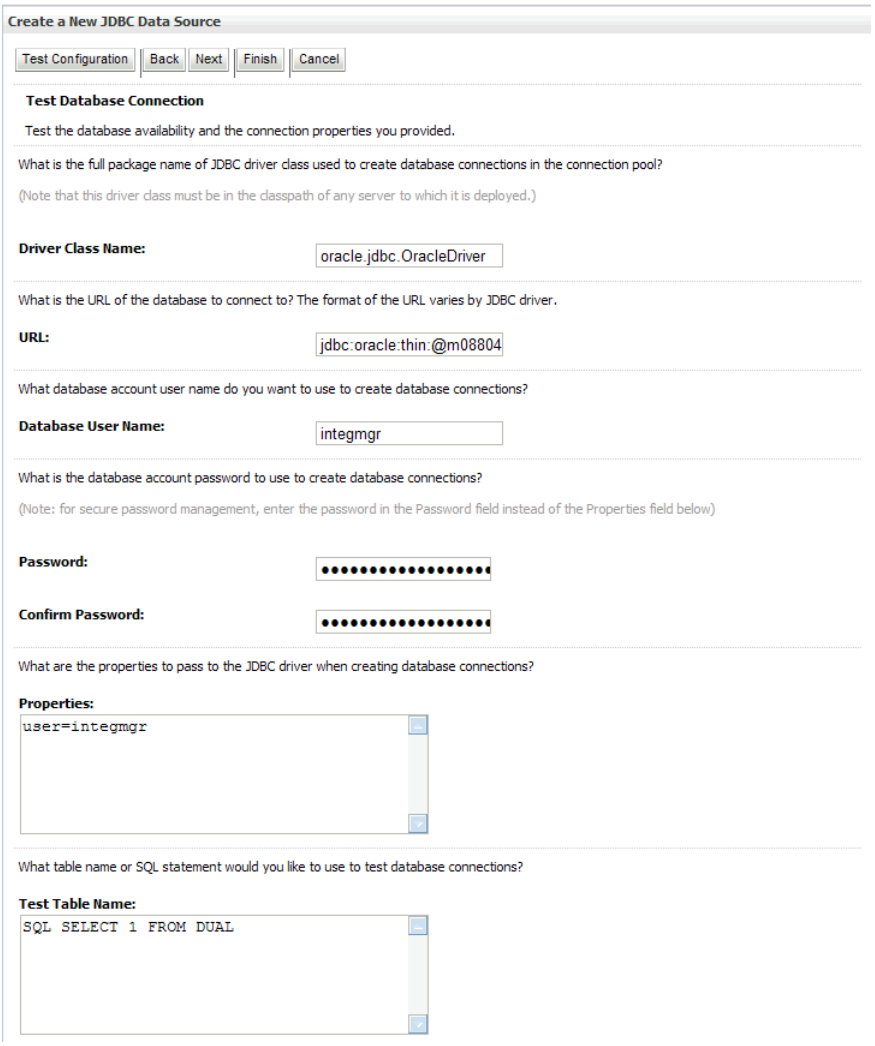
7. Click **Next**. The next page is displayed.



8. Enter the following connection properties:

Database Name	Name of the database to which you are connecting
Host Name	IP address or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*integmgr*
Password	Password for the `integmgr` user
Confirm Password	Confirmation of the password

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



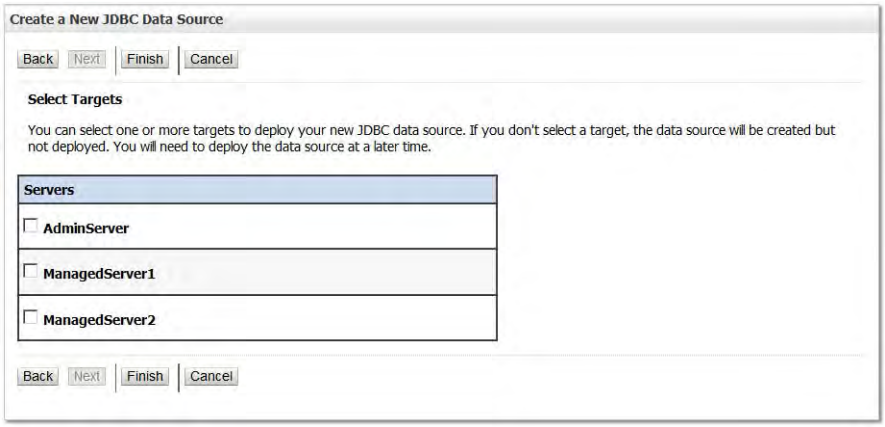
10. Verify the property values.

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.

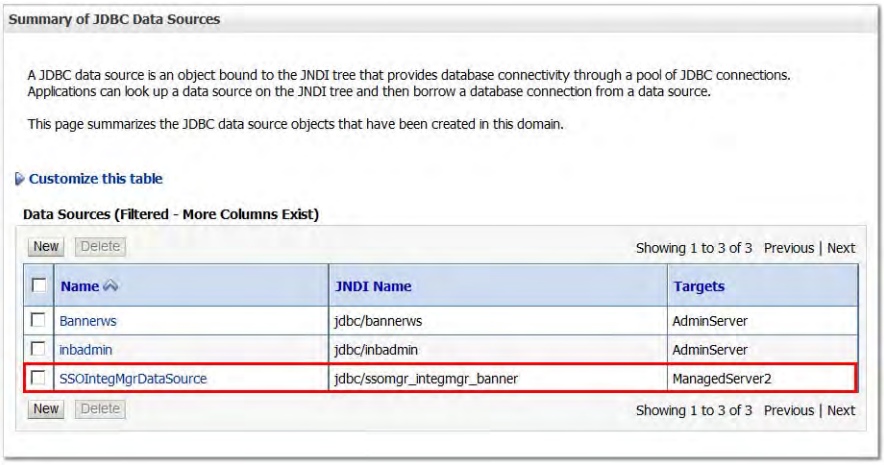
    11.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.

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



13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the SSO Manager will be deployed.

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



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

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

Step 3 Define the data source for the Banner security administrator

If the Create Datasource option was not selected when running the configuration utility, several datasources must be created manually for the SSO Manager. Use the following

steps, on Oracle WebLogic 11g (version 10.3.2), to define the data source for the Banner security administrator.

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	*SSOINBAdminDataSource*
JNDI Name	*jdbc/ssomgr_inbadmin*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

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 or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*bansecr*

Password	Password for the `bansecr` 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 **Next**. The next page is displayed.
13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the SSO Manager will be deployed.
14. Click **Finish**. The Summary of JDBC Data Sources page is displayed with the new data source.
15. Verify that the new data source is associated with the server.
16. In the Change Center pane, click **Activate Changes.**

Step 4 Define the data source for connecting to the SSO Manager schema

If the Create Datasource option was not selected when running the configuration utility, several datasources must be created manually for the SSO Manager. Use the following steps, on Oracle WebLogic 11g (version 10.3.2), to define the data source for connecting to the SSO Manager schema.

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	*SSOManagerDataSource*
JNDI Name	*jdbc/ssomgr*
Database Type	*Oracle*
Database Driver	Appropriate database driver that is used to create database connections. If your database is RAC-based, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11*. Otherwise, select **Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11*.

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 or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	*ssomgr*
Password	Password for the `ssomgr` 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 **Next**. The next page is displayed.

13. Select the server(s) where you want to deploy the new data source. At a minimum, this should be the Managed Server where the SSO Manager will be deployed.

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

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

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

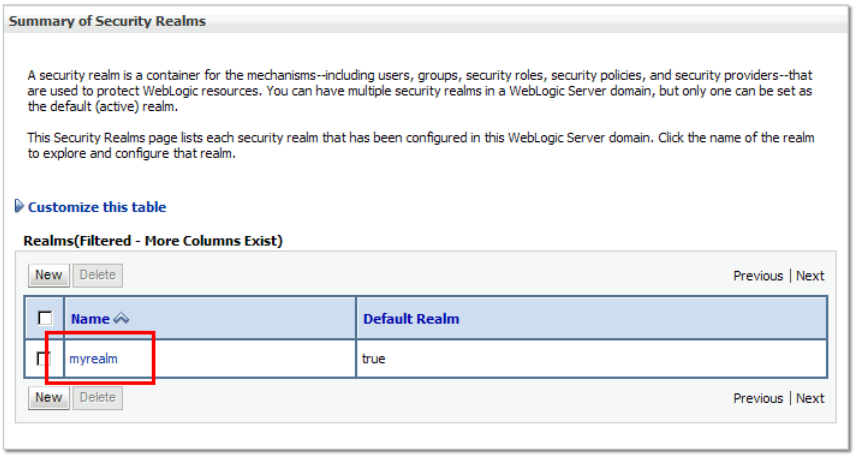
Step 5 Create a security user

If the Create Security Users option was not selected when running the configuration utility, a security group and associated users must be created manually for accessing the SSO Manager. Use the following steps, on Oracle WebLogic 11g (version 10.3.2), to configure the security group and user.

1. Connect to the Oracle WebLogic Server Administration Console for the domain into which the SSO Manager will be deployed.

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

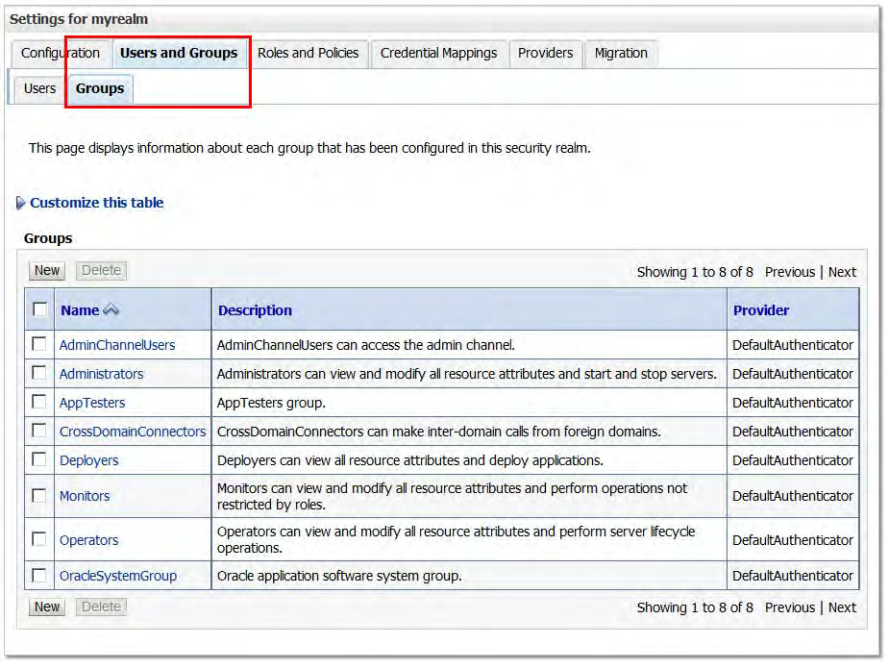
The Summary of Security Realms page is displayed.



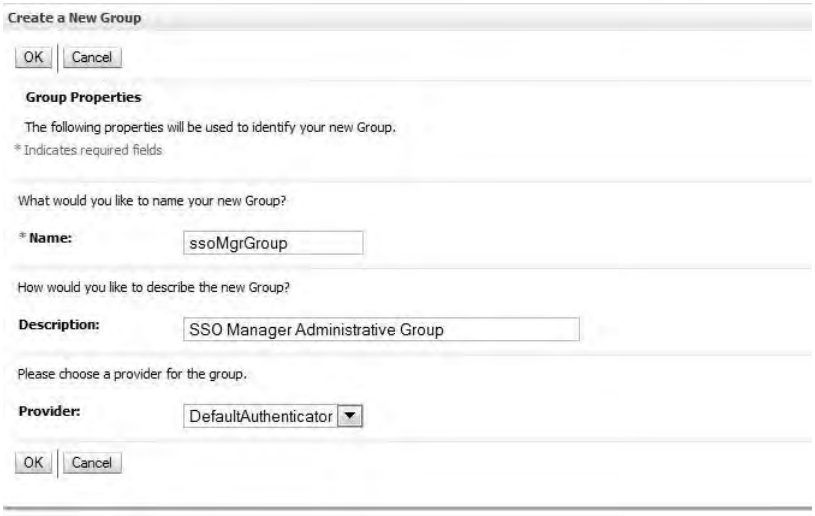
3. Click **myrealm**. The Settings page is displayed.

4. Select the **Users and Groups** tab.

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



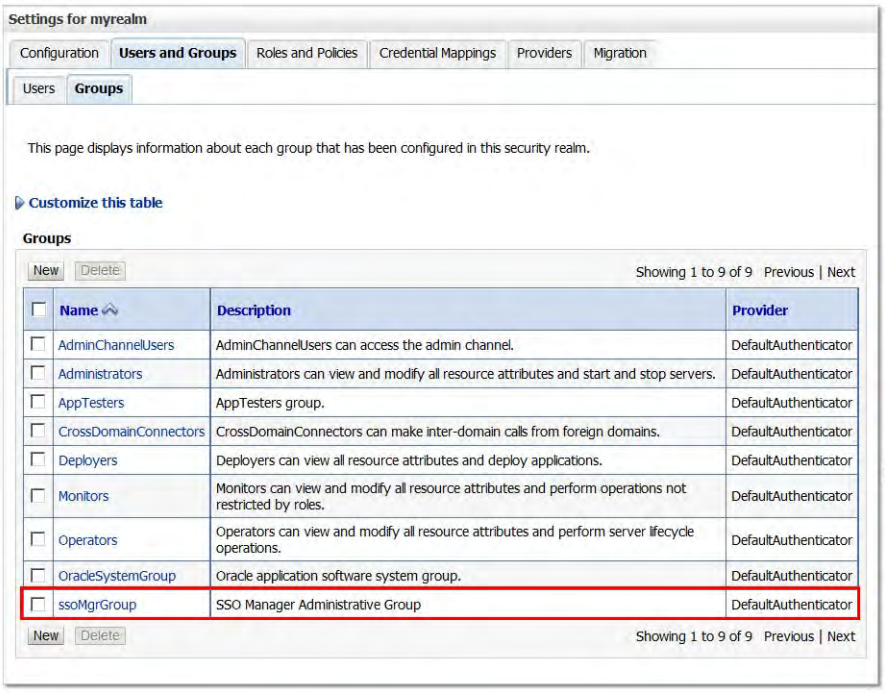
6. Click **New**. The Create a New Group page is displayed.



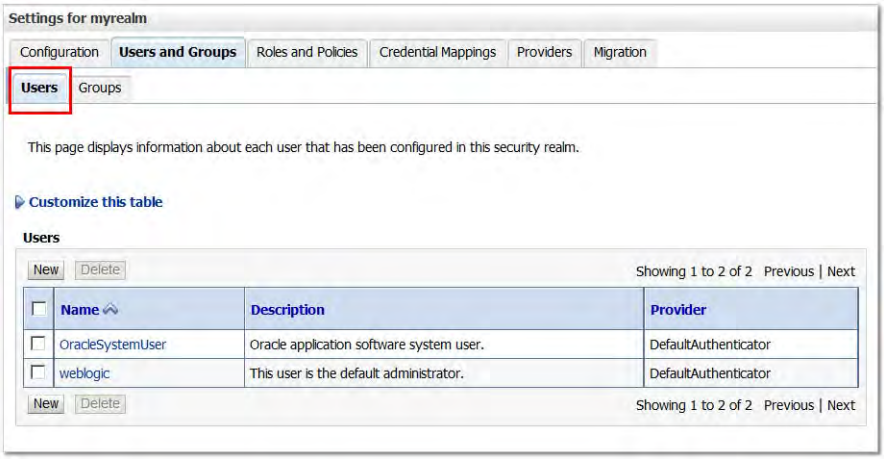
7. Enter the following information to create a group:

Name	*ssoMgrGroup*
Description	*SSO Manager Administrative Group*
Provider	*DefaultAuthenticator*

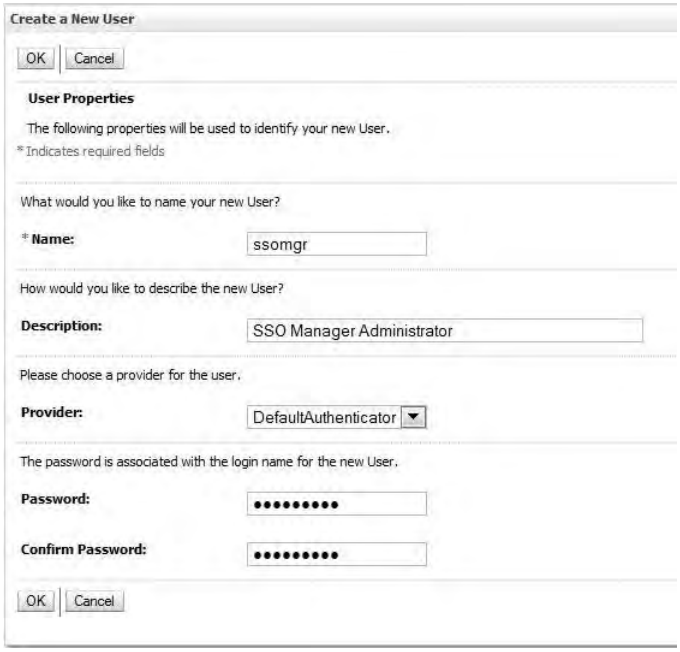
8. Click **OK**. The table of groups is redisplayed with the new group.



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



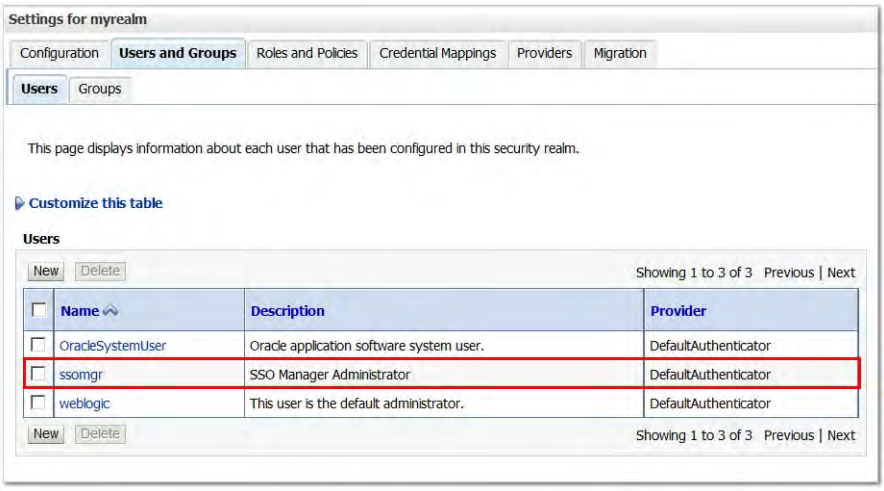
10. Click **New**. The Create a New User page is displayed.



11. Enter the following information to create a user:

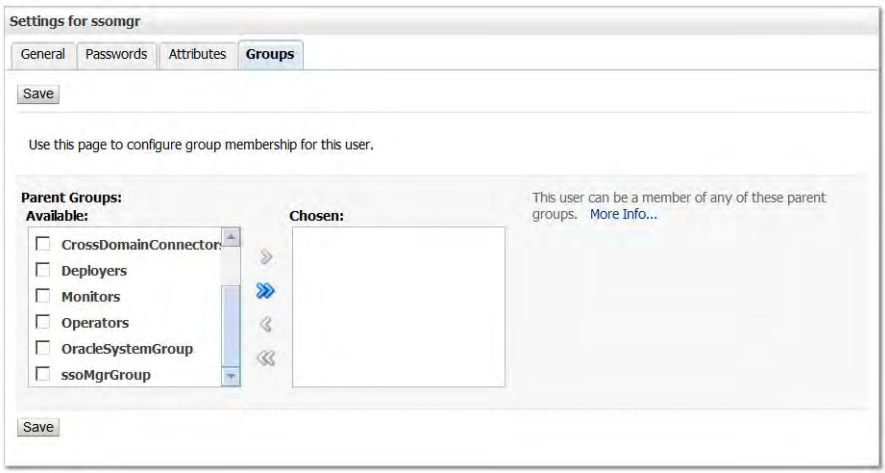
Name	*ssomgr* (This is an example. Enter the name of your choice.*)*
Description	*SSO Manager Administrator*
Provider	*DefaultAuthenticator*
Password	Password used to log in to the SSO Manager administrative interface
Confirm Password	Confirmation of the password

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

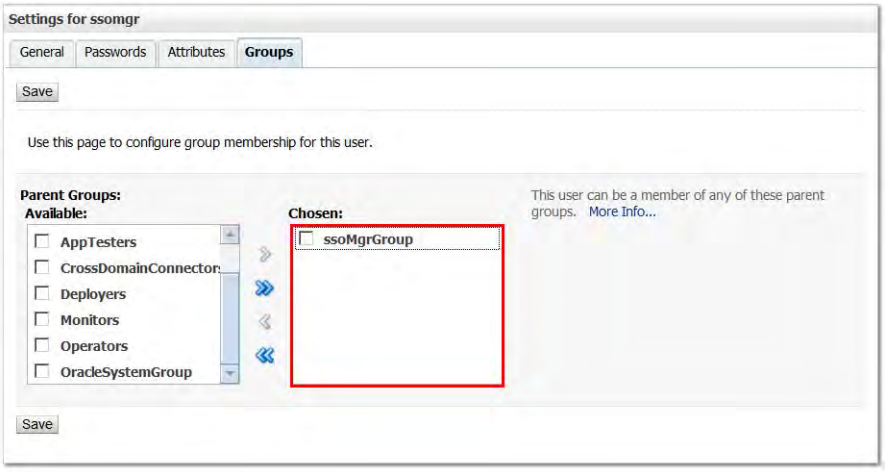


13. Click the name of the user you just created. The Settings page for the user is displayed.

14. Select the **Groups** tab.



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



16. Click **Save**.

Step 6 Deploy the ear file

If the Deploy EAR option was not selected when running the configuration utility, the SSO Manager ear file must be deployed manually. Use the following steps, Oracle WebLogic 11g (version 10.3.2), to deploy the ear file.

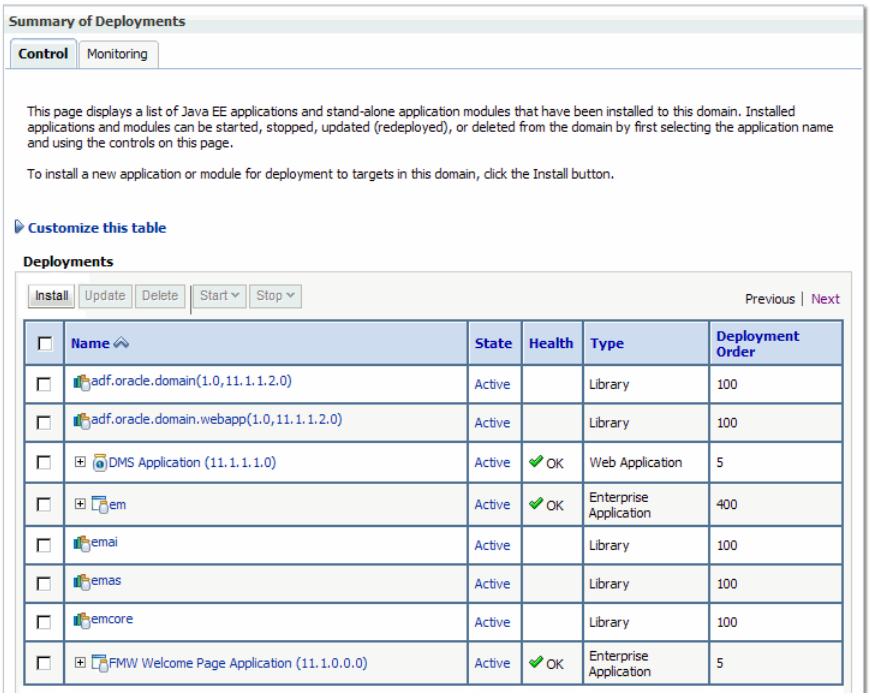
Note

You must know the location of the SSO Manager ear file that you configured with the configuration utility, as described on page 11-31. ■

1. Connect to the Oracle WebLogic Server Administration Console for the domain into which the SSO Manager will be deployed.

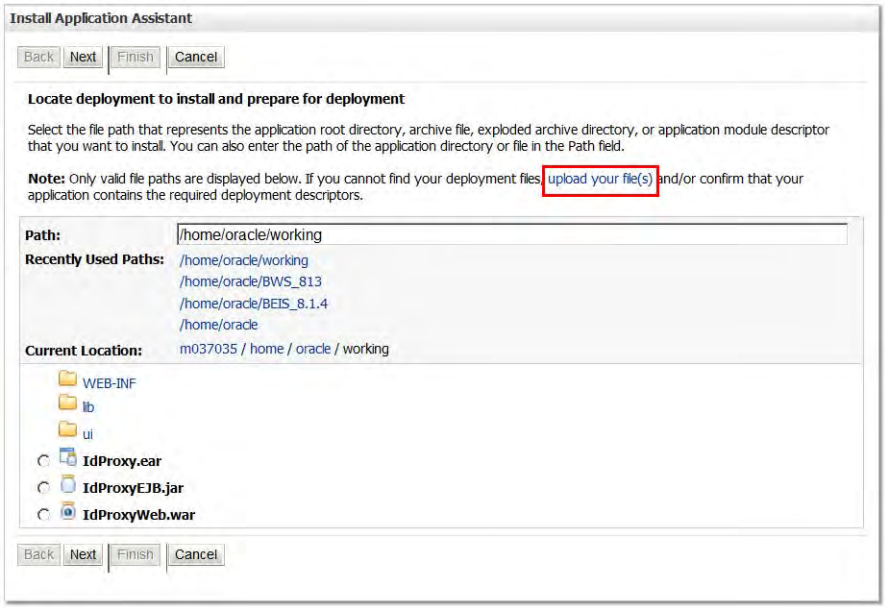
2. In the Domain Structure pane, click **Deployments**.

The Summary of Deployments page is displayed.

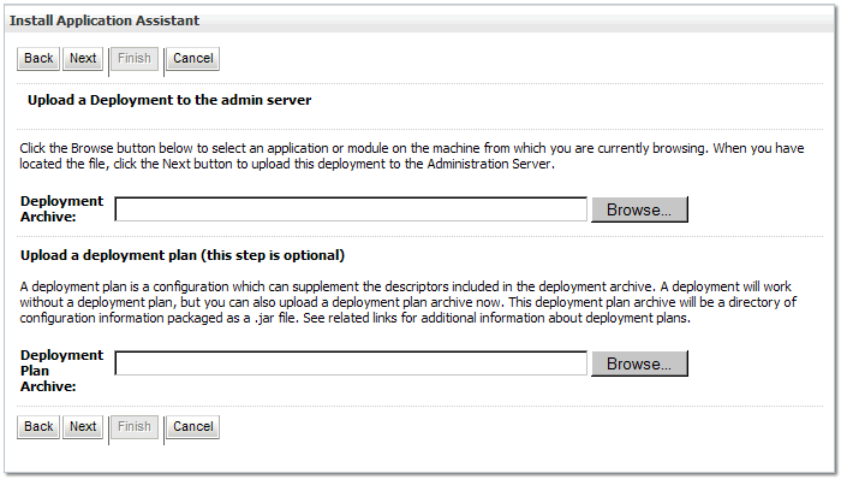


3. In the Change Center pane, click **Lock & Edit**.

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



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

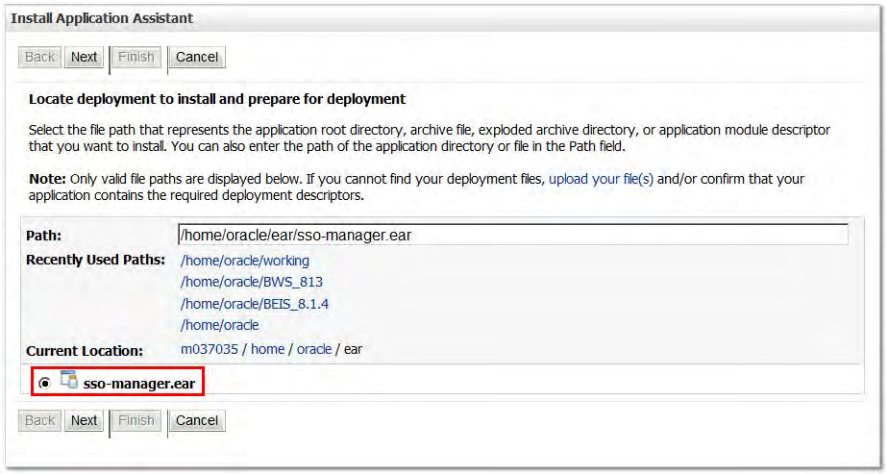


6. Select the file to be uploaded:

   6.1. In the **Deployment Archive** field, click **Browse** and navigate to the `sso-manager.ear` file. (This file was copied to a specified location when the configuration utility was run.)

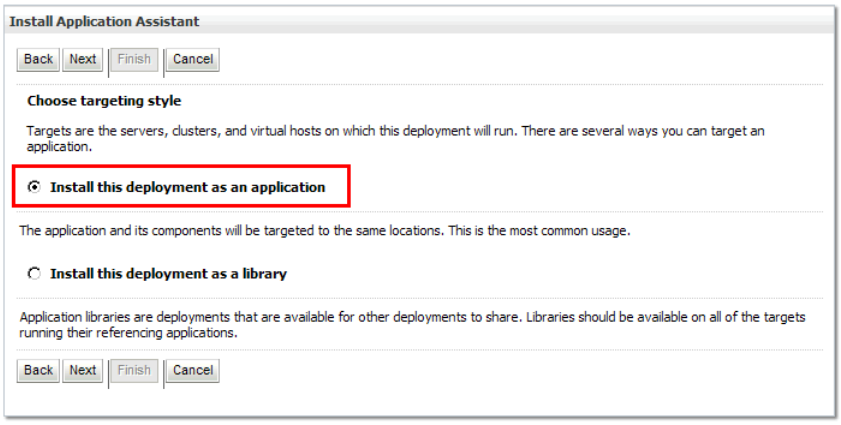
   6.2. Select the file and click **Open**.

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



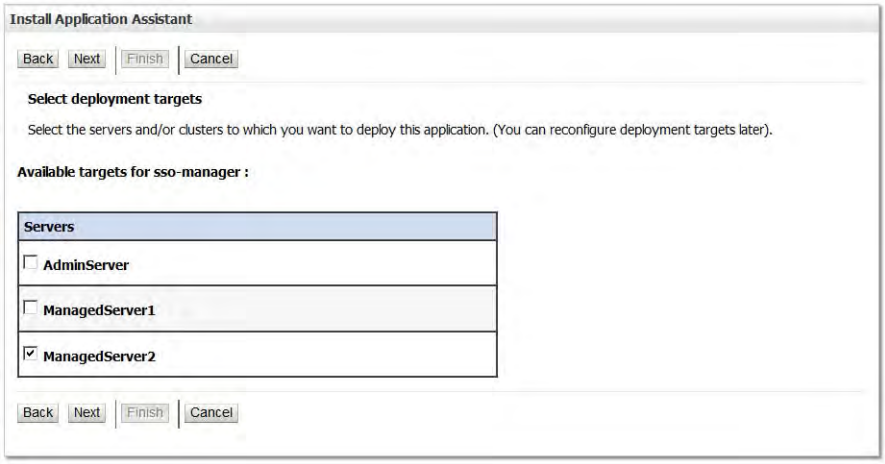
8. Select the `sso-manager.ear` file from the list.

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.

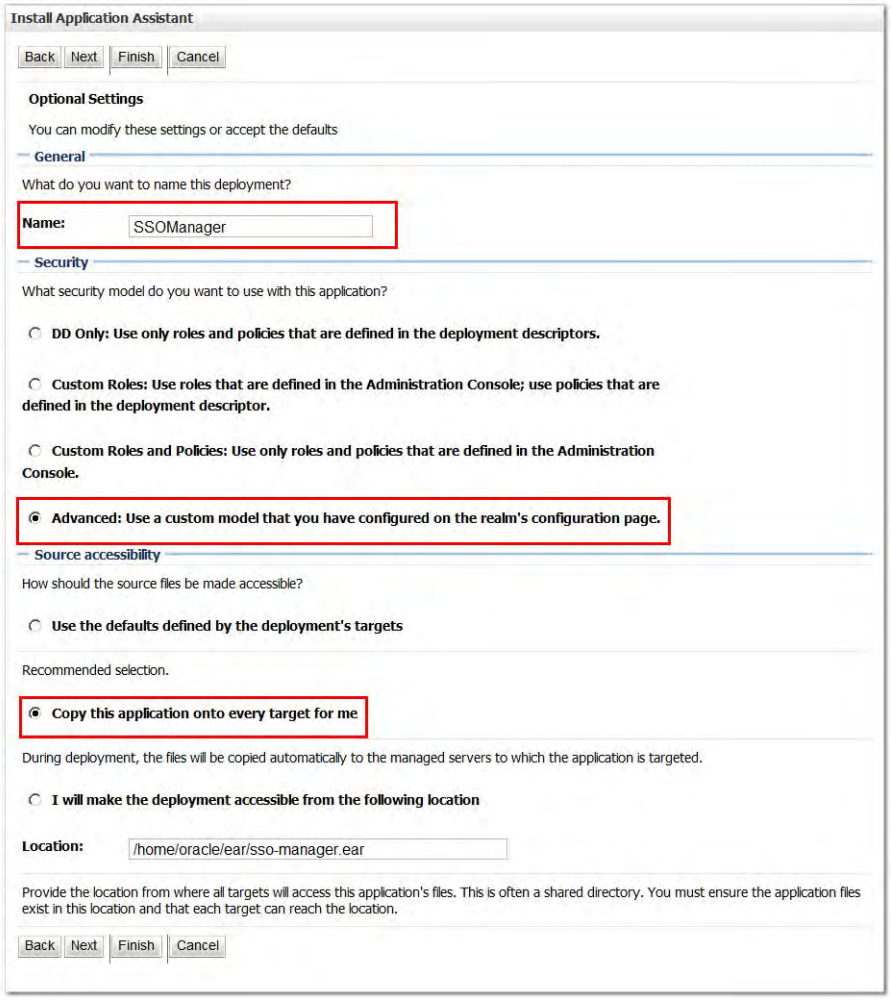


12. Select the server where the application should be deployed. (The application can be installed on an existing server.)

Note

SunGard Higher Education recommends deploying applications to a WebLogic Managed Server and not to the Administration Server. If you do not see the preceding page, you should check your WebLogic Server configuration to ensure that a Managed Server is available for deployment of applications. If a Managed Server is not available, the application will be deployed to the Administration Server, which is not a recommended configuration. For more information, consult the Oracle WebLogic Server Documentation Library.

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

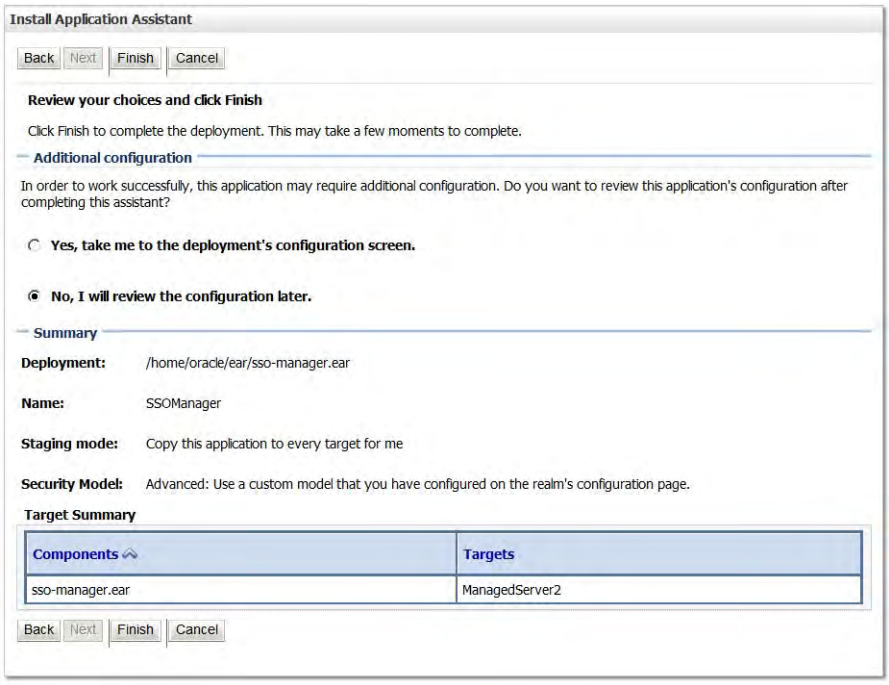


14. Enter a name for the application (for example, *SSOManager*) in the **Name** field.

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

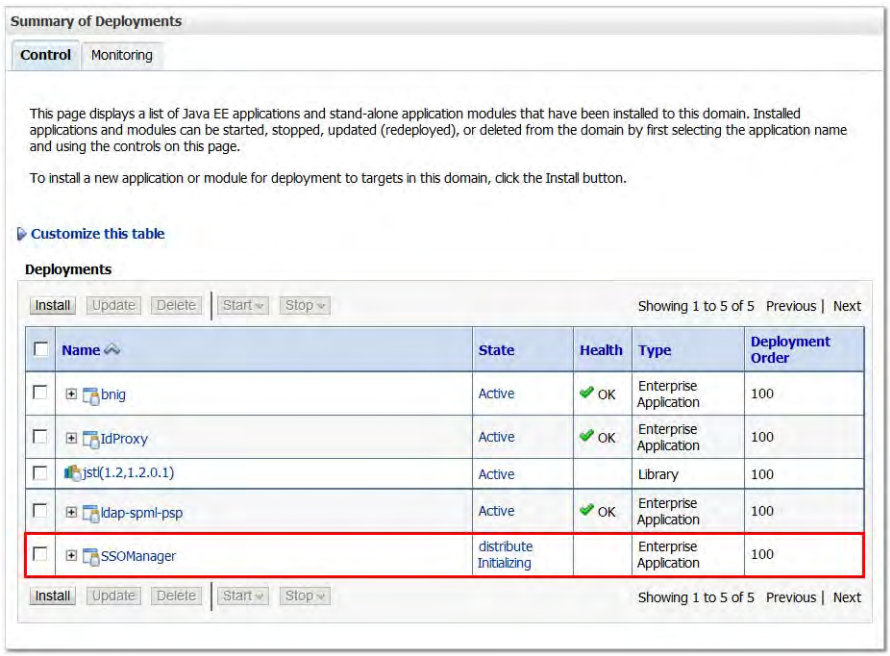
16. Select **Copy this application onto every target for me**.

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



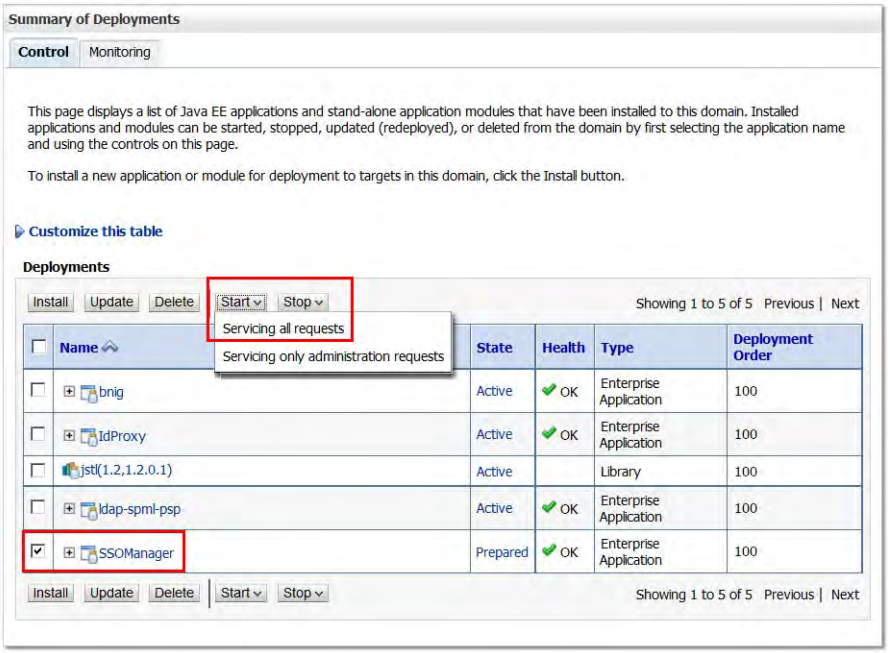
18. Select **No, I will review the configuration later**.

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



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

21. Start the newly deployed application as follows:



21.1. Select the newly deployed application.

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

21.3. Click **Yes**.

Migrate credentials (optional)

Use the following steps if you want to migrate credentials from an existing database schema to the newly created SSO Manager schema.

1. Extract the `sso-manager-oc4j-installer.jar` or the `sso-manager-weblogic-installer.jar` archive file.

2. Navigate to the directory where the file was extracted. This directory is referred to as `<SSOMGR_INSTALLER_HOME>`.

   ```
   cd <SSOMGR_INSTALLER_HOME>
   ```

3. Run SQL*Plus and connect as ssomgr.

4. Execute the `migrate_credentials.sql` script:

   ```
   sqlplus> @sql/migrate_credentials.sql
   ```

5. Exit SQL*Plus:

   ```
   sqlplus> exit
   ```

Configure the supporting components

The following components must be configured:

- Banner Web Tailor (for using the SSO Manager with Banner Self-Service)
- Oracle Forms server (for using the SSO Manager with Internet-native Banner)
- baniam.jar (for using the SSO Manager with Internet-native Banner)
- SSO Manager

Configure Banner Web Tailor for Self-Service Banner SSO

SSO for Self-Service Banner requires Banner Web Tailor:

- If Cascade is enabled, Banner Web Tailor 8.4.2 is the minimum version required.
- If Cascade is not enabled, Banner Web Tailor 8.3.1 plus patch `p1-dc06g9_twb8030101` is the minimum version required.

Use the following steps to edit the Banner Web Tailor parameters that are used by the SSO Manager.

1. Enter the Secure Area of Self-Service Banner.

2. Navigate to Web Tailor Administration.

3. From the Web Tailor menu, select Web Tailor Parameters.

4. For each of the following parameters, click the parameter name, enter the parameter value, and click **Submit Changes**.

Parameter	Description
IDMLOGINURI	CAS login URL (for example, *http://cas-server/cas/login)*.
IDMLOGOUTURI	Logout URL: • For a CAS implementation without Luminis® Platform 5.x, use *http://cas-server/cas/logout*. • For a CAS implementation using the embedded CAS server of Luminis Platform 5.x, use *http://server/c/portal/logout.* • For an implementation with a third-party access manager, use the URL established by the third-party access manager.
IDMTIMEOUT	Time in seconds that information contained in the `IDMSESSID` cookie is trusted. `IDMTIMEOUT`= *0* means session timeout is not enforced.
IDMSSO	Flag that determines whether SSO with BEIS is enabled (*Y*) or not enabled (*N*). If enabled, Self-Service Banner can be accessed via BEIS.
IDMCOOKIE	Name of the SSO Manager SSO cookie. Suggested value is *IDMSESSID.* This value must be the same as the **UDC ID Key** specified on the SSB Configuration page of the SSO Manager administrative interface.
IDMCOOKIEDOMAIN	Domain where the cookie is created (for example, *.university.edu*.). This value must be the same as the **Cookie Domain Name** specified on the SSB Configuration page of the SSO Manager administrative interface.
IDMCOOKIEPATH	Path where the cookie is created (for example, /).

Configure the Oracle Forms server for Internet-native Banner SSO

Your Oracle Forms server (OAS 10gR2 or Oracle WebLogic Server 11g) must be configured to support SSO for Internet-native Banner. You can modify settings directly in the configuration files, or you can use the Oracle Enterprise Manager console. The

following sections give instructions for both methods. Separate instructions are given for OAS and Oracle WebLogic implementation.

Modify settings directly in configuration files - OAS

Use the following steps if Internet-native Banner is implemented on OAS and you want to modify the settings directly in the configuration files. Assume that the Oracle Application Server home location is <OAS_HOME>.

1. Place the baniam.jar file on your forms server. Best practice is to place baniam.jar with the other jar files:

   <OAS_HOME>\forms\java

2. Navigate to <OAS_HOME>\forms\server\<Forms environment file>.

   The Forms environment file has an .env extension. This file is used to set variables (such as ORACLE_HOME, FORMS_PATH, and CLASSPATH) in the Oracle Forms runtime environment.

3. Modify the CLASSPATH variable in the Forms environment file to include a reference to baniam.jar. Locate this reference immediately before the reference to the repository.jar file as follows:

   ```
   CLASSPATH=OAS_HOME\j2ee\OC4J_BI_Forms\applications\formsapp\
   formsweb\WEB-INF\lib\frmsrv.jar;OAS_HOME\forms\java\baniam.jar;
   OAS_HOME\jlib\repository.jar;OAS_HOME\jlib\ldapjclnt10.jar;
   OAS_HOME\jlib\debugger.jar;OAS_HOME\jlib\ewt3.jar;OAS_HOME\
   jlib\share.jar;OAS_HOME\jlib\utj.jar;OAS_HOME\jlib\
   zrclient.jar;OAS_HOME\reports\jlib\rwrun.jar;OAS_HOME\forms\
   java\frmwebutil.jar
   ```

Warning

The order in this variable *is* important, because the baniam.jar is on the forms server at <OAS_HOME>\forms\java. ■

4. Navigate to <OAS_HOME>\forms\server\formsweb.cfg.

5. Add a new parameter to formsweb.cfg for passing the ticket to the Welcome to Banner Form (GUAINT).

   The following lines show an example snippet from the formsweb.cfg file that was used with initial testing. The new iamticket item is **bolded**. SunGard Higher Education recommends that this variable (iamticket) be added to the forms configuration similar to that of the following variable:

   ```
   # Other Forms runtime arguments: grouped together as one
   parameter.

   # These settings support running and debugging a form from the
   Builder:
   ```

```
otherparams=buffer_records=%buffer%
debug_messages=%debug_messages% array=%array% obr=%obr%
query_only=%query_only% quiet=%quiet% render=%render%
record=%record% tracegroup=%tracegroup% log=%log% term=%term%
iamticket=%iamticket%
```

Modify settings directly in configuration files - Oracle WebLogic Server

Use the following steps if Internet-native Banner is implemented on the Oracle WebLogic Server and you want to modify settings directly in the configuration files. Assume that the Oracle WebLogic Server home location is `<OAS_HOME>`.

1. Place the baniam.jar file on your forms server. Best practice is to place baniam.jar with the other jar files:

   ```
   <OAS_HOME>\forms\java
   ```

2. Navigate to the following file:

   ```
   <domain_home>/../<domain_name>/config/fmwconfig/servers/
   WLS_FORMS/applications/<forms_application_name>/config/<Forms
   environment file>
   ```

   The Forms environment file has an .env extension. This file is used to set variables (such as `ORACLE_HOME`, `FORMS_PATH`, and `CLASSPATH`) in the Oracle Forms runtime environment.

3. Modify the `CLASSPATH` variable in the Forms environment file to include a reference to baniam.jar. Locate this reference immediately after the reference to the `frmsrv.jar` file as follows:

   ```
   CLASSPATH=OAS_HOME\j2ee\OC4J_BI_Forms\applications\formsapp\
   formsweb\WEB-INF\lib\frmsrv.jar;OAS_HOME\forms\java\baniam.jar;
   OAS_HOME\jlib\ldapjclnt10.jar;OAS_HOME\jlib\debugger.jar;
   OAS_HOME\jlib\ewt3.jar;OAS_HOME\jlib\share.jar;OAS_HOME\jlib\
   utj.jar;OAS_HOME\jlib\zrclient.jar;OAS_HOME\reports\jlib\rwrun.
   jar;OAS_HOME\forms\java\frmwebutil.jar
   ```

Warning

The order in this variable *is* important, because the baniam.jar is on the forms server at `<OAS_HOME>\forms\java`.

4. Navigate to the following file:

   ```
   <domain_home>/../<domain_name>/config/fmwconfig/servers/
   WLS_FORMS/applications/<forms_application_name>/config/
   formsweb.cfg
   ```

5. Add a new parameter to `formsweb.cfg` file for passing the ticket to the Welcome to Banner Form (GUAINIT).

The following lines show an example snippet from the `formsweb.cfg` file that was used with initial testing. The new `iamticket` item is **bolded**. SunGard Higher Education recommends that this variable (`iamticket`) be added to the forms configuration similar to that of the following variable:

```
# Other Forms runtime arguments: grouped together as one
parameter.

# These settings support running and debugging a form from the
Builder:

otherparams=buffer_records=%buffer%
debug_messages=%debug_messages% array=%array% obr=%obr%
query_only=%query_only% quiet=%quiet% render=%render%
record=%record% tracegroup=%tracegroup% log=%log% term=%term%
iamticket=%iamticket%
```

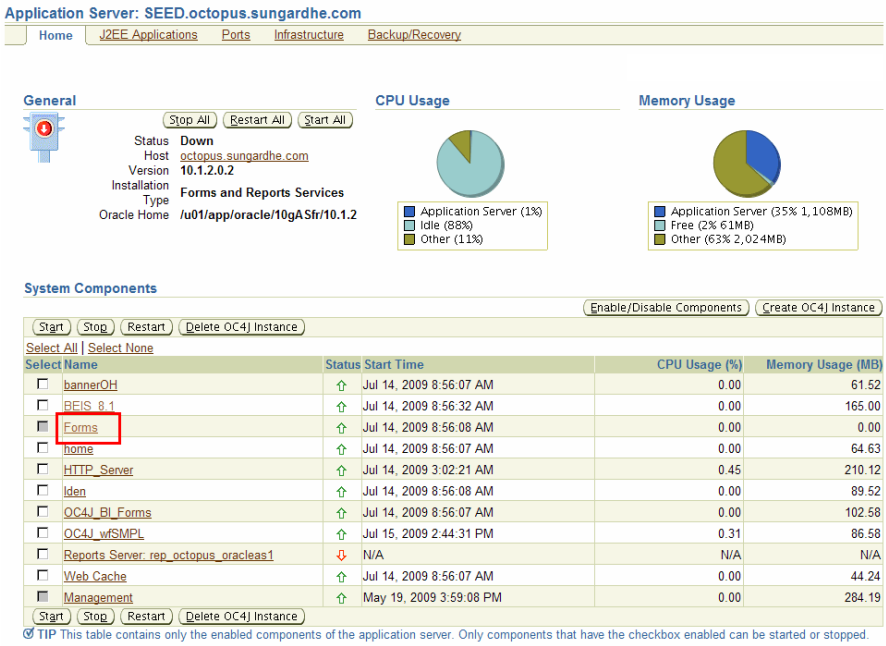
Modify settings using the console - OAS

Use the following steps if Internet-native Banner is implemented on OAS and you want to use the Oracle Enterprise Manager console to modify the configuration settings. Assume that the Oracle Application Server home location is <OAS_HOME>.

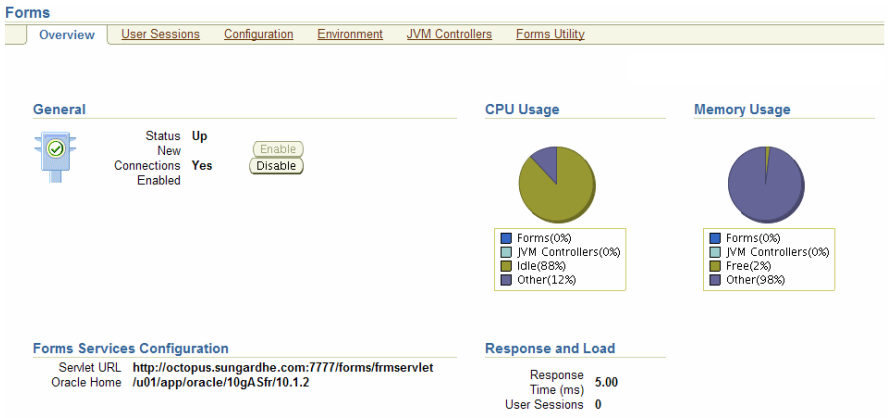
1. Place the baniam.jar file on your forms server. Best practice is to place baniam.jar with the other jar files:

   ```
   <OAS_HOME>\forms\java
   ```

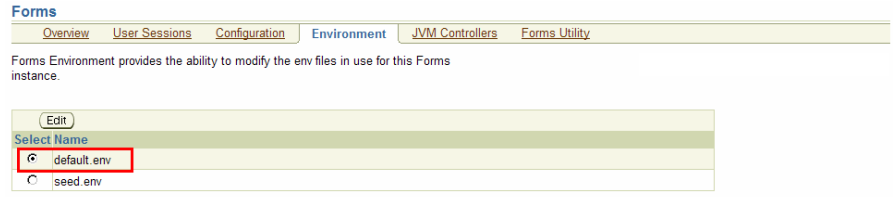
2. Connect to the Oracle Enterprise Manager. The Home page is displayed.



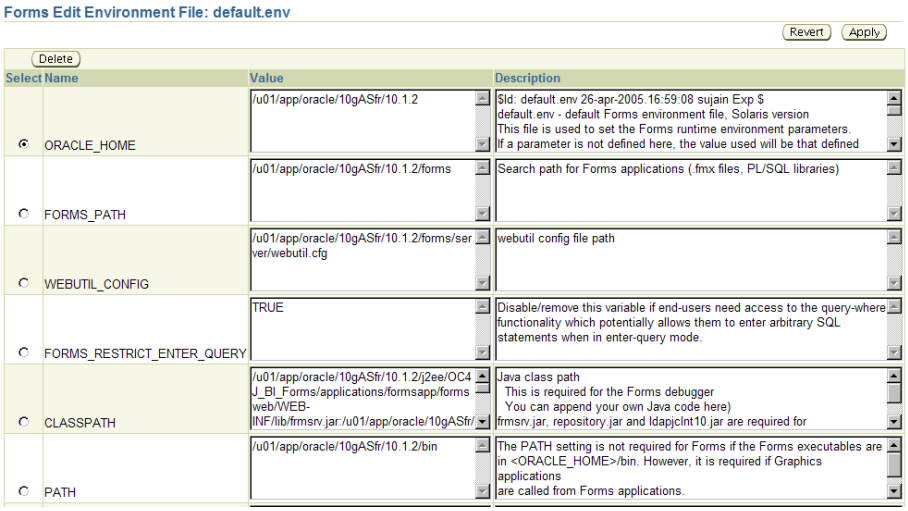
3. Click the **Forms** link in the list of system components. The Forms Overview page is displayed.



4. Select the **Environment** tab. The Forms Environment page is displayed.



5. Select the default or environment-specific name and click **Edit**. The Forms Edit Environment File page is displayed.



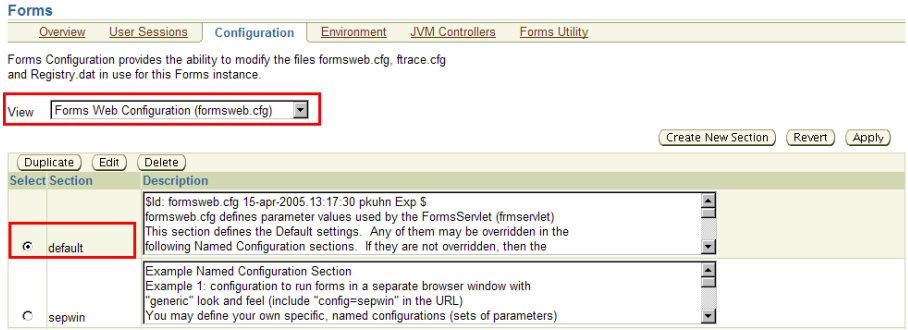
6. Scroll to and select the CLASSPATH definition.

7. Add the path to locate baniam.jar immediately after the `frmsrv.jar` entry.

8. Click **Apply**.

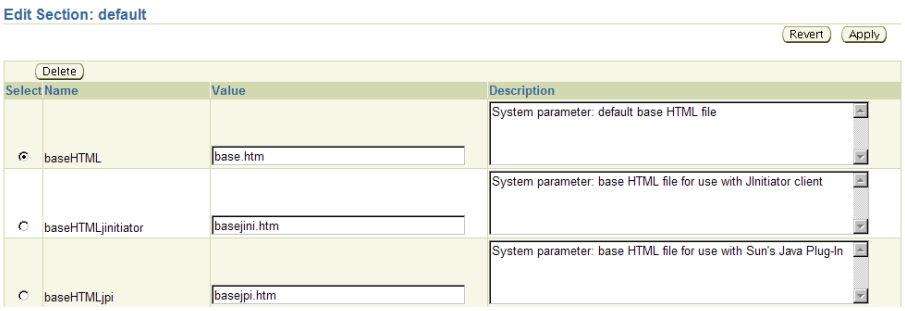
9. Navigate to the Forms page.

10. Select the **Configuration** tab.

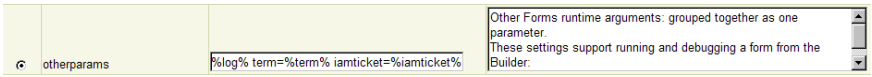


11. Select *Forms Web Configuration (formsweb.cfg)* in the **View** drop-down list.

12. Select the default or configuration-specific name and click **Edit**. The Edit Section page is displayed.



13. Scroll to and select the `otherparams` parameter.



14. Add `iamticket=%iamticket%` to the end of the string.

15. Click **Apply**.

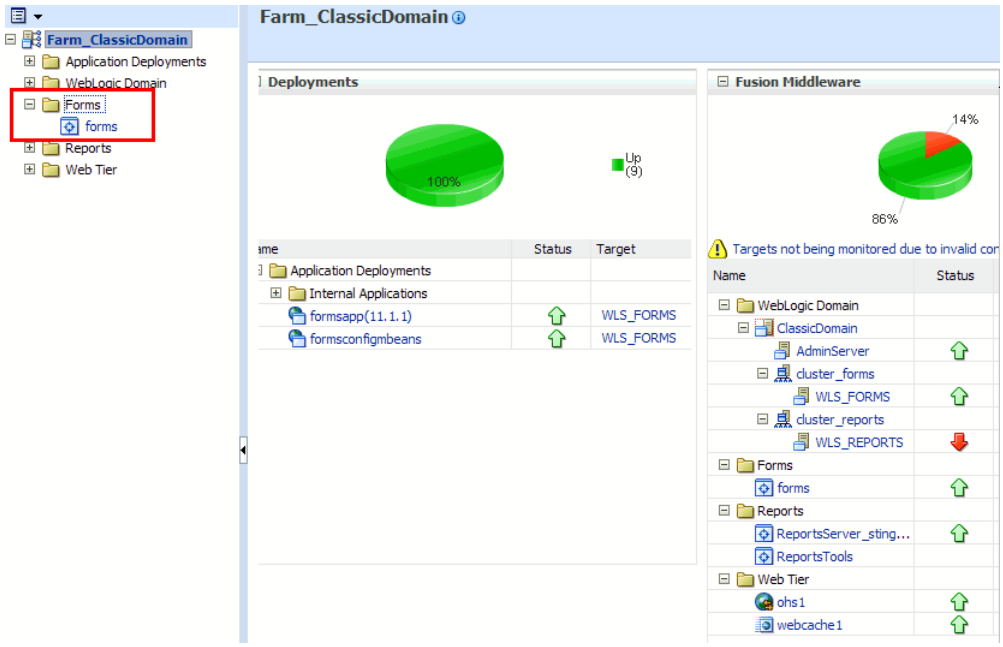
Modify settings using the console - Oracle WebLogic Server

Use the following steps if Internet-native Banner is implemented on the Oracle WebLogic Server and you want to use the Oracle Enterprise Manager console to modify the configuration settings. Assume that the Oracle WebLogic Server home location is `<OAS_HOME>`.

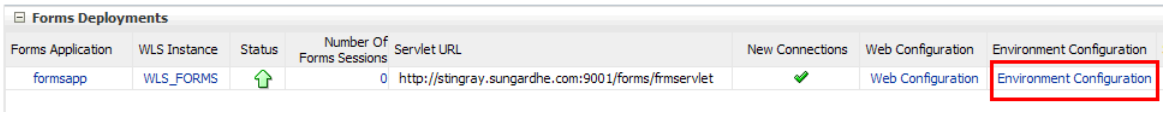
1. Place the baniam.jar file on your forms server. Best practice is to place baniam.jar with the other jar files:

   `<OAS_HOME>\forms\java`

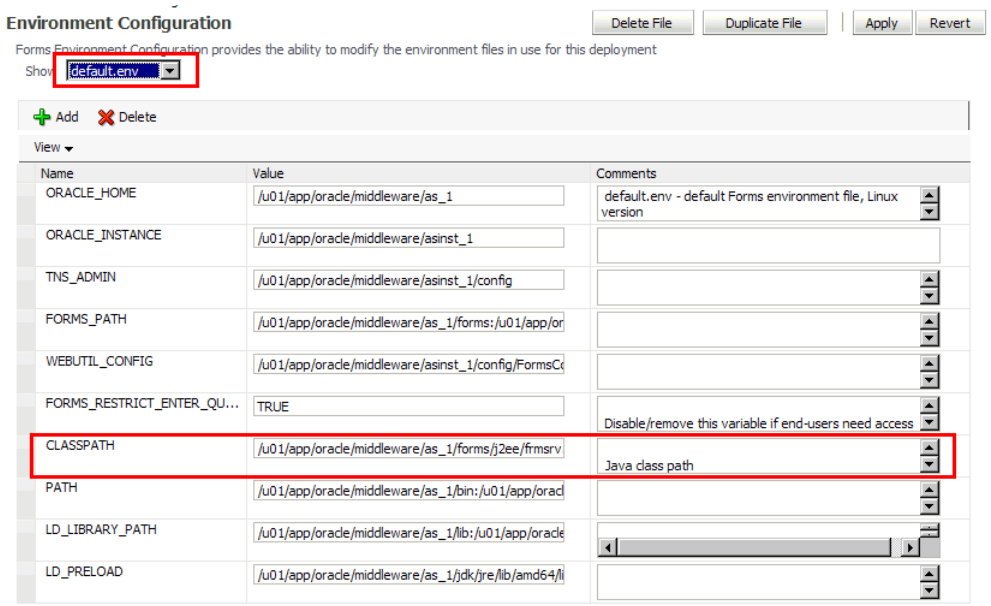
2. Connect to the Oracle Enterprise Manager. The console is displayed.



3. In the left pane, expand and click **Forms -> forms**.The Forms Deployments page is displayed.



4. Click the **Environment Configuration** link. The Environment Configuration page is displayed.



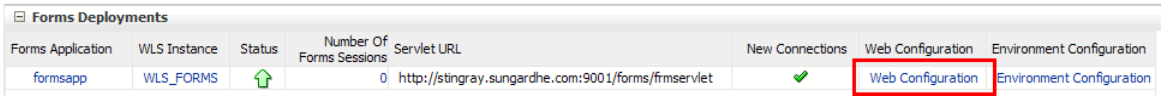
5. Select the default or environment-specific name from the **Show** drop-down list.

6. For the `CLASSPATH` variable, add a reference to baniam.jar. Locate this reference immediately after the reference to the `frmsrv.jar` file as follows:

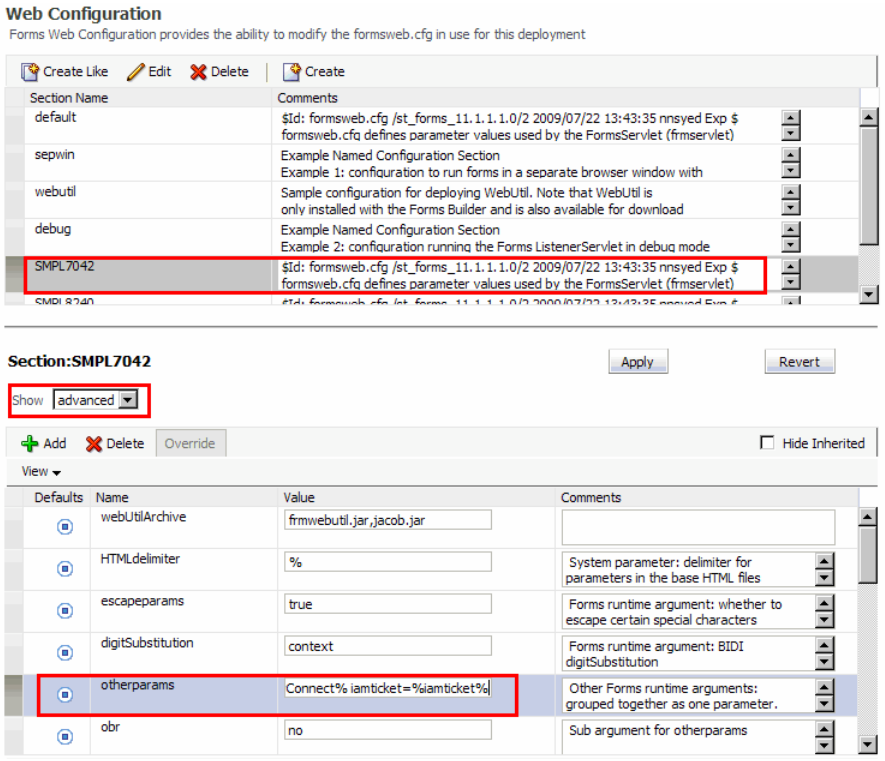
```
/u01/app/oracle/middleware/as_1/forms/j2ee/frmsrv.jar:/u01/app/
oracle/middleware/as_1/forms/java/baniam.jar:/u01/app/oracle/
middleware/as_1/jlib/ldapjclnt11.jar:/u01/app/oracle/
middleware/as_1/jlib/debugger.jar:/u01/app/oracle/middleware/
as_1/jlib/ewt3.jar:/u01/app/oracle/middleware/as_1/jlib/
share.jar:/u01/app/oracle/middleware/as_1/jlib/utj.jar:/u01/
app/oracle/middleware/as_1/jlib/zrclient.jar:/u01/app/oracle/
middleware/as_1/reports/jlib/rwrun.jar:/u01/app/oracle/
middleware/as_1/forms/java/frmwebutil.jar:/u01/app/oracle/
middleware/as_1/jlib/start_dejvm.jar:/u01/app/oracle/
middleware/as_1/opmn/lib/optic.jar
```

7. Click **Apply**.

8. In the left pane, expand and click **Forms -> forms**.The Forms Deployments page is displayed.



9. Click the **Web Configuration** link. The Web Configuration page is displayed.



10. In the **Section Name** field, select the name of the database where the forms are deployed.

11. Select *advanced* from the **Show** drop-down list.

12. For the `otherparams` variable, add *iamticket=%iamticket%* at the end of the string as follows:

    ```
    obr=%obr% record=%record% tracegroup=%tracegroup% log=%log%
    term=%term% ssoProxyConnect=%ssoProxyConnect%
    iamticket=%iamticket%
    ```

13. Click **Apply**.

Configure baniam.jar for Internet-native Banner SSO

The baniam.jar contains Java components that communicate with the Credential Web service to obtain the user credentials that are required for a user to gain access to Internet-native Banner. The baniam.jar uses Oracle Forms Java integration to allow Java classes to execute from the Oracle Forms server. A property file named `baniam.properties` supports the configuration for this component.

Use the following steps to configure baniam.jar to support SSO for Internet-native Banner.

1. Change your current working directory to the location of baniam.jar (`<OAS_HOME>/forms/java`).

2. Extract the `baniam.properties` file from baniam.jar:

   ```
   jar xvf baniam.jar baniam.properties
   ```

3. Edit `baniam.properties` and set the values as follows:

Property	Description
`service.endpoint`	URL endpoint for the Credential Web service, which is exposed by the SSO Manager.
`service.username`	Username used to authenticate to the Credential Web service. This is the same user defined to protect the SSO Manager.
`service.password`	Password used to authenticate to the Credential Web service.
`service.realm`	Realm name used to support HTTP basic authentication. Default value is one of the following: • OC4J - *jazn.com* • Oracle WebLogic - *myrealm* The value, however, must match the actual name used in the OC4J container or the name of the default realm listed in the Oracle WebLogic Server Administration Console's Security Realms page.
`<ApplicationName>.connection`	Optional parameter that specifies the application connection string used to connect to Oracle. `ApplicationName` is the name used to store the username and password in the Credential Web service. **Note:** SunGard Higher Education recommends that you disable this property by adding the comment character (#) in front of the property name.

Example

```
service.endpoint=http://<host>:<port>/ssomanager/ws/credential-
service

service.username=ssomgr

service.password=u_pick_it

service.realm=jazn.com

#Banner.connection=s10b80
```

4. Save `baniam.properties`.

5. Update the baniam.jar with the updated `baniam.properties` file:

   ```
   jar uvf baniam.jar baniam.properties
   ```

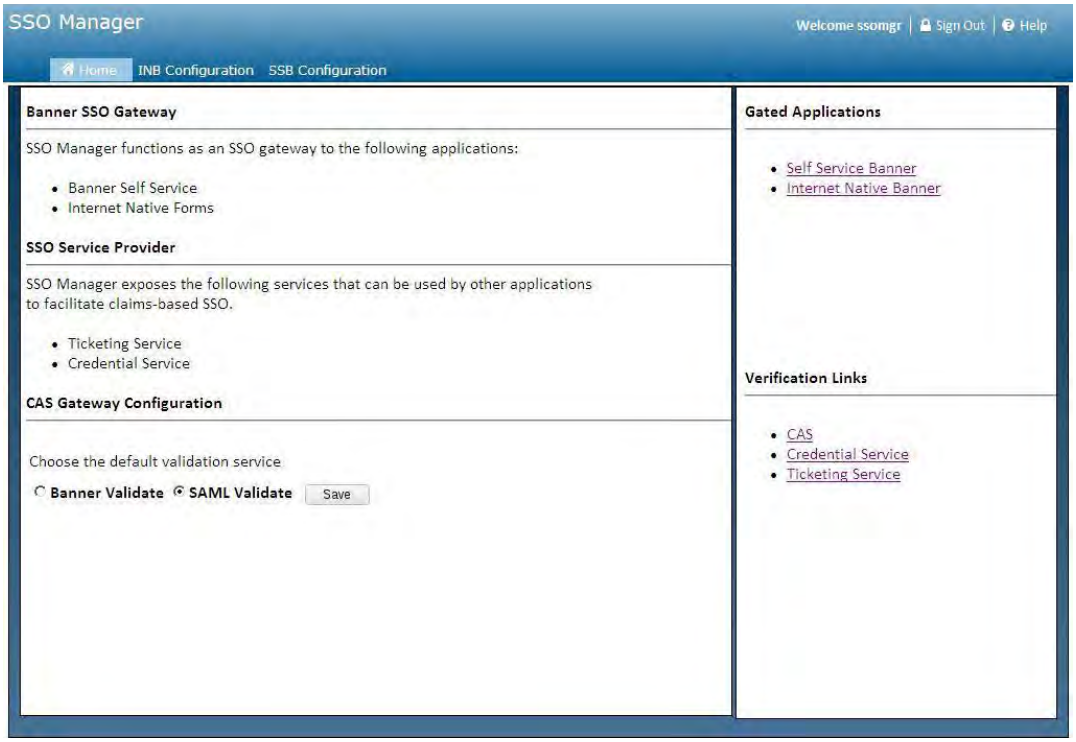
Configure the SSO Manager

Use the following steps to configure the SSO Manager.

1. Log in to the SSO Manager administrative interface:

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

   The SSO Manager Home page is displayed.

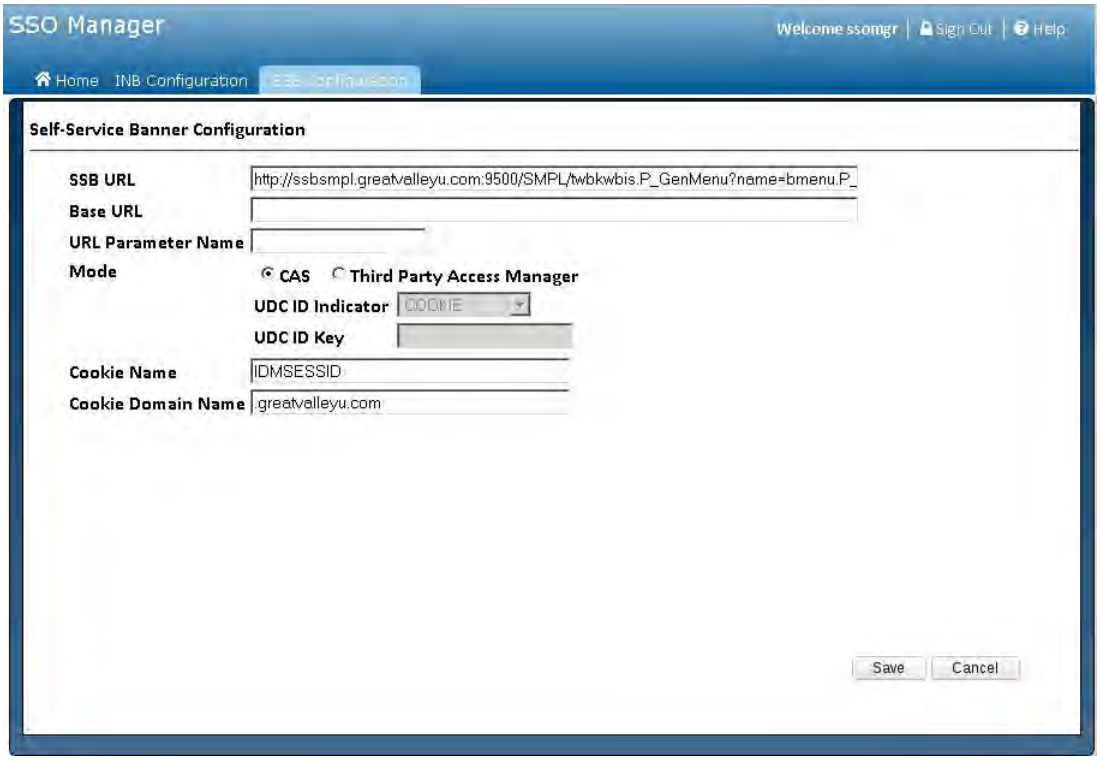


2. If CAS is being used as the central access manager, select one of the following radio buttons to identify the default validation service:

Banner Validate	The SSO Manager validation filter calls the /bannerValidate service to get information for the enterprise user. The /bannerValidate service validates the CAS service ticket and returns a UDCIdentity XML fragment response.
SAML Validate	The SSO Manager validation filter calls the /samlValidate service to retrieve the UDCIdentifier of the authenticated NetID via SAML (Security Assertion Markup Language).

3. Click **Save**.

4. Select the SSB Configuration tab. The Self-Service Banner Configuration page is displayed.



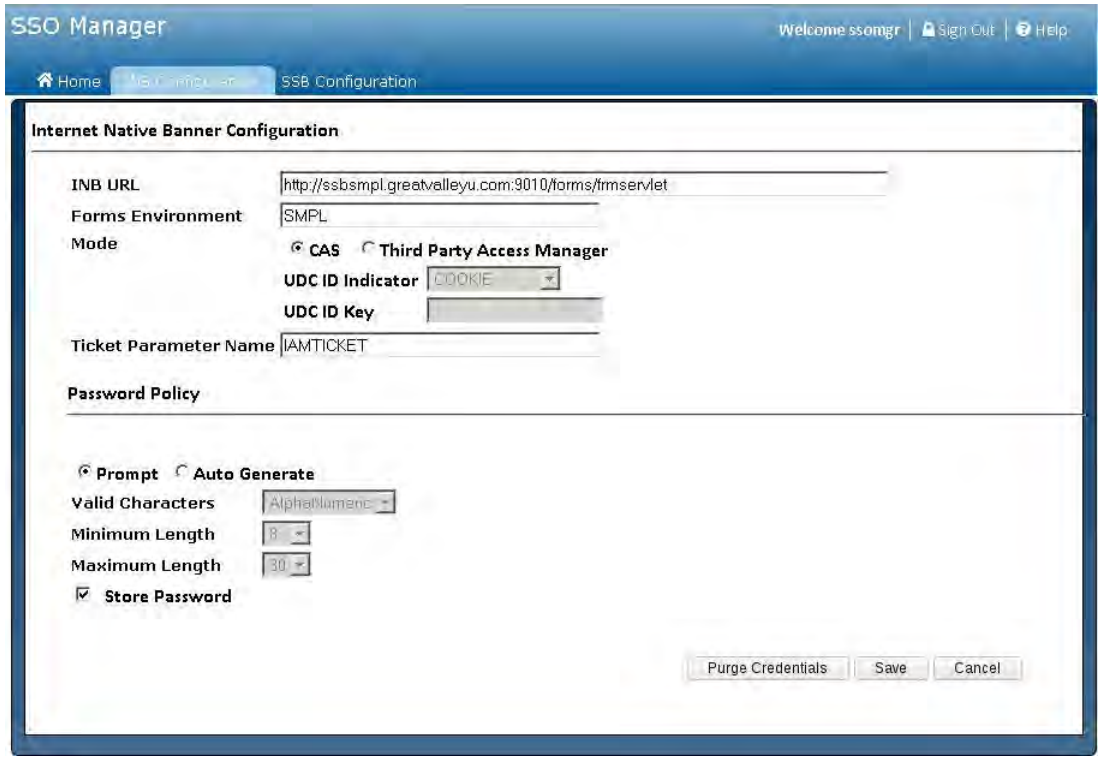
5. Enter the following information:

Field	Description
SSB URL	Default URL where users are redirected when they request the protected SSB URL. This is normally configured to be the Self-Service Banner main menu page, which typically looks like `http(s)://<host>:<port>/<dad-name>/twbkwbis.P_GenMenu?name=bmenu.P_MainMnu`.
Base URL	Base URL that is used to construct the full URL when a deep-linked page is requested from the SSO Manager. The base URL is the combination of the protocol, host, port, and dad-name for accessing Self-Service Banner: `http(s)://<host>:<port>/<dad-name>` The base URL does not identify a specific SSB page to access. See "Deep-linking" on page 11-92 for more information.
URL Parameter Name	Delimiter used to specify a target resource (packaged procedure, function, or URL) to the SSO Manager. The SSO Manager looks for this delimiter in deep-linked URL requests to determine the requested target resource. See "Deep-linking" on page 11-92 for more information.
Mode	Type of central access manager that authenticates users for SSO: *CAS* or *Third Party Access Manager*. **Note:** CAS 3.2.1.1, 3.3.1, and 3.4.2.1 are supported.
UDC ID Indicator	Method used to assert a user's identity (UDCIdentifier) to the SSO Manager: *COOKIE*, *HEADER*, or *PARAMETER*. **Note:** This parameter is only used for third-party access managers.
UDC ID Key	Name of the cookie, HTTP header, or parameter that is used to assert a user's identity (UDCIdentifier) to the SSO Manager. **Note:** This parameter is only used for third-party access managers.

Field	Description
Cookie Name	Name of the cookie that is used to assert the user's identity (UDCIdentifier) to SSB. The value of this parameter must equal the value of the IDMCOOKIE parameter that is defined in Banner Web Tailor. Suggested value is *IDMSESSID*. **Note:** The value of this parameter does not have to equal the value of the parameter that is specified in the Internet-native Banner SSO configuration.
Cookie Domain Name	Your institution's domain name or server domain name (for example, *.institution.edu*). This value must be the same as the IDMCOOKIEDOMAIN parameter that is defined in Banner Web Tailor.

6. Click **Save**.

7. Select the INB Configuration tab. The Internet Native Banner Configuration page is displayed.



8. Enter the following information:

Field	Description
INB URL	URL where users are redirected when they request the protected INB URL. This is the location of the INB server, which typically looks like `http(s)://<host>:<port>/forms/frmservlet`. When a valid SSO request is received, the SSO Manager redirects the user request to this location with the appropriate parameters. This document refers to this value as the `INB_URL`.
Forms Environment	Oracle Forms environment to which the SSO Manager is linked. This parameter is appended to the INB URL (for example, `http(s)://<host>:<port>/forms/frmservlet?config=<forms environment value>`). The configuration of this environment is normally specified through a Forms environment file of like name (for example, `smpl` and `smpl.env`).
Mode	Type of central access manager that authenticates users for SSO: *CAS* or *Third Party Access Manager*. **Note:** CAS 3.2.1.1, 3.3.1, and 3.4.2.1 are supported.
UDC ID Indicator	Method used to assert a user's identity (UDCIdentifier) to the SSO Manager: *COOKIE*, *HEADER*, or *PARAMETER*. **Note:** This parameter is only used for third-party access managers.

Field	Description
UDC ID Key	Name of the cookie, HTTP header, or parameter that is used to assert a user's identity (UDCIdentifier) to the SSO Manager. **Note:** This parameter is only used for third-party access managers and must be set to the same value that is established by the third-party access manager. For example, if this parameter has the value *UDC_ASSERT*, one of the following occurs: • If the **UDC ID Indicator** is set to *COOKIE*, the SSO Manager looks for a cookie called *UDC_ASSERT* to get the UDCIdentifier of the person. • If the **UDC ID Indicator** is set to *HEADER*, the SSO Manager looks for a HTTP header called *UDC_ASSERT* to get the UDCIdentifier of the person. • If the **UDC ID Indicator** is set to *PARAMETER*, the SSO Manager looks for an HTTP request parameter called *UDC_ASSERT* to get the UDCIdentifier of the person. If the SSO Manager cannot get the UDCIdentifer of the person, the application returns an HTTP 401 or 403 to indicate invalid credentials. **Note:** The value of this parameter does not have to equal the value of the parameter specified in the Self-Service Banner SSO configuration.

Field	Description
Ticket Parameter Name	Name of the HTTP request parameter that the SSO Manager creates to pass an INB ticket to the Oracle Forms server to perform SSO. This name must match the value configured in the Oracle Forms server (see "Configure the Oracle Forms server for Internet-native Banner SSO" on page 11-74). The default name *IAMTICKET* is used in this document. SSO with INB involves the exchange of an INB ticket between the SSO Manager and the Oracle Forms server. The ID of the INB ticket is passed via a parameter that is appended to the `INB_URL` in an HTTP request. INB (the Oracle Forms server and baniam.jar) is configured to look for this request parameter. The baniam.jar component extracts the ID of the INB ticket from the parameter and uses it to request the user's INB username and password from the Credential Web service to complete sign on by the Oracle Forms application. The following `INB_URL` shows the default Ticket Parameter Name and associated value appended: `http://<host>:<port>/forms/frmservlet?iamticket=<ticket parameter value>`
Password Policy	Indicator that determines how a password is created if the Credential Web service does not know a user's password: *Prompt* — Prompt user to enter a password. (The entered password must match the user's INB password.) *Auto Generate* — Automatically generate a password.
Valid Characters	Valid characters for generated passwords: *AlphaNumeric* — Both alphabetic and numeric characters *Numbers* — Numeric characters only *Characters* — Alphabetic characters only
Minimum Length	Minimum length of generated passwords. Must be numeric.
Maximum Length	Maximum length of generated passwords. Must be numeric.

Field	Description	
Store Password	Check box that determines whether INB user passwords are stored for future use.	
	selected	Store passwords for future SSO requests.
	cleared	Do not store passwords for future SSO requests.

9. Click **Save**.

Note

A server restart is *not* required for these entries to take effect. ■

Verify the configuration

After the SSO Manager is configured, use the following steps to verify that single sign on to INB and SSB is configured properly.

1. Select the Home tab.

   The Home page of the SSO Manager administrative interface is displayed.

2. In the right pane of the home page, click **CAS** under the Verification Links heading.

   The login page for the configured CAS server is displayed.

3. Verify that the correct server is accessed.

4. Log in to CAS using valid credentials for the environment.

   A CAS login success page is displayed.

5. Clear the cookies and close the browser.

6. Log in to the SSO Manager administrative interface again:

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

   The SSO Manager Home page is displayed.

7. In the right pane of the home page, click **Self Service Banner** under the Gated Applications heading.

   The login page for the configured CAS server is displayed.

8. Log in with valid credentials for a known SSB user.

   The main menu for SSB is displayed.

9. Clear the cookies and close the browser.

10. Log in to the SSO Manager administrative interface again:

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

    The SSO Manager Home page is displayed.

11. In the right pane of the home page, click **Internet Native Banner** under the Gated Applications heading.

    The login page for the configured CAS server is displayed.

12. Log in with valid credentials for a known INB user. The main menu for INB is displayed.

Application development with the SSO Manager

The SSO Manager contains services and features that can be used to develop single sign on capabilities for your institution's digital campus. This section describes how these features support single sign on, which may or may not involve Banner.

Deep-linking

Deep-linking is the ability to bypass a menu page and hyperlink directly to a specific page in Self-Service Banner or to a specific form in Internet-native Banner. The SSO Manager supports deep-linking in both SSB and INB through the protected URLs that it exposes for accessing these applications. Applications that need to deep-link into either SSB or INB need to supply the appropriate URL parameters that identify the page or form to which the user should be transferred.

Deep-linking to SSB pages

The default SSB URL is defined on the SSB Configuration page within the SSO Manager administrative interface. The browser is redirected to this default URL whenever the SSB URL that is exposed by the SSO Manager is requested.

Example

The SSO Manager is configured to access the Self-Service Banner protected main menu whenever SSB is requested. In a CAS-based environment, the following URL is used to access SSB:

`http://<host>:<port>/ssomanager/c/SSB`

Once the user is successfully authenticated, SSO Manager redirects the user to the Self-Service Banner main menu:

```
http(s)://<host>:<port>/<dad-name>/
twbkwbis.P_GenMenu?name=bmenu.P_MainMnu.
```

The SSO Manager also supports the ability to link to a specific SSB page other than the SSB main menu. This deep-linking is accomplished by adding parameters to the exposed URL to specify a target resource. The resource can be the combination of a database package and method (package.method), or it can be a complete URL.

To use this feature, the following parameters must be defined on the Self-Service Banner Configuration page within the SSO Manager administrative interface:

- **Base URL** - This URL is the combination of the protocol, host, port, and dad-name for accessing Self-Service Banner (`http(s)://<host>:<port>/<dad-name>`).

  Example: `https://ssbserver.institution.edu:9500/smpl/`

- **URL Parameter Name** - This is the delimiter in URL requests that identifies the target resource that is being requested. For example, if the value of the **URL Parameter Name** is *pkg*, the string "?pkg=" in URL requests determines the target resource.

The SSO Manager expects deep-link requests to SSB to use the following syntax:

```
<SSO Manager SSB URL>?<URL parameter name>=<target resource>
```

When a request for a resource is received, the SSO Manager evaluates the URL, looking for the delimiter specified as the **URL Parameter Name**. One of the following occurs:

- If the delimiter is not found, the SSO Manager redirects the browser to the default SSB URL.
- If the delimiter is found and the target resource is a URL, the SSO Manager redirects to the specified URL.
- If the delimiter is found and the target resource is not a URL, the SSO Manager constructs a redirect URL and redirects the browser to the constructed URL:

  ```
  <Base URL>/<target resource>
  ```

Example

SSB access for a CAS-based environment is configured as follows:

SSO Manager SSB URL	*https://beisserver.institution.edu:7777/ssomanager/c/SSB*
SSB URL	*https://ssbserver.institution.edu:9500/smpl/twbkwbis.P_GenMenu?name=bmenu.P_MainMnu*
Base URL	*https://ssbserver.institution.edu:9500/smpl/*
URL parameter name	*pkg*

A deep-link to the Update E-mail Addresses - Select Address page (`bwgkogad.P_SelectEmalUpdate`) in Self-Service Banner would be made to the SSO Manager as follows:

```
https://beisserver.institution.edu:7777/ssomanager/c/
SSB?pkg=bwgkogad.P_SelectEmalUpdate
```

A deep-link to a URL (for example, the URL for a different geographic locale) would be made as follows:

```
https://beisserver.institution.edu:7777/ssomanager/c/
SSB?pkg=https://ssbsmpl.greatvalley.edu:9500/frfr83/
bwgkogad.P_SelectEmalUpdate
```

The following request would redirect the browser to the SSB Main Menu, because the URL parameter name is not recognized:

```
https://beisserver.institution.edu:7777/ssomanager/c/
SSB?res=bwgkogad.P_SelectEmalUpdate
```

Deep-linking to INB forms

Deep-linking to specific Internet-native Banner forms is similar to deep-linking to Self-Service Banner pages. Deep-linking to INB forms, however, is simpler. Requests to the SSO Manager only need to include a parameter that specifies the desired form. The SSO Manager expects deep-link requests to INB to use the following syntax:

```
<SSO Manager INB URL>?otherParams=launch_form=<target form>
```

Oracle Forms does not require any configuration changes to accept this parameter.

Example

```
https://beisserver.institution.edu:7777/ssomanager/c/
INB?otherParams=launch_form=AFACAMP BAN_ARGS=CAMPAIGN::XCELL
```

If the requested form is not found, the Banner menu page is displayed.

Using the Ticketing Web service

The Ticketing Web service provides an operation that generates a ticket (a globally unique identifier) for a given credential or credential identifier. This service is exposed by the SSO Manager and works with the Credential Web service to provide credential information to Internet-native Banner. These services can be used to develop SSO support for applications that require the user's actual credentials for authentication. These services provide a secure mechanism for requesting and retrieving application-specific credentials.

The following URL exposes the Ticketing Web service:

```
http://<host>:<port>/ssomanager/ws/sso-ticket-service
```

The WSDL document for the Ticketing Web service is accessible via the **Ticketing Service** link under Verification Links on the home page of the SSO Manager administrative interface.

The Ticketing Web service exposes a single operation, CreateSSOTicket. This operation is used to create a ticket for the application credential that is associated with the UDCIdentifier provided in the request. The request contains the following tags:

- UDCIdentifier
- ApplicationName
- UserID

The response contains a single tag: SSOTicketID.

Sample request

```
<CreateSSOTicket>
   <UDCIdentifier>Banner_test</UDCIdentifier>
   <ApplicationCredential applicationName="Banner">
      <UserID>triddle</UserID>
   </ApplicationCredential>
</CreateSSOTicket>
```

Sample response

```
<ShowSSOTicket>
   <SSOTicket>
      <SSOTicketID>E8CE904BE89909FB5786938A0CFF209A
      </SSOTicketID>
   </SSOTicket>
</ShowSSOTicket>
```

Using the Credential Web service

The Credential Web service provides operations that store and retrieve credential information based on a ticket generated by the Ticketing Web service. The credential information is encrypted before being stored and decrypted when retrieved. This service is exposed by the SSO Manager and works with the Ticketing Web service to provide credential information to Internet-native Banner. These services can be used to develop SSO support for applications that require the user's actual credentials for authentication. They provide a secure mechanism for requesting and retrieving application-specific credentials.

The following URL exposes the Credential Web service:

```
http://<host>:<port>/ssomanager/ws/credential-service
```

The WSDL document for the Credential Web service is accessible via the **Credential Service** link under Verification Links on the home page of the SSO Manager administrative interface.

The Credential Web service includes the following operations:

- AddApplicationCredential
- GetApplicationCredential
- DeleteApplicationCredential

AddApplicationCredential

The AddApplicationCredential operation is used to add application credentials to the store. The request contains the following tags:

- UDCIdentifier
- ApplicationName
- UserID
- Password

The response contains a single tag called the Result**,** which has the value s*uccess* or *failure* depending on the result of the addition.

Sample request

```
<AddApplicationCredential>
   <UDCIdentifier>Banner_test</UDCIdentifier>
   <ApplicationCredential applicationName=“Banner”>
      <UserID>triddle</UserID>
      <Password>sct123</Password>
   </ApplicationCredential>
</AddApplicationCredential>
```

Sample response

```
<ConfirmAddApplicationCredential>
   <Result>success</Result>
</ConfirmAddApplicationCredential>
```

Error response

An error response is generated if the credentials are already present in the store.

```
<ConfirmAddApplicationCredential>
   <Result>failure</Result>
</ConfirmAddApplicationCredential>
```

GetApplicationCredential

The GetApplicationCredential operation is used to retrieve the application credentials based on the ticket generated using the Ticketing Web service. The request contains a single tag for the SSOTicketID. The response contains the following tags:

- Result
- ApplicationName
- UserID
- Password

Sample request

```
<GetApplicationCredential>
   <SSOTicket>
      <SSOTicketID>E8CE904BE89909FB5786938A0CFF209A
      </SSOTicketID>
   </SSOTicket>
</GetApplicationCredential>
```

Sample response

```
<ShowApplicationCredential>
   <Result>success</Result>
   <SSOCredentials applicationName="Banner">
      <UserID>triddle</UserID>
      <Password>sct123</Password>
   </SSOCredentials>
</ShowApplicationCredential>
```

Error response

The error response is generated by an invalid or expired ticket.

```
<ShowApplicationCredential>
   <Result>failure</Result>
</ShowApplicationCredential>
```

DeleteApplicationCredential

The DeleteApplicationCredential operation is used to delete the application credentials from the store. Invoking this operation causes all tickets corresponding to the application credentials to be deleted. The delete request contains the following tags:

- UDCIdentifier
- ApplicationName
- UserID

The response contains a single tag called the Result**,** which has the value s*uccess* or *failure* depending on the result of the deletion.

Sample request

```
<DeleteApplicationCredential>
   <UDCIdentifier>Banner_test</UDCIdentifier>
   <ApplicationCredential applicationName="Banner">
      <UserID>triddle</UserID>
   </ApplicationCredential>
</DeleteApplicationCredential>
```

Sample response

```
<ConfirmDeleteApplicationCredential>
   <Result>success</Result>
</ConfirmDeleteApplicationCredential>
```

Error response

The error response is generated when the requested credentials are not present in the store.

```
<ConfirmDeleteApplicationCredential>
   <Result>failure</Result>
</ConfirmDeleteApplicationCredential>
```

Incorporating CAS service validation

CAS authentication has the following steps:

1. The user tries to access an application URL.
2. Because the URL is protected by CAS, the user is redirected to the CAS login URL over an HTTPS connection, passing the name of the requested service as a parameter.
3. The user is presented with a CAS login dialog box.
4. The user logs in with his/her NetID and password.
5. CAS tries to authenticate the user. If authentication fails, the target application never hears about it, and the user remains at the CAS server. If authentication succeeds, processing continues with the next step.
6. CAS redirects the user to the target application, passing a parameter (ticket) to the URL.
7. The application opens an HTTPS connection, calls the CAS /serviceValidate URL, and passes the ticket and service name as parameters.

8. CAS checks that the supplied ticket is valid and is associated with the requested service. If validation is successful, CAS returns the CAS NetID to the application.

In addition to the NetID, the SSO Manager requires the UDCIdentifier to identify the user. Upon successful CAS authentication, the SSO Manager calls one of the following validation services to retrieve the UDCIdentifier of the authenticated user:

- **/bannerValidate** - This service validates the CAS service ticket and returns a UDCIdentity XML fragment response. This is a proprietary validation service. The CAS server must be configured to support the /bannerValidate service.
- **/samlValidate** - This service validates the CAS service ticket and returns a SAML (Security Assertion Markup Language) 1.1 ticket validation response that contains the UDCIdentifier. This service is provided natively in JA-SIG CAS starting with version 3.1.

When you configure the SSO Manager, you must choose to use /bannerValidate or /samlValidate as your validation service. The following sections provide more details about each validation service.

/bannerValidate

CAS attribute assertion features are used to implement the /bannerValidate service. This service performs the same checks as /serviceValidate. In addition to the NetID, however, /bannerValidate returns a UDCIdentity XML fragment response to the calling application.

SunGard Higher Education provides CAS version-specific extensions that implement /bannerValidate. Your version of BEIS determines whether an extension is required:

- For BEIS 8.1.4 and earlier, an extension is required. In these versions, a component of the Banner Identity Gateway uses /bannerValidate to retrieve the UDCIdentifier.
- For BEIS 8.1.5 and later, an extension is optional. The SSO Manager, available starting with BEIS 8.1.5, supports the /bannerValidate service and the native CAS /samlValidate service to retrieve the UDCIdentifier. The /bannerValidate service is being phased out in the SSO Manager and other SunGard applications. The /samlValidate service is recommended for any new applications that use CAS. See "/samlValidate" on page 11-102 for more details.

For applications that currently use /bannerValidate, note the following:

- /bannerValidate does not generate and issue proxy-granting tickets when requested.
- /bannerValidate must not return a successful authentication if it receives a proxy ticket.

/bannerValidate parameters

The following HTTP request parameters must be specified to /bannerValidate. They are case sensitive and are handled by /bannerValidate.

Parameter	Description
/BANNER-SV	Identifier of the service for which the CAS ticket was issued. Refer to http://www.ja-sig.org/products/cas/overview/protocol/index.html section 2.2 of CAS protocol for information pertaining to the login.
/BANNER-ST	Service ticket issued by the CAS login service when a client presents credentials. A service ticket is an opaque string that the client uses as a credential to access a service.

/bannerValidate responses

If the ticket validation is successful, /bannerValidate returns the following XML-formatted response:

```
<urn:UDCIdentity
xmlns:urn="urn:sungardhe:enterprise:domain:identity:1.0">
   <urn:UDCIdentifier>3C9CBAC307313872E0440003BA1015A4
   </urn:UDCIdentifier>
   <urn:LogonID>triddle</urn:LogonID>
   <urn:Extension>
      <urn:Attribute/>
      <urn:Attribute>
         <urn:name>BANNER-SV</urn:name>
         <urn:value>maldevl5.sct.com/cas-client-3.1/
         authorized/banner/SelfService</urn:value>
      </urn:Attribute>
   </urn:Extension>
</urn:UDCIdentity>
```

If the `UDC_IDENTIFIER` is not asserted as a part of the validation response, /bannerValidate returns an AssertionValidation error.

If the ticket validation fails, an HTTP 401 error code is returned.

The CAS server asserts the following attributes:

Attribute	Location in Response
`//UDCIdentity/UDCIdentifier`	`UDC_IDENTIFIER`
`//UDCIdentity/LogonID`	CAS NetID

/samlValidate

Starting with version 3.1, CAS server provides a validation service that releases additional information for an authenticated NetID.The /samlValidate service supports the standardized Security Assertion Markup Language (SAML) 1.1 protocol. SAML is an XML standard for exchanging authentication and authorization data between security domains.

Note

The /samlValidate service is a feature of CAS, not a feature of the SSO Manager or any other BEIS component.

An application creates a valid SAML request, wraps it in a SOAP envelope, and POSTs it to the CAS /samlValidate URL. The service checks that the ticket is valid and is associated with the target service. If validation is successful, /samlValidate returns a valid SAML response that contains the NetID and any identity attributes that the service is configured to return. The `<Attribute>` tags in the `<AttributeStatement>` section contain the additional information that was released for the authenticated NetID, which is found in the `<AuthenticationStatement>` section.

Example

```
<Response xmlns="urn:oasis:names:tc:SAML:1.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
IssueInstant="2011-08-13T10:56:13.378Z"
MajorVersion="1"
MinorVersion="1"
Recipient="http://jellyfish.greatvalleyu.com:7777/ssomanager/c/
SSB"
ResponseID="_df63003ff7bf6f645a2d08a2e2c2cc76">
  <Status>
    <StatusCode Value="samlp:Success"></StatusCode>
  </Status>
  <Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
  AssertionID="_306176fd94a8677cab6056dc3e8c892e"
  IssueInstant="2011-08-13T10:56:13.378Z"
  Issuer="localhost"
  MajorVersion="1"
```

```
MinorVersion="1">
  <Conditions NotBefore="2011-08-13T10:56:13.378Z"
  NotOnOrAfter="2011-08-13T10:56:43.378Z">
    <AudienceRestrictionCondition>
      <Audience>http://jellyfish.greatvalleyu.com:7777/
      ssomanager/c/SSB
      </Audience>
    </AudienceRestrictionCondition>
  </Conditions>
  <AttributeStatement>
    <Subject>
      <NameIdentifier>saisusr</NameIdentifier>
      <SubjectConfirmation>
        <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:
        cm:artifact</ConfirmationMethod>
      </SubjectConfirmation>
    </Subject>
    <Attribute AttributeName="UDC_IDENTIFIER"
    AttributeNamespace="http://www.ja-sig.org/products/cas/">
      <AttributeValue>2F10C881AC7D55942329E149405DC2F5
      </AttributeValue>
    </Attribute>
  </AttributeStatement>
  <AuthenticationStatement
  AuthenticationInstant="2011-08-13T10:56:13.347Z"
  AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:
  unspecified">
    <Subject>
      <NameIdentifier>saisusr</NameIdentifier>
      <SubjectConfirmation>
        <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:
        cm:artifact
        </ConfirmationMethod>
      </SubjectConfirmation>
    </Subject>
  </AuthenticationStatement>
 </Assertion>
</Response>
```

Use of the SSO Manager without BEIS account provisioning

One of the fundamental principles of Banner Enterprise Identity Services (BEIS) is the designation of a globally unique identifier (GUID), known as the UDCIdentifier, for persons in the enterprise. Whenever a new person record is created, a UDCIdentifier is assigned and stored in the central identity vault for the enterprise. The UDCIdentifier is then used to identify the user whenever access to protected resources is requested.

BEIS account provisioning assigns and stores a UDCIdentifier in Banner whenever a person record is created. BEIS-based provisioning also transports the generated UDCIdentifier to the central identity vault. By correlating the UDCIdentifier in Banner and in the central identity vault, BEIS-based provisioning facilitates single sign on. The SSO Manager asserts the UDCIdentifier that is stored in the central identity vault to Banner to identify the user after successful authentication. If the identifiers match, access is granted to the authenticated user.

To use the SSO Manager without BEIS account provisioning, you must correlate the GUID that is stored in Banner with the GUID that is stored in the central identity vault:

- GUID in Banner - The GUID for a person is stored in the GOBUMAP table. The column `GOBUMAP_UDC_ID` is VARCHAR2(225) and is designed to store the UDCIdentifier. Any unique identifier for the person can be stored in this column.
- GUID in central identity vault - The same GUID must be stored in the central identity vault so it can be retrieved by /bannerValidate or /samlValidate requests, wrapped in the appropriate XML tag. A baseline BEIS configuration sets the LDAP cn attribute with the value of the UDCIdentifier XML tag. The CAS server is configured to retrieve this value for both /bannerValidate and /samlValidate requests.

As long as the GUIDs stored in these two repositories are correlated and retrievable as the UDCIdentifier, access to SSB and INB through the SSO Manager is possible.

12 Middleware Validation

This chapter provides instructions for validating the installation and configuration of Oracle Streams, Banner® Identity Gateway, and Enterprise Identity Proxy Services.

Prerequisites

The following conditions must be met before you validate your installations and configurations:

- Components of Banner Enterprise Identity Services (BEIS) must be installed on Oracle Application Sever 10.1.3.4/5 (or later) or Oracle WebLogic Server 11g 10.3.2 (or later).
- The Oracle Application Server must be visible from a browser (Internet Explorer or Mozilla Firefox).
- Banner General 8.3.1 or higher must be installed.
- The Banner database must be enabled for Oracle Streams.
- Banner Identity Gateway and Enterprise Identity Proxy Services must be installed and configured.
- An SPML Provisioning Service Provider (PSP) must be available to receive the SPML provisioning messages.

Oracle Streams

BEIS uses Oracle Streams to capture Banner identity changes and propagate the changes to Banner Identity Gateway.

Validation

Use the following steps to validate the installation and configuration of Oracle Streams for BEIS.

1. Log in to the Banner database as the `streamsadmin` user.

2. Run the following SQL statements:

SQL Seq.	Validation SQL Statement	Result
1	`Select count(*) from GSVCARE where GSVCARE_CAPTURE_NAME='IAM_EVENTS_CAPTURE'`	=1
2	`Select count(*) from GSVAPDT where GSVAPDT_APPLY_NAME='IAM_EVENTS_APPLY'`	>=1
3	`Select count(*) from GSVCADT where GSVCADT_CAPTURE_NAME='IAM_EVENTS_CAPTURE'`	>=1
4	`Select count(*) from GSVRSRL where GSVRSRL_STREAMS_NAME ='IAM_EVENTS_CAPTURE'`	>=7
5	`Select count(*) from GSVRSRL where GSVRSRL_STREAMS_NAME ='IAM_EVENTS_APPLY'`	>=7

Note

The expected result of SQL Seq. 4 and 5 is the number of tables that the IAM process is monitoring. This is the number of records in GORCTAB where the SQPR code equals IAM. The minimum value should be 7, assuming a capture rule was added for GOBEACC. If you added capture rules for other tables, the result should increase accordingly.

If the result from an SQL statement does not match the result shown in the preceding table, there is an issue with the Oracle Streams configuration for BEIS. Refer to the following troubleshooting section for tips on resolving the issue.

Troubleshooting

Use the following steps to troubleshoot issues with the Oracle Streams configuration.

Error	Resolution
SQL Seq. 1, 2, and 3 failures These failures are common when a configuration step (such as the creation of a new Oracle Streams apply or capture process for `'IAM'`) is missed.	1. Log in to Banner as the `streamsadmin` user. 2. Execute the following SQL: `exec gp_streams_util.p_create_streams('IAM');`
SQL Seq. 4 and 5 failures These failures occur when the seed data for the capture process was not loaded and the process that converts the GUASADM seed data into Oracle Streams metadata was not run.	1. Locate the seed data scripts for Banner General 7.5.1 and reapply them. 2. Log in to Banner as the `streamsadmin` user. 3. Execute the following SQL: `exec gp_streams_util.p_configure_rules('IAM');`

Banner Identity Gateway

This application performs synchronization and reconciliation operations for Banner user identity.

Validation

Use the following steps to validate the installation and configuration of the Banner Identity Gateway.

1. Log in to the Banner Identity Gateway administrative interface (`http://<host>:<port>/bnigWeb/`) with the user name and password that were mapped to the `bnixadmin` role (OAS) or the `bnigAdminGroup` (Oracle WebLogic).

   1.1. If the login succeeds, the user and role/group required for Banner Identity Gateway are set up correctly.

   1.2. If the login fails, refer to Chapter 7, "Banner Identity Gateway" for information on configuring a user and role/group for the Banner Identity Gateway. Add the user and role/group. Repeat this step until login is successful.

2. Select Streams Admin from the menu bar.

3. Ensure that the Oracle Streams apply and capture processes are started and are in the following state:

   Capture process *ENABLED*
   Apply process *IDLE*

   If the capture and apply processes are not running, there are several reasons for failure. Two possible reasons follow:

Reason	Resolution
The configuration rules for the capture process might have changed through the use of the Streams Rules Configuration Form (GUASADM).	In the **Banner Streams Ops.** list, select *Load Capture Rules* and click **Apply** to create the capture metadata for Oracle Streams and start the capture and apply processes.
The configuration rules for the capture process were not changed.	In the **Banner Streams Ops.** list, select *Start Capture and Apply* and click **Apply** to start the capture and apply processes.

4. Launch Banner in a separate browser.

5. Open the General Person Identification Form (SPAIDEN).

6. Create a new Banner user, or query an existing Banner user on SPAIDEN.

7. Update information for the user. For example, change the middle name or update the user's primary address that is identified by the GTVSDAX rule.

8. Log in to Banner Identity Gateway administrative interface (`http://<host>:<port>/bnigWeb/`) as the administrative user.

9. Select Error Logger from the menu bar to see if there are any errors.

   9.1. If Banner General 7.5.1 was applied correctly, the Oracle Streams processes should trigger a new user identity (create/update) sync message. The Banner Identity Gateway receives this message, performs the appropriate transformation steps, and sends this message to a pre-configured JSM destination on the Oracle Application Server where the Banner Identity Gateway is installed.

   9.2. If an error occurs, a corresponding error message and its Banner identity event are logged to the database schema configured for the Banner Identity Gateway. You can view these error messages in the Error Logger of the Banner Identity Gateway administrative interface.

Troubleshooting

Two other errors might occur when a Banner identity message is not published:

Error	Resolution
A PL/SQL package that is needed by the Oracle Streams capture and apply processes is not compiling. Oracle Streams logs an error. Banner does not publish any sync messages as a result.	1. Ensure that the PL/SQL packages required for the capture and apply processes are valid and compiled. 2. Log in to the Banner Identity Gateway. 3. Select Streams Admin from the menu bar. 4. In the **Banner Streams Ops.** list, select *Start Capture and Apply* and click **Apply** to start the capture and apply processes. 5. Open Banner and update information for a Banner user on SPAIDEN. 6. Verify that the Banner Identity Gateway processed the change.
The permissions required for the `streamsadmin` account to start the Oracle Streams capture and apply processes are not correct. The capture and apply processes fail at startup.	Assign the correct privileges, if known, or contact the SunGard® Higher Education Action Line.

Enterprise Identity Proxy Services

The Enterprise Identity Proxy Services (Identity Proxy) application provides the services required to propagate identity to SPML-enabled applications. The validation steps are centered around the SPML Request Authority (RA), which is the core of the Identity Proxy.

Use the following steps to validate the Identity Proxy installation and configuration.

1. Log in to the Identity Proxy administrative interface (`http://<host>:<port>/IdProxyWeb`) with the user name and password that were mapped to the `idpadmin` role (OAS) or the `idpAdminGroup` group (Oracle WebLogic).

    1.1. If the login succeeds, the user and role/group required for the Identity Proxy are set up correctly.

    1.2. If the login fails, verify that the credentials provided are correct and that the application server is configured correctly to assign the appropriate users and roles to the application. Repeat this step until login is successful.

2. Select PST Configuration > Add/Update PST from the menu bar.

3. Verify that at least one active target under Provisioning Service Targets is configured.

4. Launch Banner in a separate browser.

5. Open the General Person Identification Form (SPAIDEN).

6. Create a new Banner user, or query an active Banner user on SPAIDEN.

7. Update information for the user. For example, change the middle name or update the user's primary address that is identified by the GTVSDAX rule.

    If Banner General 7.5.1 was applied correctly, the Oracle Streams processes should trigger a new user identity (create/update) sync message. The Banner Identity Gateway receives this message, performs the appropriate translation and transformation steps, and sends this message to a pre-configured JSM destination on the Oracle Application Server where the Banner Identity Gateway is installed. The Identity Proxy picks up the test message, if successful, and sends it to all active Provisioning Service Targets. The application logs the response from the PSP.

8. Log in to the Identity Proxy administrative interface (`http://<host>:<port>/IdProxyWeb)` as the administrative user.

9. Select SPML Provisioning > View Messages from the menu bar.

10. Verify that there is an entry in the Identity Proxy message log showing the request message sent to the PST and the response received from the PSP.

Note

These messages are stored in the database schema that is configured for the Identity Proxy. The table `T_UDC_SPML_MSG_LOG` holds all the request and response messages in an encrypted format. The messages are only visible from the Identity Proxy administrative interface. ■

13 Luminis Platform Configuration

Luminis® Platform, beginning with version 4.1, can be integrated with Banner® Enterprise Identity Services (BEIS). This integration enables the following functionality:

- CAS-based and third-party-based single sign on (SSO)
- Provisioning of user accounts in Luminis Platform 4.x via SPML 2.0 messages that contain the UDCIdentity XML structure

Note

Luminis Platform 5.x uses Banner Integration for eLearning, rather than BEIS, to provision user accounts.

This chapter includes the steps for configuring Luminis Platform with BEIS. For Luminis 4.x, this configuration enables SSO and account provisioning. For Luminis 5.x, this configuration enables SSO.

Prerequisite

A third-party enterprise identity management system (EIMS) must be installed before you configure Luminis Platform with BEIS. The EIMS contains and manages user IDS for the enterprise, and provisions all protected systems.

Luminis Platform supports the following EIMSs:

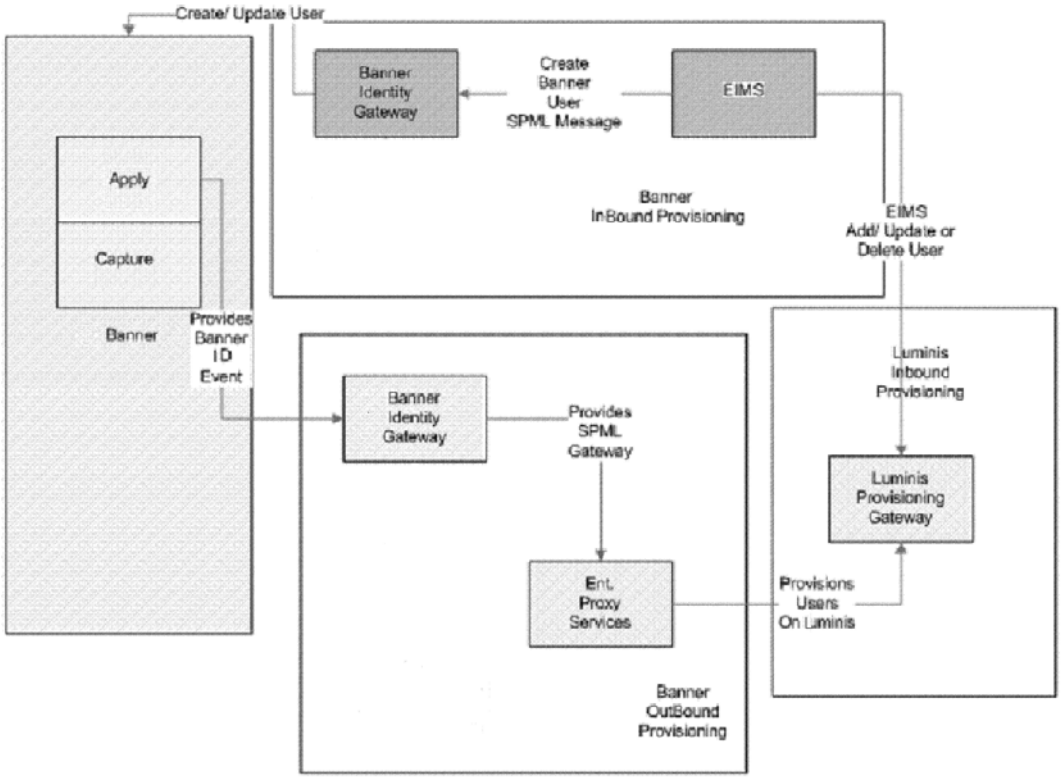
- Central Authentication Service (CAS) 3.2.1.1, 3.3.1, or 3.4.2.1 in conjunction with an LDAP repository. See Appendix D, “CAS Installation and Configuration” for customization details.
- Oracle Access Manager. This identity management solution is supported for Luminis Platform 4.x only.

Note

Luminis Platform 5.x does not support account provisioning with BEIS. If your institution has implemented Luminis Platform 5.x, you can use Banner Integration for eLearning to synchronize account data.

Information flow for account provisioning

The following illustration shows communication among the third-party EIMS, Luminis Platform 4.x, and Banner for account provisioning:



The requesting authority (either a third-party EIMS or Enterprise Identity Proxy Services) communicates with other systems using the industry standard Service Provisioning Markup Language (SPML). Additional provisioning activities occur between Banner and Luminis Platform using Banner Integration for eLearning to provision user role information.

Luminis Platform 4.x must be configured to receive SPML messages before identity management can be used. Luminis Platform 4.x provides a URL, username, and password to the requesting authority for SPML traffic.

Each identity management solution requires different tasks to install a Luminis Provisioning Gateway capable of communicating with it. "Luminis Platform configuration" on page 13-3 details the specific procedures for setting up the Luminis Provisioning Gateway for the two supported identity management solutions (Oracle Access Manager and CAS).

Luminis Platform configuration

Once an identity management solution with a central access manager is installed, Luminis Platform is configured using the following steps:

- Step 1, “Configure Luminis Platform properties”
- Step 2, “Set up and enable SSO”
- Step 3, “Verify the URL for the CAS server”
- Step 4, “Access the Luminis Provisioning Gateway”
- Step 5, “Migrate current users to be enterprise users”

Step 1 Configure Luminis Platform properties

To accomplish SSO, Luminis Platform must access two Web services that are delivered with the SSO Manager:

- Credential Web service, exposed by `http://<host>:<port>/ssomanager/ws/credential-service`
- Ticketing Web service, exposed by `http://<host>:<port>/ssomanager/ws/sso-ticket-service`

Use the configman tool to configure specific properties on Luminis Platform to enable access to the Credential and Ticketing Web services. The following tables list the properties that must be set for each Web service. Values for some properties must be obtained from the Banner administrator.

Note

Proxy configurations listed in the following tables are required only if there is a proxy server between Luminis Platform and the SSO Manager. This is not a typical configuration.

Credential Web service configuration

Command	Property	Property value
`configman -s`	`credentialservice.proxy.enable`	*false*
`configman -s`	`credentialservice.proxy.protocol`	Web protocol for the proxy the service is running (*http* or *https*)
`configman -s`	`credentialservice.proxy.host`	Host name of the server for the proxy where the service is running

Command	Property	Property value
`configman -s`	`credentialservice.proxy.port`	Port number of the proxy where the service is running
`configman -s`	`credentialservice.protocol`	Web protocol the service is running (*http* or *https*)
`configman -s`	`credentialservice.host`	Host name of the server where the service is running
`configman -s`	`credentialservice.port`	Port number where the service is running
`configman -s`	`credentialservice.urlpath`	Access URL to the service
`configman -s`	`credentialservice.username`	Service user name
`configman -s`	`credentialservice.password`	Service user name password

Ticketing Web service configuration

Command	Property	Property value
`configman -s`	`ticketservice.proxy.enable`	*false*
`configman -s`	`ticketservice.proxy.protocol`	Web protocol for the proxy the service is running (*http* or *https*)
`configman -s`	`ticketservice.proxy.host`	Host name of the server for the proxy where service is running
`configman -s`	`ticketservice.proxy.port`	Port number of the proxy where the service is running
`configman -s`	`ticketservice.protocol`	Web protocol the service is running (*http* or *https*)
`configman -s`	`ticketservice.host`	Host name of the server where the service is running
`configman -s`	`ticketservice.port`	Port number where the service is running
`configman -s`	`ticketservice.urlpath`	Access URL to the service

Command	Property	Property value
`configman -s`	`ticketservice.username`	Service user name
`configman -s`	`ticketservice.password`	Service user name password

Step 2 Set up and enable SSO

Several configuration operations and a system restart are required to set up and enable Luminis Platform to use single sign on with the central access manager. Most of these configuration operations are accomplished by running a script specific to the installed access manager:

Script	Description
`oracleidm`	Configures Luminis Platform to work with the Oracle Access Manager
`banneridm`	Configures Luminis Platform to work with CAS

These scripts are located in the `$CP_ROOT/bin` directory. The scripts stop the system, set the required system parameters, and restart Luminis Platform.

The following table lists the system parameters, descriptions, and possible parameter values set by the scripts for configuring Luminis Platform for single sign on with an access manager.

Note

Any time you change one of the following system parameters, you must stop and restart the Luminis Platform Web server.

Parameter	Description	Values Default
`credentialservice.proxy.enable`	Turns Credential Web service proxy on or off	*false*
`credentialservice.proxy.protocol`	Protocol for Credential Web service proxy	*https*
`credentialservice.proxy.host`	Host name for Credential Web service proxy	<host name>
`credentialservice.proxy.port`	Port for Credential Web service proxy	*80*
`credentialservice.protocol`	Protocol for Credential Web service	*http*

Parameter	Description	Values Default
`credentialservice.host`	Host name for Credential Web service	<host name>
`credentialservice.port`	Port for Credential Web service	*80*
`credentialservice.urlpath`	URL path to Credential Web service	*/ssomanager/ws/credential-service*
`credentialservice.username`	Username for Credential Web service	<username>
`credentialservice.password`	Password for Credential Web service	<password>
`ticketservice.proxy.enable`	Turns Ticketing Web service proxy on or off	*false*
`ticketservice.proxy.protocol`	Protocol for Ticketing Web service proxy	*https*
`ticketservice.proxy.host`	Host name for Ticketing Web service proxy	<host name>
`ticketservice.proxy.port`	Port for Ticketing Web service proxy	*80*
`ticketservice.protocol`	Protocol for Ticketing Web service	*http*
`ticketservice.host`	Host name for Ticketing Web service	<host name>
`ticketservice.port`	Port for Ticketing Web service.	*80*
`ticketservice.urlpath`	URL path to Ticketing Web service	*/ssomanager/ws/sso-ticket-service*
`ticketservice.username`	Username for Ticketing Web service	<username>
`ticketservice.password`	Password for Ticketing Web service	<password>
`udc.idm.enabled`	Global value that indicates Luminis Platform is running with an EIMS. Set by running the appropriate script.	*true*/*false* *false*

Parameter	Description	Values Default
`udc.idm.type`	EIMS being used. Set by running the appropriate script.	*oracle, banner, novell* no default
`udc.idm.logoutURL`	URL of the page displayed when the user logs out of identity management	*/jsp/misc/ LumOrIdm.jsp*
`udc.idm.cas. server.host`	Host name for the server that is running CAS. Set by running the `banneridm` script.	<server host name>
`udc.idm.cas. server.port`	Port used for the server that is running CAS. Set by running the `banneridm` script.	*443*
`udc.idm.user.page`	URL of the page displayed when the user logs out of Luminis Platform. Must be changed. Default value returns user to Luminis Platform login page.	<page URL> /
`udc.idm.cas. session.id`	Cookie name of UDCIdentifier when a channel is created for Banner in an identity management environment.	*IDMSESSID*
`idm.filter. headervar.udcid`	HEADERVAR configured in Oracle Access Manager for the UDCIdentifier. Used when Oracle Access Manager is the EIMS.	*LP_UCD_ID*
`idm.filter. headervar.user`	HEADERVAR configured in Oracle Access Manager for the user's login ID. Used when Oracle Access Manager is the EIMS.	*LP_LOGIN_ID*

Step 3 Verify the URL for the CAS server

If you are using CAS to implement SSO, you must make sure that the `banneridm` script uses the correct URL to the CAS server. The script uses *cas-server-3.1* as the default value for the CAS server URL. If your CAS server has a different URL, you must manually

change the script. Use the following steps to change the URL value that the `banneridm setcashost` option uses:

1. Open the `banneridm` script in a text editor.

2. Locate the part of the script where the CAS host is set:

```
#-------------------------------------------------------------
# Set CAS server host and port properties.
SetCasHost()
{
echo "# Setting CAS configman properties"
configman -s udc.idm.cas.server.host "$CAS_HOST_NAME"
configman -s udc.idm.cas.server.port "$CAS_HOST_PORT"
if [ -f $CAS_CLIENT_PROPERTIES ]
then
$AWK '
   /^cas.server.url=/ { print "cas.server.url=" url; next }
   /^cas.server.proxyCallbackUrl=/ { print
"cas.server.proxyCallbackurl=" callback; next }
   { print $0 }
   ' url="https://$CAS_HOST_NAME:$CAS_HOST_PORT/cas-server-3.1/
" \
     callback="https://$CAS_HOST_NAME:$CAS_HOST_PORT/manager/
proxy/Receptor"\
     $CAS_CLIENT_PROPERTIES > $CAS_CLIENT_PROPERTIES.new
   if [ $? -eq 0 ]
   then
       mv $CAS_CLIENT_PROPERTIES $CAS_CLIENT_PROPERTIES.old ||
exit 1
       mv $CAS_CLIENT_PROPERTIES.new $CAS_CLIENT_PROPERTIES
   fi
fi
}
```

3. Change *cas-server-3.1* to the value of your CAS server.

4. Save the script.

5. Run the `banneridm` script using the `setcashost` option:

```
banneridm setcashost -t host -r port
```

Step 4 Access the Luminis Provisioning Gateway

To provision users into Luminis Platform using SPML, send the SPML messages to the Luminis Provisioning Gateway at the following URL:

```
http(s)://<LP Web Server Host>:<LP Web Server Port>/spml/
services/psp/DocumentLiteral
```

The Luminis Provisioning Gateway uses basic authentication and requires the username and password of a Luminis Platform administrator.

Step 5 Migrate current users to be enterprise users

Luminis Platform is installed with local users (such as administrators, fragment owners, and some test users) that perform functions within Luminis Platform only. Local users do not have UDCIdentifiers, which are required for users to be a part of the identity and access management solution that controls and manages users for the entire institution. There is probably no need to convert these local users to enterprise users.

If your institution was running Luminis Platform before Banner Enterprise Identity Services (BEIS) was installed, your Luminis administrator must use an external program, supplied with Luminis Platform, to convert the non-local Luminis users to enterprise users with UDCIdentifiers. Use the following steps to migrate users.

1. Obtain a UDCIdentifier for each Luminis Platform user you want to transfer to the control of BEIS.

2. Log in to the Luminis Platform server with administrative privileges.

3. Open a command window (Cygwin on Windows servers).

4. Use a text editor to prepare a file that maps Luminis Platform IDs to UDCIdentifiers:

   ```
   user1=UDC_ID_0001
   user2=UDC_ID_0002
   user3=UDC_ID_0003
   ```

5. Save the file with a suitable file name and close it.

6. Enter the following command:

   ```
   cptool import udcids <file name>
   ```

   where `<file name>` is the name of the text file containing the mapping of each user's Luminis Platform login ID to the desired UDCIdentifier.

After Luminis Platform users are upgraded, their passwords and usernames are controlled through BEIS and cannot be changed through Luminis Platform.

After BEIS is implemented, new Luminis Platform users log in to the identity management server by default and access to Luminis Platform is handled through that server.

14 Banner Workflow Configuration

Banner® Workflow can be integrated with Banner Enterprise Identity Services (BEIS) beginning with Banner Workflow 8.0. This integration provides the following functionality:

- Provisioning of account information from an SPML Request Authority (RA) to Banner Workflow
- Single sign on (SSO) among applications protected by a central access manager

Prerequisites

The Banner Workflow Provisioning Gateway was tested to run in an Oracle Application Server 10.1.3.4/5 OC4J virtual machine using default OC4J VM settings. Specific hardware and software specifications depend on the Oracle Application Server and platform that are selected.

A version of Java 5 (build 1.5.x) is required to run the ant-based installation script. In most cases, the jdk bundled as part of Oracle Application Server can be used.

Account provisioning

A central identity vault stores the enterprise definition of identity for user accounts. When standard business processes add, change, or delete information in this identity vault, account information in enterprise applications such as Banner Workflow is automatically added, changed, or deleted. This processing is called account provisioning.

Information flow

As a result of identity events, an SPML Request Authority (RA) dispatches requests to add, update, delete, or look up user accounts in appropriate systems. The RA uses standard Service Provisioning Markup Language (SPML) messages, encapsulating identity information in the UDCIdentity XML structure, to make these requests. The Banner Workflow Provisioning Gateway receives the SPML messages and determines how a Banner Workflow instance should respond to the request. When properly configured, a message results in a corresponding user account being created, updated, or deleted in Banner Workflow.

All provisioned accounts in Banner Workflow are distinguished by a UDCIdentifier. This value is displayed as the external ID in the Banner Workflow User Management view. An error occurs if a duplicate SPML add request is forwarded to Banner Workflow with the same UDCIdentifier.

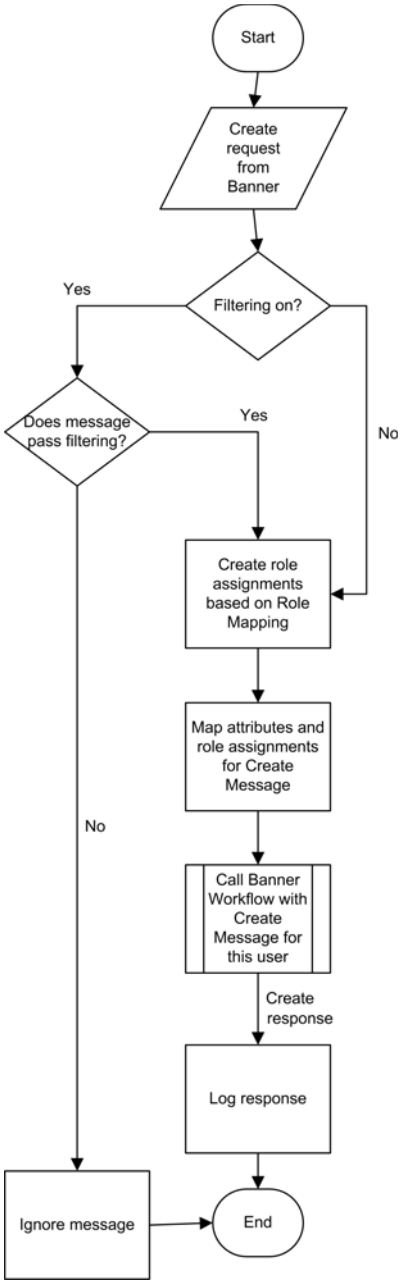
Add or update messages to Banner Workflow either pass completely or fail completely. There are no incomplete adds or updates. For example, the Banner Workflow Provisioning Gateway creates an add or update message for Banner Workflow partly based on the role mappings specified. If the role mappings specify a role or organization that does not exist in Banner Workflow, and the Banner Workflow Provisioning Gateway sends a message containing role assignments based on those mappings, the add or update action fails completely. No other properties are updated if the role assignments fail. For an add request, the account is not created if the role assignments fail.

If Banner Workflow is inaccessible or if there is any error in Banner Workflow while processing the message, the Gateway does not resend a message.

The following sections describe the processing for add, update, and delete requests in more detail.

Add request

If an SPML add request is received, the Banner Workflow Provisioning Gateway communicates with a Banner Workflow instance to create a Banner Workflow user account for the identity. This assumes that the identity in the add request is eligible to be a Banner Workflow user. The Gateway can be configured to filter account creation requests if the identity matches a specific institution role. In addition, the Gateway can determine, based on the institution roles of the identity, which initial Banner Workflow roles should be assigned to the Banner Workflow user.



Update request

When an SPML update request is received, the Banner Workflow Provisioning Gateway checks to see if the user specified in the request exists in Banner Workflow. The subsequent processing depends on whether filtering is enabled.

If filtering is enabled, the following processing occurs:

- If the user does not exist in Banner Workflow, the Gateway checks to see if the SPML update request has a match in the qualifying roles. If there is a match, the update message is converted to an add message and sent to Banner Workflow.
- If the user exists in Banner Workflow, the Gateway checks to see if the user possesses a qualifying role in the SPML message. If yes, the update request is treated as a normal update and sent to Banner Workflow. If the SPML update request no longer contains a qualifying role, the update request is converted to a delete message and sent to Banner Workflow.

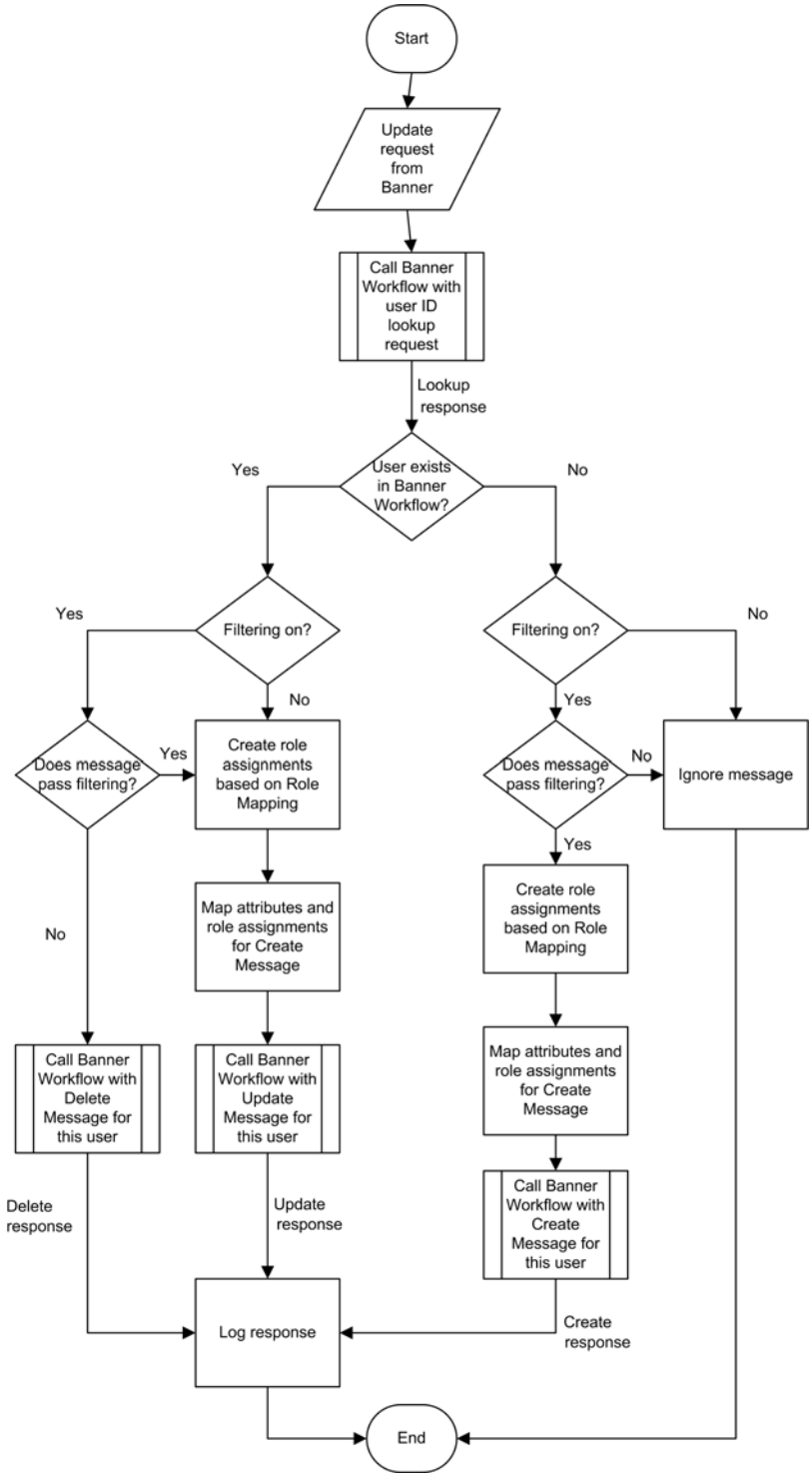
If filtering is disabled, the following processing occurs:

- If the user does not exist in Banner Workflow, the update message is ignored.
- If the user exists in Banner Workflow, the update request is sent to Banner Workflow.

The Banner Workflow Provisioning Gateway treats an update as a complete replacement of the data. Any manual changes made to the Banner Workflow user account outside of BEIS are lost the next time an update message arrives.

Example

User account UserA is created in Banner Workflow through BEIS and the Banner Workflow Provisioning Gateway. A Banner Workflow administrator signs in to Banner Workflow and assigns new roles for UserA. Then, the last name of UserA is changed. An update message is sent to the Banner Workflow Provisioning Gateway without any knowledge of the new role assignments that were manually created for UserA. The update message has the new last name and the role assignments that are present in the role mappings. The Gateway pushes this message to Banner Workflow. After this update, UserA loses the role assignments that the administrator assigned manually.

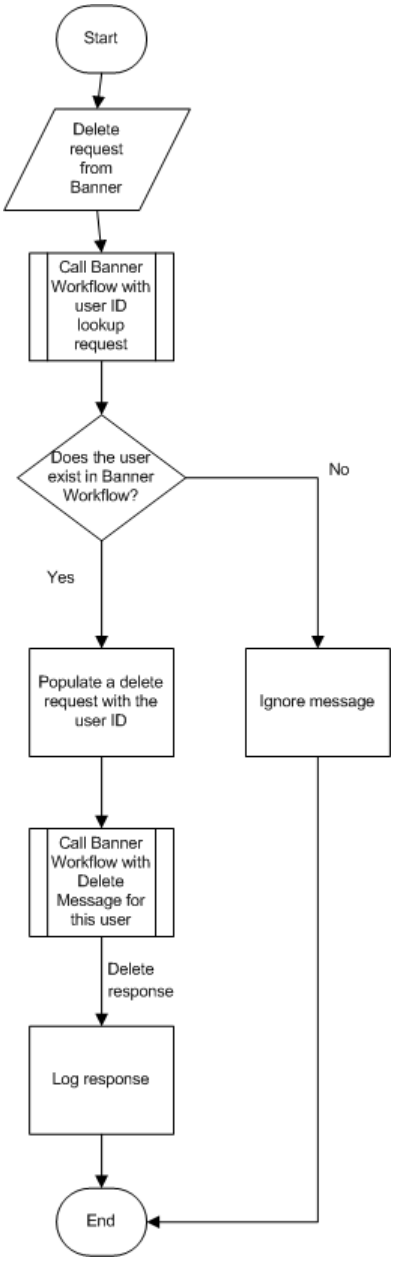

Start
Update request from Banner
Call Banner Workflow with user ID lookup request
Lookup response
User exists in Banner Workflow?
Yes
No
Filtering on?
Yes
No
Filtering on?
No
Yes
Does message pass filtering?
Yes
Create role assignments based on Role Mapping
Does message pass filtering?
No
Ignore message
No
Yes
Map attributes and role assignments for Create Message
Create role assignments based on Role Mapping
Call Banner Workflow with Delete Message for this user
Call Banner Workflow with Update Message for this user
Map attributes and role assignments for Create Message
Delete response
Update response
Call Banner Workflow with Create Message for this user
Log response
Create response
End

Delete request

If an SPML delete request is received, a delete message is sent to Banner Workflow. Banner Workflow performs a soft delete by disabling the user account. The user ID is preserved.

Note

This functionality is true only when the delete message is sent from the Gateway. If a user account is deleted manually from Banner Workflow, the user account is deleted permanently and cannot be retrieved.



Configuration of the Banner Workflow Provisioning Gateway

The Banner Workflow Provisioning Gateway is configured by using the following steps:

- Step 1, "Install the Banner Workflow Provisioning Gateway"
- Step 2, "Configure the Banner Workflow Provisioning Gateway"
- Step 3, "Build ear file"
- Step 4, "Deploy ear file"
- Step 5, "Configure Enterprise Identity Proxy Services"

Step 1 Install the Banner Workflow Provisioning Gateway

Use the following steps to install the Gateway.

1. Create a directory for the Banner Workflow Provisioning Gateway installation. This directory is referred to as `WFP_HOME` in the following instructions.

2. Copy the `installer.jar` (for Banner Workflow provisioning) to the `WFP_HOME` directory.

3. Use the jar tool to expand the jar file:

   ```
   $JAVA_HOME/bin/jar xf installer.jar
   ```
   or
   ```
   %JAVA_HOME%\bin\jar xf installer.jar
   ```

   The expanded `installer.jar` contains the following:

`<WFP_HOME>`	Directory of the expanded Banner Workflow provisioning installer archive
`<WFP_HOME>/apache-ant-1.7.0`	Apache ant scripting engine used to drive the build scripts
`<WFP_HOME>/dist`	Location of `wfprovisioning.ear`, which is created during ear assembly
`<WFP_HOME>/resources/config`	Location of configuration files that an administrator updates and maintains
`<WFP_HOME>/resources/ear`	Static resources needed to assemble the ear
`<WFP_HOME>/wardir`	Static resources needed to assemble the ear

Step 2 Configure the Banner Workflow Provisioning Gateway

Three files in `<WFP_HOME>/resources/config` need to be modified:

- `server.properties` for connectivity
- `log4j.properties` for logging
- `configuration.xml` for business rules

You are only required to customize `server.properties` and `configuration.xml`.

Note

To reflect a change from any of the configuration files, a new `wfprovisioning.ear` needs to be generated (Step 3, "Build ear file") and deployed to the application server (Step 4, "Deploy ear file"). ■

server.properties

This file defines the connectivity to the Gateway and assigns the target Banner Workflow endpoint to propagate, create, modify, delete, and look up requests. The following property values must be assigned.

Property	Description
`psp_username=wfproot` and `psp_password=password`	Basic http authentication credentials needed to connect to the Gateway instance.
`workflowPrincipal=wfwebservices`	Client user account used to make Web service requests into Banner Workflow. This is typically the wfwebservices (WebServicesUser) but can be any user account granted permissions to make Web services calls into Banner Workflow.
`workflowCredential=password`	Password for the user account used to process Web service requests into Banner Workflow.
`workflowEndpointAddress=http\://localhost\:8888/workflow`	Distinguishing URL to the Banner Workflow instance. Format: `<protocol>://<host>:<port>/<web context name>`

log4j.properties

This file controls the output of log messages coming from the Gateway. You do not have to customize this file.

configuration.xml

This file controls the role mapping and filtering features of the Banner Workflow Provisioning Gateway. This file is located in the `<WFP_HOME>/resources/config` directory. Before deploying the Banner Workflow Provisioning Gateway for the first time, this file must be updated to reflect the desired role mappings and filtering options.

Role mapping

In Banner Workflow, users participate in workflows for which they are assigned a role. These roles do not need the same role names and functions that are present in the EIMS. To accommodate this scenario, administrators and analysts can map roles received by the Banner Workflow Provisioning Gateway to equivalent Banner Workflow roles. This mapping is specified in the `configuration.xml` file.

When the Banner Workflow Provisioning Gateway receives an add or update SPML message, the message can specify user roles in the InstitutionRole element. The Gateway retrieves the string value of the InstitutionRole element and searches the `configuration.xml` file to see if there is an equivalent Banner Workflow role defined. If a match is found, the Gateway adds the Banner Workflow role and organization name to the message and sends the message to Banner Workflow. If no match is found in the `configuration.xml` file, that role is ignored.

The element structure for role mappings in the `configuration.xml` file is as follows:

```
<RoleMappings>
   <RoleMapping>
      <InstitutionRole role="analyst"/>
      <WorkflowRole organization="root" name="Analyst" />
   </RoleMapping>
   <RoleMapping>
      <InstitutionRole role="admin"/>
      <WorkflowRole organization="root" name="Admin" />
   </RoleMapping>
   <RoleMapping>
      <InstitutionRole role="WFuser@OrganizationA"/>
      <WorkflowRole organization="OrganizationA"
      name="WFUser" />
   </RoleMapping>
</RoleMappings>
```

Multiple RoleMapping elements can be specified in the RoleMappings element. Each RoleMapping element has an InstitutionRole and a WorkflowRole element:

- The InstitutionRole element value should match the role name that is specified in the SPML message. In a Multi-Entity Processing (MEP) environment, the role name is followed by a "@" symbol and the organization name. The "@" is the delimiter.
- The WorkflowRole element is the equivalent role in Banner Workflow for that InstitutionRole. The WorkflowRole element has two properties. It is important to confirm that values specified in the `configuration.xml` file for these properties are present in Banner Workflow.
  - The organization property refers to the organization where the role is relevant. Every Banner Workflow role assignment has an organization assigned. Even if MEP is not enabled in Banner Workflow, a Banner Workflow organization needs to be specified.

Note

Typically this is "root," which is the default name of the top level organization in Banner Workflow. If you are specifying a sub-organization in Banner Workflow, qualify the sub-organization with the name of the root organization and a period. For example, to specify the financial aid department you might have an organization equal to "root.Financial Aid". ■

- The name property holds the name of the Banner Workflow role.

The Banner Workflow Provisioning Gateway allows for mapping a single institution role to multiple Banner Workflow role and organization combinations.

Example

```
<RoleMappings>
   <RoleMapping>
      <InstitutionRole role="admin"/>
      <WorkflowRole organization="root" name="Analyst" />
   </RoleMapping>
   <RoleMapping>
      <InstitutionRole role="admin"/>
      <WorkflowRole organization="root" name="Admin" />
   </RoleMapping>
</RoleMappings>
```

If the Gateway receives an SPML add or update message that has an institution role of admin, then the Gateway adds role assignments of Admin and Analyst at the root organization in the message that it sends to Banner Workflow.

If there is no match for the SPML institution role, the Gateway ignores that role when creating a message for Banner Workflow.

Filtering

The Banner Workflow Provisioning Gateway filters messages and prevents some messages from reaching Banner Workflow. This is useful in cases where some identities in Banner are provisioned but should not be Banner Workflow participants. For example, a workflow acts upon student identities, but student identities do not get direct access to the Banner Workflow console or to an individual worklist. Restricting user accounts in Banner Workflow to active participants improves the scalability and performance of Banner Workflow.

The Filter element determines whether filtering is enabled. The child elements define the set of qualifying institution roles that indicate a given identity should be a workflow member. The Banner Workflow Provisioning Gateway checks if the message passes filtering before performing any role mappings.

This configuration is specified in the `configuration.xml` file:

```
<Filter enabled="true">
   <QualifyingRoles>
      <InstitutionRole role="workflowUser"/>
   </QualifyingRoles>
</Filter>
```

The InstitutionRole child element specified under the QualifyingRoles element should match the string value of the InstitutionRole element specified in the SPML message received by the Gateway. In the case of MEP-enabled Banner Workflow, the InstitutionRole element in `configuration.xml` has a role value of role name without the "@" delimiter or the MEP code. Multiple institution roles can be specified under the QualifyingRoles element. This means that filtering can be based only on the existence of roles and not on roles at a specific organization.

Note

This set of qualifying roles is a different list than the set of role mappings that are defined in the same `configuration.xml` file. It is possible that the institution roles defined under the QualifyingRoles element are not defined under the RoleMappings element.

Filtering can be enabled or disabled. If the filter is disabled, then adds, updates, and deletes are processed as is. With a filter property set to off, the Gateway does not convert updates to adds or deletes in any situation.

Step 3 Build ear file

Execute the following command:

```
<WFP_HOME>/ant ear
```

This operation packages a new ear file by bundling the changes made to the resources configuration files. The new file is located in `<WFP_HOME>/dist/wfprovisioning.ear`.

Note

Whenever you change any file in the resources directory, you should repeat this step and re-deploy the ear file. ■

Step 4 Deploy ear file

The ear file created by the installer must be deployed to an OC4J instance in the Oracle Application Server. The file should be deployed to a new instance that has no other application deployed to it. Use the following steps to deploy the ear file.

1. Use the `createinstance` script found in the Oracle Application Server `bin` directory to create the OC4J instance. Consult the Oracle documentation for specific instructions for creating an OC4J instance.

   ```
   <ORACLE AS HOME>\bin createinstance -instanceName
   wfprovisioning -groupName wfprovisioning
   ```

2. Deploy the ear file by selecting `<WFP_HOME>/dist/wfprovisioning.ear` for deployment. If you are manually deploying within the iAS Enterprise Manager Console, select **Automatically create a new deployment plan** for the Deployment Plan.

3. On the Deploy: Application Attributes page, modify the context root as required to maintain your applications (by default 'wfprovisioning'), proceed to the next step, and complete the deployment.

Step 5 Configure Enterprise Identity Proxy Services

Once the Banner Workflow Provisioning Gateway is deployed, it must be registered as a Provisioning Service Provider (PSP) in the Enterprise Identity Proxy Services so that identity messages can be published to it. Use the following steps to configure the Identity Proxy.

1. Access the Identity Proxy:

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

2. Log in with the user name and password for the `idpadmin` role.

3. Select PSP Configuration > Add/Update PSP from the menu bar. The Provisioning Service Provides page is displayed.

4. Configure the following properties:

PSP Location	Location and name where you deployed the Workflow Provisioning Gateway, followed by `/xfire/psp/DocumentLiteral`. ***Example*** `http://internalServer.edu:8888/wfprovisioning/xfire/psp/DocumentLiteral`
User Name	Same value as the `psp_username` that was defined in the `Server.properties` file
Password	Same value as the `psp_password` that was defined in the `Server.properties` file
Auth Realm	*Workflow Provisioning Gateway*
Enabled	*True*, to enable messaging to the Banner Workflow Provisioning Gateway

5. Click **Add/Update**.

Single sign on (SSO) authentication

Banner Workflow supports two methods for integrating with an external SSO configuration:

- Central Authentication Service (CAS)
- Third-party access manager such as the Oracle Access Manager

The following sections describe both methods for setting up SSO.

SSO under CAS

Central Authentication Service (CAS) can be used to provide a SSO solution. In this SSO implementation, Banner Workflow delegates authentication to the CAS 2.0 server that has been extended to include the UDCIdentifier to support the Banner Validator service specified in the configuration.

If the user is not logged in, Workflow redirects the user to the CAS login page.

Note

The external ID of the Banner Workflow user must be set to the UDCIdentifier of the Banner Workflow user.

The following steps are used to configure Banner Workflow under CAS:

- Step 1, “Configure Banner Workflow for CAS”
- Step 2, “Configure <LogoffUrl> element (optional)”
- Step 3, “Set up Oracle Application Server SSL”
- Step 4, “Register Banner Workflow with CAS server”
- Step 5, “Configure Luminis Platform for Banner Workflow”
- Step 6, “Modify Banner Forms technology type”

Step 1 Configure Banner Workflow for CAS

Use the following steps to configure Banner Workflow for CAS.

1. Modify the Banner Workflow `configuration.xml` file as follows:

   ```
   Authentication mode - "CAS"
   ```

2. Modify the following properties in the CAS section of the same `configuration.xml` file:

Property	Description
`ServiceURL`	Banner Workflow URL used to service requests from the CAS server. Format is `<protocol>:<workflow host>:<workflow port>/<workflow root>/j_spring_cas_security_check` ***Example*** `http://school.edu:7777/workflow/j_spring_cas_security_check`
`LoginUrl`	URL used to log in to CAS
`AuthenticationProviderKey`	Key required by Banner Workflow’s CAS authentication provider to identify tokens that it previously authenticated
`ProxyTicketValidatorUrl`	URL of CAS. ***Example*** `http://school.edu:7777/cas`

Step 2 Configure <LogoffUrl> element (optional)

When running Banner Workflow in CAS mode, the Logoff link does not log a user out of Banner Workflow because the user is still logged in to CAS. If you wish to provide a link to an external URL to log users out of CAS, set the optional LogoffUrl element as the last child element of SecurityIntegration.

Step 3 Set up Oracle Application Server SSL

CAS servers run under SSL (Secure Sockets Layer). You must register the SSL certificate used by the CAS server into the keystore file (`cacerts`) that the Oracle Application Server uses for the java installation. This is typically done by using the keytool command.

Example:

```
ORACLE_AS_HOME\jdk\jre\bin\keytool -import -file my_certificate
-keystore ORACLE_AS_HOME\jdk\jre\lib\security\cacerts
```

`ORACLE_AS_HOME` is the path of the server, and `my_certificate` is the path of the certificate file for the CAS server.

The keytool prompts for a password, which is typically `'changeit'` for a default java installation.

Step 4 Register Banner Workflow with CAS server

After you configure and deploy Banner Workflow under CAS, register the Banner Workflow deployment with the CAS server the same way you register any other CAS application. See your CAS server documentation for more details.

Step 5 Configure Luminis Platform for Banner Workflow

Use the following steps to configure Luminis® Platform for Banner Workflow.

1. If SafeUTFURL is enabled to prevent XSS attacks, either add Banner Workflow to the allowed list or disable SafeUTFURL.

   Example

   If Banner Workflow is deployed to a host named `sample.sungardhe.edu` using http, then it might be necessary to manually add `sample.sungardhe.edu` to the allowed list in both normal and in encoded form:

   ```
   configman -s com.pipeline.web.SafeUTFURL.url.0=http://
   sample.sungardhe.edu
   configman -s com.pipeline.web.SafeUTFURL.url.1=http
   %3A%2F%2Fsample.sungardhe.edu
   ```

2. Set up a Banner Workflow tab as follows:

   2.1. Log on to Luminis Platform.

   2.2. Select **Content/Layout**.

   2.3. Click **Add New Tab**.

   2.4. Specify the name to be displayed on the tab (for example, *Workflow*).

   2.5. Select the Framed tab type and supply the URL used to log in to your Banner Workflow instance with the relative path after the context root. For example:

   ```
   http://workflow.school.edu/workflow/home/
   worklist.do?renderer=luminis&hidecrumbs=false&
   hidenav=false
   ```

   2.6. Select the desired position of the tab.

   2.7. Click **Submit**

Step 6 Modify Banner Forms technology type

Use the following steps to specify the URL for accessing Internet-native Banner when SSO is implemented.

1. Log in to Banner Workflow as an administrative user.

2. Select Administration > Workflow System Administration. The Workflow System Administration page is displayed.

3. Click **Technology Types**. The Technology Types page is displayed.

4. Click **Banner Forms**. Details for the Banner Forms technology type are displayed.

5. Click the values displayed in **Web Launch Parameters**.

6. Change the value of **banner_inb** to the following:

   ```
   http://<host>:<port>/ssomanager/c/INB
   ```

   This is the URL used to access Internet-native Banner when SSO is implemented.

7. Click **Save**.

SSO under a third-party access manager

A third-party access manager, such as Oracle Access Manager, can be used to provide a SSO solution. In this SSO implementation, the IDM Gateway secures access to Banner Workflow by acting as an access control Gateway to the Banner Workflow server. The Banner Identity Gateway passes the external ID (UDCIdentifier) in the HTTP header or cookie. The IDM Gateway uses the external ID in the header or cookie to identify which Banner Workflow user is requesting access. Any active user that has an external ID can be granted access to Banner Workflow.

Note

The UDCIdentifier of the Workflow user must be inserted into an HTTP header or cookie. ■

The following steps are used to configure Banner Workflow under a third-party access manager:

- Step 1, “Configure IDM Gateway”
- Step 2, “Configure Banner Workflow for IDM Gateway”
- Step 3, “Configure <LogoffUrl> element (optional)”
- Step 4, “Configure Luminis Platform for Banner Workflow”
- Step 5, “Modify Banner Forms technology type”

Step 1 Configure IDM Gateway

Configure the IDM Gateway for Banner Workflow to protect the following URL prefix under the Banner Workflow application context:

```
<webapp>/ssoGateway
```

In the preceding path, `webapp` is the name of the Banner Workflow Web application as deployed on the server.

Step 2 Configure Banner Workflow for IDM Gateway

Use the following steps to configure Banner Workflow for the IDM Gateway.

1. Modify the Banner Workflow `configuration.xml` file as follows:

```
Authentication mode - "IdmGateway"
```

2. Modify the following properties in the IdmGateway section of the same `configuration.xml` file:

Property	Description
`Source`	*Header* or *Cookie*
`HttpVariableName`	Name of the HTTP header or cookie variable (depending on the Source attribute) that the IDM Gateway populates with the current user's UDCIdentifier

Step 3 Configure <LogoffUrl> element (optional)

When running Banner Workflow in IDM Gateway mode, the Logoff link does not log a user out of Banner Workflow because the user is still logged in to the IDM Gateway. If you wish to provide a link to an external URL to log users out of the Gateway, set the optional LogoffUrl element as the last child element of SecurityIntegration.

Step 4 Configure Luminis Platform for Banner Workflow

Use the following steps to configure Luminis Platform for Banner Workflow.

1. If SafeUTFURL is enabled to prevent XSS attacks, either add Banner Workflow to the allowed list or disable SafeUTFURL.

   Example

   If Banner Workflow is deployed to a host named `sample.sungardhe.edu` using http, then it might be necessary to manually add `sample.sungardhe.edu` to the allowed list in both normal and in encoded form:

   ```
   configman -s com.pipeline.web.SafeUTFURL.url.0=http://
   sample.sungardhe.edu
   configman -s com.pipeline.web.SafeUTFURL.url.1=http
   %3A%2F%2Fsample.sungardhe.edu
   ```

2. Set up a Banner Workflow tab as follows:

   2.1. Log on to Luminis Platform.

   2.2. Select **Content/Layout**.

   2.3. Click **Add New Tab**.

   2.4. Specify the name to be displayed on the tab (for example, *Workflow*).

2.5. Select the Framed tab type and supply the URL used to log in to your Banner Workflow instance with the relative path after the context root. For example:

```
http://workflow.school.edu/workflow/home/
worklist.do?renderer=luminis&hidecrumbs=false&
hidenav=false
```

2.6. Select the desired position of the tab.

2.7. Click **Submit**

Step 5 Modify Banner Forms technology type

Use the following steps to specify the URL for accessing Internet-native Banner when SSO is implemented.

1. Log in to Banner Workflow as an administrative user.
2. Select Administration > Workflow System Administration. The Workflow System Administration page is displayed.
3. Click **Technology Types**. The Technology Types page is displayed.
4. Click **Banner Forms**. Details for the Banner Forms technology type are displayed.
5. Click the values displayed in **Web Launch Parameters**.
6. Change the value of **banner_inb** to the following:

   ```
   http://<host>:<port>/ssomanager/c/auth/INB
   ```

   This is the URL used to access Internet-native Banner when SSO is implemented.
7. Click **Save**.

15 Banner Document Management Suite Configuration

Banner® Document Management Suite (BDMS) can be integrated with Banner Enterprise Identity Services (BEIS) beginning with BDMS 8.2. This integration provides the following functionality:

- Provisioning of account information from an SPML Request Authority (RA) to BDMS
- Single sign on (SSO) among applications protected by a central access manager

Account provisioning integration and SSO integration are installed separately. You can install either or both, based on your institution's needs.

Prerequisites

Integration between BEIS and BDMS requires the following software:

- BDMS 8.2 or higher using ApplicationXtender 5.40.322 (SP1) or higher. The `BEIS_AX_<n.nn.nnn>.zip` file, which is required for this integration, is delivered with BDMS and can be found on the SunGard® Higher Education Customer Support Center - Software Downloads under the Banner-Document Management Suite Integration Component.
- BEIS 8.1 or higher
- Banner General 8.3.1 or higher

Note

BDMS supports both the CM security model and the Directory Service (LDAP) security model of ApplicationXtender. The CM security model, however, is highly recommended if you plan to integrate BDMS with BEIS.

Account provisioning

A central identity vault stores the enterprise definition of identity for user accounts. When standard business processes add, change, or delete information in this identity vault,

account information in enterprise applications such as BDMS is automatically added, changed, or deleted. This processing is called account provisioning.

Information flow

As a result of identity events, an SPML Request Authority (RA) dispatches requests to add, update, delete, or look up user accounts in appropriate systems. The RA uses standard Service Provisioning Markup Language (SPML) messages, encapsulating identity information in the UDCIdentity XML structure, to make these requests. The BDMS User Provisioning Web service receives the SPML messages and determines how it should respond to the requests. When properly configured, a message results in a corresponding user account being created, updated, or deleted in BDMS.

All provisioned accounts in BDMS are distinguished by a UDCIdentifier. This value is mapped against the AppXtender user name and maintained in the BDMS User Mapping table (EOBUMAP).

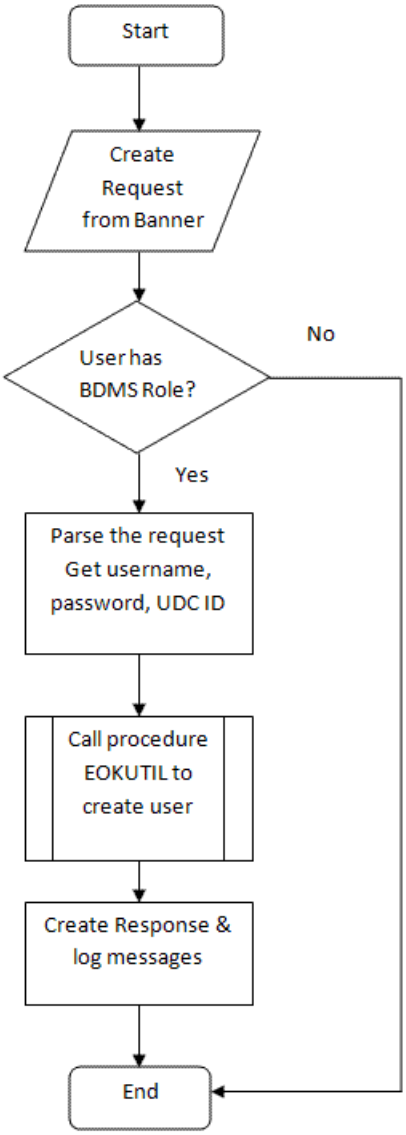
The following sections describe the processing for add, update, and delete requests in more detail.

Add request

If an SPML add request is received, the BDMS User Provisioning Web service parses the incoming SPML message and uses the extension attributes to create the new user. This does not assume that the identity in the add request is eligible to be a BDMS user. If the incoming add request has a BDMS role as a part of the Institution role, then the user is created in BDMS.

Note

New users are granted the same permission as the template user. The user name of this template user is defined in the configuration file for the BDMS User Provisioning Web service. ■

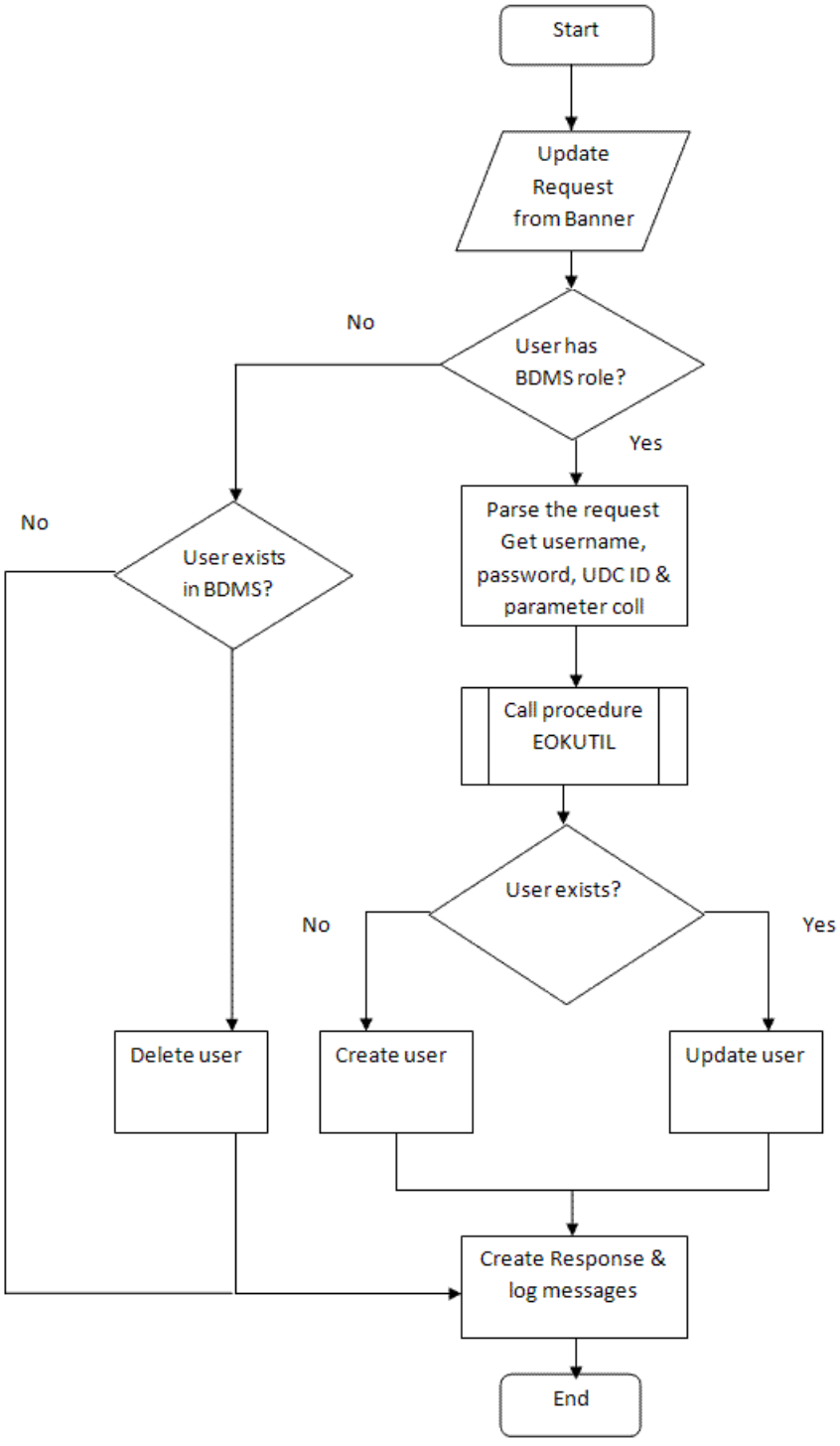


Update request

When an SPML update request is received, the BDMS User Provisioning Web service checks to see if the user specified in the request contains the BDMS role:

- If the user has a BDMS role and the user exists in BDMS, the update request is treated as a normal update and the user is updated.
- If the user has a BDMS role but the user does not exist in BDMS, the user is added into BDMS.
- If the user does not have a BDMS role but the user exists in BDMS, the user is permanently deleted from BDMS.

The BDMS User Provisioning Web service treats an update as a complete replacement of the data. Any manual changes made to the user account outside of BEIS are lost the next time an update message arrives.

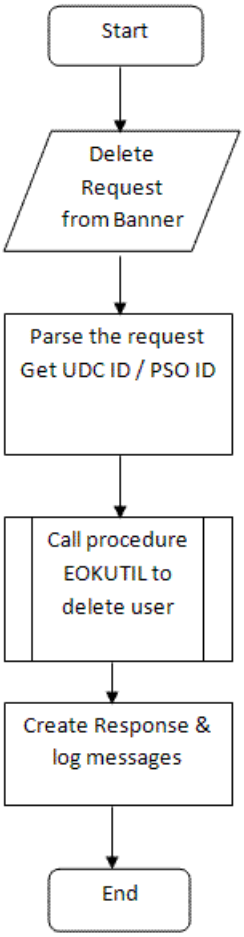


Delete request

If an SPML delete request is received, the BDMS User Provisioning Web service extracts the UDCIdentifier / PSO ID from the SPML message and performs a permanent delete from BDMS and the BDMS User Mapping table. This delete is based on the UDCIdentifier, not the username.

Note

If the user associated with the passed UDCIdentifier does not exist in BDMS but the UDCIdentifier is valid, the UDCIdentifier is also permanently deleted from the BDMS User Mapping table.



Installation of BDMS User Provisioning Web service

Use the following steps to install the BDMS User Provisioning Web service on the ApplicationXtender Web Access server.

1. Extract `BEIS_AX_<n.nn.nnn>.zip` to a temporary directory on the ApplicationXtender Web Access server.

   The value <n.nn.nnn> represents the ApplicationXtender version. For example, if you are using ApplicationXtender 6.50.124, use *BEIS_AX_6.50.124.zip*.

2. Go to the UserProvisioning directory in the extracted files.

3. Determine whether AppXtender Web Access is installed in the default location (`c:\inetpub\wwwroot\AppXtender`):

   3.1. If it is installed in the default location, go to step 4.

   3.2. If it is not installed in the default location, modify `install.bat` and set variable INSTALLFOLDER to point to the AppXtender Web Access installation folder.

4. Run `install.bat`.

5. Open `bdms.userprovisioning.config` with a text editor.

   This configuration file is located in the directory where AppXtender Web Access is installed, typically at `c:\inetpub\wwwroot\AppXtender.`

6. Modify `bdms.userprovisioning.config` as follows:

   6.1. Locate the tag `DataSource` and enter an AppXtender Web Access data source name.

   6.2. Locate the tag `TemplateUserName` and enter the username that is already defined within AppXtender Generator.

   `TemplateUserName` can be any user that is defined in AppXtender Generator, other than SYSOP.

   6.3. Locate the tag `attributeUser` and enter the name that would be present in an SPML message as a part of Extension Attribute that would represent username.

   6.4. (Optional) Add any tags within `<SPMLNameAttributes>` `</SPMLNameAttributes>` that should appear in the procedure for the logging or customizing user creation logic using the new attributes.

Example

```
<SPMLNameAttributes>
       <attributeGName>GivenName</attributeGName>
</SPMLNameAttributes >
```

The value for the `GivenName` tag in the SPML message (for example, XYZ) is part of the paraColl in procedure as `GivenName=XYZ` separated by `";"`.

7. Restart IIS (Internet Information Service).

Configuration of Enterprise Identity Proxy Services

Once the BDMS User Provisioning Web service is deployed on IIS, it must be registered as a Provisioning Service Provider (PSP) in the Enterprise Identity Proxy Services so that identity messages can be published to it. Use the following steps to configure the Identity Proxy.

1. Use the following URL to access the Identity Proxy:

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

2. Log in with the user name and password for the `idpadmin` role.

3. Select PSP Configuration > Add/Update PSP from the menu bar.

4. Configure the following properties:

PSP Location	Location and name where the BDMS User Provisioning Web service is deployed. ***Example*** `http://<server>:<host>/AppXtender/UserProvisioning.asmx`
User Name	Empty
Password	Empty
XPath	Empty
Enabled	*True*, to enable messaging to the BDMS User Provisioning Web service
Add Allowed	*False*, to disable Create message

Update Allowed *True*, to enable Update message

Delete Allowed *True*, to enable Delete message

Time Out *120*

5. Click **Add/Update**.

Configuration of the BDMS role

Integration with BEIS allows BDMS to set up new Banner users in ApplicationXtender when user provisioning is installed. The BDMS role is used for this purpose. To create a user in BDMS, the BDMS role must be assigned to that user.

Update and delete functionality is also provided. If the BDMS role is revoked from a user, that user is removed from BDMS.

The following steps are used to configure the BDMS role:

- Step 1, “Create a new Additional ID Type”
- Step 2, “Create the BDMS role”
- Step 3, “Define a selection rule”

After the BDMS role is configured with these steps, the BDMS role can be assigned to individual users. Refer to “BDMS Role Setup for Banner Enterprise Identity Services Integration” in Chapter 12 of the *Banner Document Management Suite Administration Guide* for instructions on setting up a role for an individual BDMS user.

Step 1 Create a new Additional ID Type

Use the following steps to create a new Additional ID Type designated for BDMS.

1. Access the Additional Identification Type Validation Form (GTVADID).
2. Enter the following:

   Additional ID Type *BDMS*
   Description *BDMS Identity*

3. Save the record.

Step 2 Create the BDMS role

Use the following steps to create the BDMS role.

1. Access the Business Rule Code Validation Form (GTVSQRU).
2. Enter the following:

Code	*BDMS*
Description	*BDMS Role*

3. Save the record.

Step 3 Define a selection rule

Use the following steps to create a rule that selects from Banner the population of persons who should be given the BDMS role.

1. Access the Business Rules Form (GORRSQL).
2. Enter the following in the key block:

Process	*INTCOMP*
Rule	*BDMS*

3. Enter an SQL SELECT statement in the Rule Data block. This statement selects the population of persons from Banner who should be given the BDMS role.

```
SELECT GORADID_PIDM FROM GORADID WHERE GORADID_ADID_CODE =
'BDMS';
```

Note

The SELECT clause must select a PIDM, and nothing else. The selected PIDM must be unique. A SELECT statement that selects the same PIDM more than once creates an error when the role is assigned to the user. ■

4. Select the **System Required** check box.
5. Click **Validate** to validate the SQL statement.
6. Select the **Active** check box to activate the rule.
7. Save the record.

Note

The BDMS rule should always be active and should never be deactivated. Once the rule is activated and validated, role determination automatically uses the rule when it determines a person's roles. ■

Single sign on (SSO) authentication

BDMS supports two types of central access managers for SSO:

- Central Authentication Service (CAS)
- Third-party access manager such as Oracle® Access Manager (OAM)

When SSO is enabled, password synchronization from Internet Native Banner to AppXtender Web Access is disabled automatically.

Note

Password synchronization still occurs from Internet Native Banner to AX Document Manager if the user preference setting is enabled to use AX Document Manager instead of AX Web Access.

Installation of SSO files

Use the following steps to install the BDMS SSO files on the ApplicationXtender Web Access server.

1. Extract `BEIS_AX_<n.nn.nnn>.zip` to a temporary directory on the ApplicationXtender Web Access server.

   The value `<n.nn.nnn>` represents the ApplicationXtender version. For example, if you are using ApplicationXtender 6.50.124, use *BEIS_AX_6.50.124.zip*.

2. Determine whether AppXtender Web Access is installed in the default location (`c:\inetpub\wwwroot\AppXtender`):

   2.1. If it is installed in the default location, go to step 4.

   2.2. If it is not installed in the default location, modify `install.bat` and set variable INSTALLFOLDER to point to the AppXtender Web Access installation folder.

3. Go to the SSO directory in the extracted files.

4. Run `install.bat`.

Optionally, you can run `install.bat` from a command prompt to verify that it ran successfully:

1. Open a command prompt.

2. Change directory to the `drive:path` of the SSO directory of the extracted files.

3. Execute the following command:

   ```
   start install.bat
   ```

   A message is displayed as each command in the `install.bat` file is completed.

Configuration of SSO under CAS

The following steps are used to configure SSO under CAS:

- Step 1, “Modify login.aspx”
- Step 2, “Modify bdms.sso.config”
- Step 4, “Configure Banner”

Step 1 Modify login.aspx

Use the following steps to modify the `Login.aspx` file to call BDMS SSO logic. This file is located on the ApplicationXtender Web Access server, typically at `C:\Inetpub\wwwroot\AppXtender`.

Note

If you want to uninstall SSO support at a later time, you must manually remove the code that you enter in `Login.aspx` or restore the backup version of `Login.aspx`.

1. Back up the `Login.aspx` file.
2. Open `Login.aspx` using a text editor.
3. Copy and paste the following code at the end of the file, between the `</form>` tag and the `</html>` tag:

```
<script runat="server">
   protected override void Page_Load(object sender,
EventArgs e)
   {
      new bdms.idm.sso.LoginHelper().PageLoadHelper();
      base.Page_Load(sender,e);
   }
</script>
```

Step 2 Modify bdms.sso.config

The `bdms.sso.config` file is located in the folder where AppXtender Web Access is installed on the ApplicationXtender Web Access server, typically at `C:\Inetpub\wwwroot\AppXtender`.

For a CAS single sign on implementation, `CASLoginUrl` is the only entry you need to modify in `bdms.sso.config`. Enter the login URL of the CAS server.

Example

```
<CASLoginUrl>
   https://<host>:<port>/cas-3.3.1/login
</CASLoginUrl>
```

Step 3 Add AX Web Access as a protected service on the CAS server

Use the following values when adding AX Web Access as a protected service on the CAS server:

Name	*BDMS AX Web Access*
Service URL	*https://<cas-server>:<port>/AppXtender/***
Description	Your own description
Theme Name	*BDMS*
Status	Select *Enabled* and *SSO Participant*
Attributes	Select *UDC_IDENTIFIER*

Step 4 Configure Banner

Use the following steps to set up SSO from Banner to AppXtender Web Access.

1. Access the BDMS System Settings Form (EXAINST).
2. Change the value of **WebXtender Root** to `http://<host>/appxtender/ISbmitQuery.aspx?sso=true`.
3. Save the record.

Once SSO is configured, the following URL can be used to log in to AppXtender Web Access through SSO:

```
http://<host>/AppXtender/Login.aspx?sso=true
```

Configuration of SSO under a third-party access manager

The following steps are used to configure SSO under a third-party access manager:

- Step 1, “Modify login.aspx”
- Step 2, “Modify bdms.sso.config”

- Step 3, “Configure BDMS on the third-party access manager server”
- Step 4, “Configure Banner”

Step 1 Modify login.aspx

Use the following steps to modify the `Login.aspx` file to call BDMS SSO logic. This file is located on the ApplicationXtender Web Access server, typically at `C:\Inetpub\wwwroot\AppXtender`.

Note

If you want to uninstall SSO support at a later time, you must manually remove the code that you enter in `Login.aspx` or restore the backup version of `Login.aspx`.

1. Back up the `Login.aspx` file.
2. Open `Login.aspx` using a text editor.
3. Copy and paste the following code at the end of the file, between the `</form>` tag and the `</html>` tag:

```
<script runat="server">
   protected override void Page_Load(object sender,
EventArgs e)
   {
      new bdms.idm.sso.LoginHelper().PageLoadHelper();
      base.Page_Load(sender,e);
   }
</script>
```

Step 2 Modify bdms.sso.config

The `bdms.sso.config` file is located in the folder where AppXtender Web Access is installed on the ApplicationXtender Web Access server, typically at `C:\Inetpub\wwwroot\AppXtender`. Use the following steps to modify `bdms.sso.config`.

1. Depending on whether your access manager is configured to use a HTTP header parameter or a cookie to pass the UDCIdentifier, enter a value for either `UDCIDHeaderVariableName` or `UDCIDCookieName`. Only one of them is required.

   Example

   ```
   <UDCIDHeaderVariableName>
   HTTP_UDC_ID
   </UDCIDHeaderVariableName>
   ```

2. Enter the URL of the Ticketing Web service.

Example

```
<TicketingWebServiceUrl>
https://<host>:<port>/ssomanager/ws/sso-ticket-service
</TicketingWebServiceUrl>
```

3. Enter the URL of the Credential Web service.

Example

```
<CredentialWebServiceUrl>
https://<host>:<port>/ssomanager/ws/credential-service
</CredentialWebServiceUrl>
```

4. Enter the Web service username and password. The Ticketing Web service and Credential Web service are protected. Username and password are required. Leave the domain field empty.

Example

```
<WebServiceUsername> username </WebServiceUsername>
<WebServicePassword> password </WebServicePassword>
<WebServiceDomain> </WebServiceDomain>
```

Step 3 Configure BDMS on the third-party access manager server

The following steps are used if you are configuring Oracle Access Manager. If you are configuring another access manager, the steps will be different.

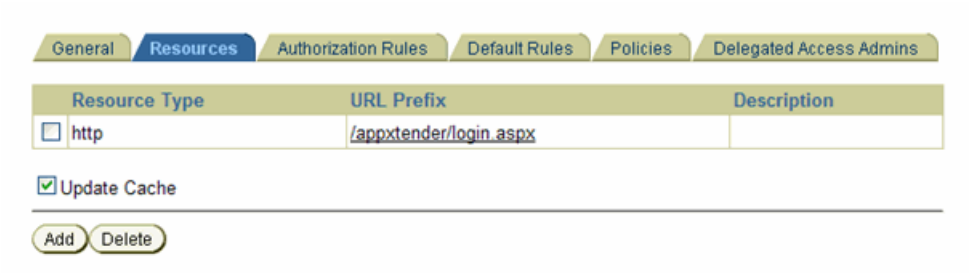
Refer to the OAM installation and configuration documents for information on configuring the OAM server. Use the following steps to enter specific configuration information for BDMS.

1. From Access System Configuration, add a new Access Gate to protect AppXtender Web Access.

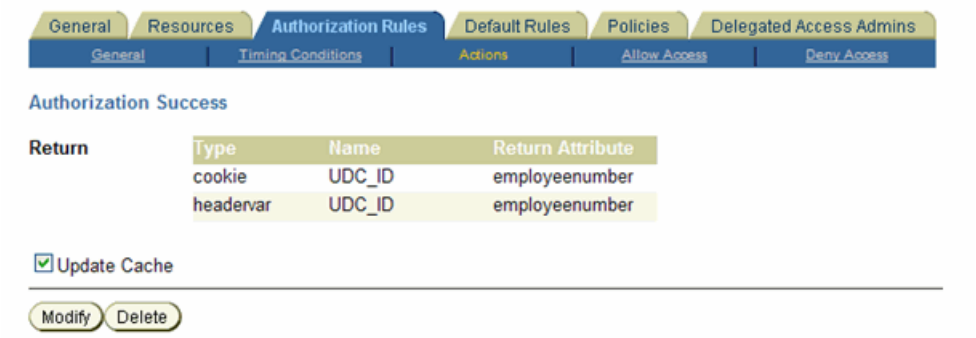
Hostname	Server name of AppXtender Web Access
Port	Port number used by Access Gate, typically *6023*
Deny on Not Protected	*Off*

2. From Policy Manager, create a new Policy Domain by adding a new resource on the Resources tab.

   Resource Type *http*
   URL Prefix */appxtender/login.aspx*



3. On the Authorization Rules tab, add a new authorization rule to configure a header variable or cookie to pass the UDCIdentifier (UDC_ID).



Step 4 Configure Banner

Use the following steps to set up SSO from Banner to AppXtender Web Access.

1. Access the BDMS System Settings Form (EXAINST).

2. Change the value of **WebXtender Root** to `http://<host>/appxtender/ISbmitQuery.aspx?sso=true`.

3. Save the record.

Once SSO is configured, the following URL can be used to log in to AppXtender Web Access through SSO:

```
http://<host>/AppXtender/Login.aspx?sso=true
```

A Oracle Streams

When a Banner® identity attribute changes, the Oracle Streams technology captures, propagates, and applies the change for use by the Banner Identity Gateway. This appendix provides information about administering Oracle Streams for use with Banner Enterprise Identity Services (BEIS).

Stop Oracle Streams processes

1. Open an SQL*Plus session and log in with the Oracle Streams administrator account.
2. Execute the following in the order given:

```
sqlplus> set serveroutput on size 1000000
sqlplus> exec gp_streams_util.p_stop_capture('IAM');
sqlplus> exec gp_streams_util.p_stop_apply('IAM');
```

Note

You can also use the Banner Identity Gateway administrative interface to stop the capture and apply processes. See "Stop Oracle Streams IAM processes" on page 7-83.

Start Oracle Streams processes

1. Open an SQL*Plus session and log in with the Oracle Streams administrator account.
2. Execute the following in the order given:

```
sqlplus> set serveroutput on size 1000000
sqlplus> exec gp_streams_util.p_start_capture('IAM');
sqlplus> exec gp_streams_util.p_start_apply('IAM');
```

Note

You can also use the Banner Identity Gateway administrative interface to start the capture and apply processes. See "Start Oracle Streams IAM processes" on page 7-82.

Create attribute selection rule

Use the Business Rules Form (GORRSQL) to define additional identity attributes in the UDCIdentity XML structure. Refer to the definition of an attribute selection rule on page 5-18 for more information.

Modify capture definitions

1. Use the Streams Rules Configuration Form (GUASADM) to modify definitions for a capture process.

   You can modify the tables that are monitored, the columns that are monitored, and the rules that are evaluated to determine whether a transaction is captured.

2. Open an SQL*Plus session and log in with the Oracle Streams administrator account.

3. Execute the following in the order given:

```
sqlplus> set serveroutput on size 1000000
sqlplus> exec gp_streams_util.p_configure_rules('IAM');
```

   This step stops the capture and apply processes (if running), reconfigures the Oracle Streams environment with the new rules, and re-starts the processes.

Note

You can also use the Banner Identity Gateway administrative interface to reconfigure the Oracle Streams environment with the new rules. See "Load updated capture rules" on page 7-83.

B UDCIdentity Message Sample

This appendix contains a UDCIdentity message sample. Refer to Chapter 2, "UDCIdentity" for more information about the UDCIdentity overview, details, and structure.

Sample message

```
<?xml version="1.0" encoding="UTF-8"?>
<UDCIdentity
xmlns="urn:sungardhe:enterprise:domain:identity:1.0"
xmlns:xdb="http://xmlns.oracle.com/xdb"
xmlns:rco="urn:sungardhe:enterprise:resources:common:1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:sungardhe:enterprise:domain:
identity:1.0
identity_domain.xsd" PUBLISHER_NAME="BANNER" action="ADD">
   <UDCIdentifier>3B99F5545DEE329BE0440003BA33B440
</UDCIdentifier>
   <PersonIdentity>
      <PersonName>
         <FormattedName>James Martin Larsson</FormattedName>
         <GivenName>James</GivenName>
         <PreferredGivenName>Jim</PreferredGivenName>
         <MiddleName>Martin</MiddleName>
         <FamilyName>Larsson</FamilyName>
         <Affix type="formOfAddress">Mr.</Affix>
         <Affix type="qualification">Sr.</Affix>
      </PersonName>
   </PersonIdentity>
   <EmailAddress>jhdegasse@metu.edu</EmailAddress>
   <PrimaryAddress>
      <PostalCode>59640</PostalCode>
      <Region>MT</Region>
      <Municipality>Carbondale</Municipality>
      <AddressLine>22 Underwood Lane</AddressLine>
   </PrimaryAddress>
```

```
      <CampusAddress>
         <PostalCode>59620</PostalCode>
         <Region>MT</Region>
         <Municipality>Helena</Municipality>
         <AddressLine>PO Box 200113</AddressLine>
      </CampusAddress>
      <CampusPhone>
         <AreaCityCode>620</AreaCityCode>
         <SubscriberNumber>7609500</SubscriberNumber>
         <Extension>113</Extension>
      </CampusPhone>
      <MobilePhone>
         <AreaCityCode>620</AreaCityCode>
         <SubscriberNumber>3332384</SubscriberNumber>
      </MobilePhone>
      <Fax>
         <AreaCityCode>620</AreaCityCode>
         <SubscriberNumber>7609551</SubscriberNumber>
      </Fax>
      <InstitutionRoles>
         <institutionrole>
            <role>BANNERINB</role>
         </institutionrole>
      </InstitutionRoles>
   </UDCIdentity>
```

C EduPerson Data Mapping

EduPerson is an LDAP object class authored and promoted by the EDUCAUSE/Internet2 eduPerson Task Force to support inter-institutional exchange of identity data. Enterprise directories are the operational foundation of most middleware services. They can contain critical information for people, processes, resources, and groups. Having this information in a central, common storage area allows diverse applications from diverse locations to access consistent and comprehensive data about these objects.

Within the institution, data is exchanged between Banner® and the enterprise identity management system (EIMS) using the UDCIdentity XML structure, which contains data that can be mapped to eduPerson attributes if implemented in the EIMS identity store.

When implementing Banner and other SunGard® Higher Education enterprise applications in conjunction with an EIMS, an institution that requires support for eduPerson must ensure that the identity store component of the EIMS supports the eduPerson specification, and that the data provided by the UDCIdentity XML structure is appropriately mapped to meet institutional policies.

LDAP

Lightweight Directory Access Protocol (LDAP) is a method created by the international technical community as a standard way to exchange information stored in LDAP directories. LDAP directories adhere to the following X.500 model:

- A directory is a tree of directory entries.
- An entry consists of a set of attributes.
- An attribute has a name (an attribute type or attribute description) and one or more values.
- The attributes are defined in a schema.

The schema for an LDAP directory defines object classes, which themselves define what kind of object a directory entry can represent. For example, an object can be a person, organization, or domain. It can also represent the attributes that are required or optional for entries associated with that class.

objectClass attribute

Each entry in the directory must have an objectClass attribute containing named classes defined in the schema. Typically, the objectClass attribute is multi-valued and contains the class “top” as well as some number of other classes. For example, an entry representing a person might belong to the classes “top” and “person.” Membership in the “person” class would require the entry to contain the “sn” and “cn” attributes and allow the entry also to contain “userPassword,” “telephoneNumber,” and other attributes.

EduPerson higher education attributes

Despite the standardization, directories at higher education institutions differ because there are no established patterns for building general-purpose institutional directories. This makes it hard for a person at one institution to get meaningful information, such as email addresses, from another institution. To address this disparity, the eduPerson object class provides a universal set of attributes about persons involved in higher education. Institutions can store the attributes to facilitate communication with other institutions.

The attributes defined by eduPerson are *specific* to higher education. They do not include common person attributes such as first name, last name, or email. For this reason, the eduPerson specification recommends that person entries in the directory have the person, organizationalPerson, and inetOrgPerson object classes defined, because they provide standard definitions for common attributes.

The eduPerson attributes are defined briefly here. More extensive documentation can be found at http://middleware.internet2.edu/eduperson/.

All attribute names are prefaced with eduPerson. All are defined as optional.

Attribute Name	Description
`eduPersonNickname`	Person's nickname or informal name
`eduPersonPrincipalName`	“NetID” of the person for inter-institutional authentication. It should be represented in the form “user@scope,” where scope defines a local security domain.
`eduPersonAffiliation`	Person's relationship to the institution in broad categories such as student, faculty, staff, alum
`eduPersonPrimaryAffiliation`	Person's primary relationship to the institution in broad categories such as student, faculty, staff, alum

Attribute Name	Description
eduPersonScopedAffiliation	Person's affiliation within a particular security domain. This is a multi-valued attribute whose values consist of a left (affiliation) and right (domain) component separated by an “@” sign. The left component is one of the values from the eduPersonAffiliation-controlled vocabulary. The right component matches the “scope” value for eduPersonPrincipalName because both identify a security domain.
eduPersonOrgDN	Distinguished name (DN) of the directory entry representing the institution with which the person is associated
eduPersonOrgUnitDN	Distinguished name(s) (DN) of the directory entries representing the person’s organizational unit(s). Can be multi-valued.
eduPersonPrimaryOrgUnitDN	Distinguished name (DN) of the directory entry representing the person’s primary organizational unit
eduPersonEntitlement	URI (either URN or URL) that indicates a set of rights to specific resources
eduPersonTargetedID	Persistent, non-reassigned, privacy-preserving identifier for a principal shared between an identity provider and service provider (or a group of service providers)

EduPerson and BEIS

Banner Enterprise Identity Services (BEIS) supports centralized identity vaults such as those available from Sun, Novell, and others. The SunGard Higher Education strategy allows for flexibility in regard to the chosen system for recording person data. Clients can choose whether Banner provisions identity information or whether this is the responsibility of another system.

If Banner is the authoritative source, the creation of a new person record in Banner triggers the creation of the person in the identity store (that is, the enterprise directory). The EIMS subsequently provisions user accounts across the installed applications. Refer to Chapter 1, “Overview” for example configurations for BEIS.

Banner identity components

Banner enterprise components provide the data required to populate the identity store, or at least provide the data for which Banner is authoritative. There are no component restrictions or requirements for the content or structure of the identity store. It can be an LDAP directory or not. Its schema can include the eduPerson object class or not. The structure, content, and architecture of the identity store are the responsibility of the EIMS.

Banner and its enterprise components provide the appropriate data for populating the identity store in a defined format. The UDCIdentity XML structure is used to exchange person data between Banner and the EIMS as well as between Banner and other Banner enterprise components.

UDCIdentity XML structure standards

The UDCIdentity XML structure and the formatting of the data within it contain several standards. In addition to using eduPerson and its related LDAP object classes as a starting point for determining the data that should be contained in the UDCIdentity XML structure, the internationally-oriented person name and address structures defined by HR-XML are the basis for the UDCIdentity XML structure. As a result, while the Banner enterprise components and the identity store are independent of one another, their respective attributes can be mapped easily. The following table provides one possible mapping to support the eduPerson specification.

eduPerson attribute	Data Source	UDCIdentity Element
`eduPersonNickname`	Banner - `SPBPERS_PREF_FIRST_NAME`	`rco:PreferredGivenName`
`eduPersonPrincipalName`	EIMS	`did:LogonID`
`eduPersonAffiliation`	Banner - `GB_INSTITUTION_ROLE`	`did:institutionrole/did:role`
`eduPersonPrimaryAffiliation`	Banner does not support the concept of a primary role.	Can be supported in the extension element.
`eduPersonScopedAffiliation`	Banner - `GB_INSTITUTION_ROLE`	`did:institutionrole/did:role` + `did:institutionrole/did:institution`
`eduPersonOrgDN`	EIMS	Derived by EIMS based on `did:institutionrole/did:institution`

eduPerson attribute	Data Source	UDCIdentity Element
eduPersonOrgUnitDN	Department or organization information from Banner could be used; however, SunGard Higher Education has determined that these fields are not unique enough to be used in a base system implementation. The base system implementation can be modified declaratively to publish it.	Can be supported in the extension element.
eduPersonPrimaryOrg UnitDN	Banner can provide the primary organization unit, based on institutional policy decisions on data.	Can be supported in the extension element.
eduPersonEntitlement	EIMS	Can be supported in the extension element.
eduPersonTargetedID	SunGard Higher Education GUID Service	UDCIdentifier

D CAS Installation and Configuration

This appendix provides information about installing and configuring the Central Authentication Service (CAS) server to support Banner® Enterprise Identity Services (BEIS). CAS versions 3.2.1.1, 3.3.1, and 3.4.2.1 are supported.

This appendix also includes information about configuring the following SunGard® Higher Education applications to participate as clients in a CAS single sign on (SSO) session:

- Self-Service Banner (SSB)
- Internet-native Banner (INB)
- Luminis® Platform 4.x
- Banner Workflow
- Banner Document Management Suite

Note

If you are using the embedded CAS server in Luminis® Platform 5.x, refer to the Luminis Platform Integration Setup Guide for installation and configuration information.

Introduction to CAS

CAS is an enterprise single sign on (SSO) solution for Web applications. CAS SSO improves the user experience when running several Web applications, each with its own means of authentication. With an SSO solution, different Web applications can authenticate to one authoritative source of trust, instead of requiring the end user to log in to each separate service.

For example, if you want to access your portal instance, you are redirected to CAS to log in. CAS detects your secure cookie, does the single sign on, and redirects you to the portal. The portal validates the ticket and logs you in to the portal. You do not have to give your username and password to your portal. With proxy authentication through CAS, you do not have to sign in and sign off between applications in your online session.

CAS allows you to implement an SSO solution in a matter of hours. Many client programs have been developed to customize applications so they can use CAS for user authentication.

Prerequisites

Before implementing CAS with BEIS, the following prerequisites must be met:

- The identity repository that is used by the CAS server must be UDCIdentifier aware.
- You must have a knowledge of the CAS protocol (2.0) and architecture (3.2). Refer to http://www.ja-sig.org/products/cas/overview/index.html for detailed information.

Configuration of the CAS server

Note

If you are using the embedded CAS server in Luminis Platform 5.x, refer to the Luminis Platform Integration Setup Guide for installation and configuration information.

Your CAS server must be configured to retrieve `UDC_IDENTIFIER` data from your LDAP directory. Use the following steps to specify LDAP connection information and identify the LDAP attribute name where the `UDC_IDENTIFIER` value is stored.

- Step 1, “Download JA-SIG CAS server distribution”
- Step 2, “Download CAS extensions jar file”
- Step 3, “Modify WEB-INF/web.xml”
- Step 4, “Modify WEB-INF/cas.properties”
- Step 5, “Modify WEB-INF/spring-configuration/uniqueIdGenerators.xml”
- Step 6, “Modify WEB-INF/spring-configuration/argumentExtractorsConfiguration.xml”
- Step 7, “Modify WEB-INF/cas-servlet.xml”
- Step 8, “Modify WEB-INF/deployerConfigContext.xml”
- Step 9, “Modify WEB-INF/classes/default_views.properties”
- Step 10, “Deploy application”
- Step 11, “Test CAS Web application”
- Step 12, “Modify Tomcat (if needed)”
- Step 13, “Define CAS managed services”

Step 1 Download JA-SIG CAS server distribution

Use the following steps to download the JA-SIG CAS server distribution and build the `SGHE_CAS_HOME` directory.

1. Download the JA-SIG CAS server 3.2.1.1, 3.3.1, or 3.4.2.1 distribution from http://www.ja-sig.org/downloads/cas/.

2. Extract the distribution to a temporary directory:

   ```
   jar -xvf cas-server-<version>-release.zip
   ```

3. Follow the instructions in `INSTALL.txt` to build the `cas.war` file from source code.
   -or-
   Use the prebuilt binary distribution located at `cas-server-<version>/modules/cas-server-webapp-<version>.war`.

4. Copy `cas-server-webapp-<version>.war` to a new directory and unpack the distribution.

   The `cas-server-webapp-<version>` directory is created. This directory is called `SGHE_CAS_HOME` in the following steps.

Example

The following commands copy `cas-server-webapp-<version>.war` to `c:/cas`:

```
copy cas-server-webapp-<version>.war c:\cas
cd cas
jar -xvf cas-server-webapp-<version>.war
```

Step 2 Download CAS extensions jar file

A standard jar file, available from SunGard Higher Education, is used to modify the CAS server. SunGard Higher Education supports CAS versions 3.2.1.1, 3.3.1, and 3.4.2.1. Use the following steps to download the file.

1. Download the CAS extension jar file from the Software Downloads page on the SunGard Higher Education Customer Support Center. Choose the file that supports your version of CAS:

   ```
   SGHE_CAS_3.2.1.1_UTILS.zip
   SGHE_CAS_3.3.1_UTILS.zip
   SGHE_CAS_3.4.2.1_UTILS.zip
   ```

2. Extract the contents of the file to a temporary directory.

3. Copy the jar files from the `WEB-INF/lib` directory to the `SGHE_CAS_HOME/WEB-INF/lib` directory. This step moves the SunGard Higher Education CAS extension jar files to the CAS server distribution.

Note

If you are using the /samlValidate service, skip to Step 8, "Modify WEB-INF/deployerConfigContext.xml".

If you are using the /bannerValidate service, continue with Step 3, "Modify WEB-INF/web.xml". ■

Step 3 Modify WEB-INF/web.xml

Note

Skip this step if you are using the /samlValidate service. ■

Use the following steps to modify `web.xml`.

1. Change to the `cas-server-<version>` directory and navigate to the `SGHE_CAS_HOME/WEB-INF` directory.

2. Open `WEB-INF/web.xml`.

3. Add the following servlet mapping:

```
<servlet-mapping>
  <servlet-name>cas</servlet-name>
  <url-pattern>/bannerValidate</url-pattern>
</servlet-mapping>
```

4. Save and close `web.xml`.

Step 4 Modify WEB-INF/cas.properties

Note

Skip this step if you are using the /samlValidate service. ■

Use the following steps to modify `cas.properties`.

1. Open `WEB-INF/cas.properties`.

2. In the first three property definitions, change `localhost:8443` to the CAS server hostname and ssl port number.

3. Change the property `host.name=cas` to `host.name=<cas server machine name>.`

4. Save and close `cas.properties`.

Step 5 Modify WEB-INF/spring-configuration/uniqueIdGenerators.xml

Note

Skip this step if you are using the /samlValidate service. ■

Use the following steps to modify `uniqueIdGenerators.xml`.

1. Open `WEB-INF/spring-configuration/uniqueIdGenerators.xml`.
2. Add the following to `util:map id="uniqueIdGeneratorsMap"`:

```
<entry key="com.sghe.cas.principal.BannerAccountsService"
  value-ref="serviceTicketUniqueIdGenerator" />
```

3. Save and close `uniqueIdGenerators.xml`.

Step 6 Modify WEB-INF/spring-configuration/ argumentExtractorsConfiguration.xml

Note

Skip this step if you are using the /samlValidate service. ■

Use the following steps to modify `argumentExtractorsConfiguration.xml`.

1. Open `WEB-INF/spring-configuration/ argumentExtractorsConfiguration.xml`.
2. Add the following xml configuration:

```
<bean id="BannerArgumentExtractor"
class="com.sghe.cas.web.support.BannerArgumentExtractor">
</bean>
```

3. Add the following reference to `util:list - argumentExtractors`:

```
<util:list id="argumentExtractors">
  <ref bean="BannerArgumentExtractor" />
  <ref bean="casArgumentExtractor" />
  <ref bean="samlArgumentExtractor" />
</util:list>
```

4. Save and close `argumentExtractorsConfiguration.xml`.

Step 7 Modify WEB-INF/cas-servlet.xml

Note

Skip this step if you are using the /samlValidate service. ■

Use the following steps to modify `cas-servlet.xml`.

1. Open `WEB-INF/cas-servlet.xml`.

2. Add the following xml configuration:

```
<bean id="bannerAccountValidateController"
class="org.jasig.cas.web.ServiceValidateController"
p:validationSpecificationClass="org.jasig.cas.validation.
Cas20WithoutProxyingValidationSpecification"
p:centralAuthenticationService-
ref="centralAuthenticationService"
p:proxyHandler-ref="proxy20Handler"
p:argumentExtractor-ref="BannerArgumentExtractor"
p:successView="bannerAccountServiceSuccessView"
p:failureView="bannerAccountServiceFailureView" />
```

3. Add the following property to the bean `handlerMappingC`:

```
<prop key="/bannerValidate">bannerAccountValidateController
</prop>
```

4. Save and close `cas-servlet.xml`.

Step 8 Modify WEB-INF/deployerConfigContext.xml

Use the following steps to add specific entries for your LDAP configuration to `deployerConfigContext.xml`.

1. Open `WEB-INF/deployerConfigContext.xml`.

2. Remove the following beans inside `list` in the `credentialsToPrincipal Resolver` property:

```
<bean class="org.jasig.cas.authentication.principal.
UsernamePasswordCredentialsToPrincipalResolver" />

<bean class="org.jasig.cas.authentication.principal.
HttpBasedServiceCredentialsToPrincipalResolver" />
```

3. Add the following bean inside `list` in the `credentialsToPrincipalResolver` property:

```
<bean class="org.jasig.cas.authentication.principal.
CredentialsToLDAPAttributePrincipalResolver">
 <property name="credentialsToPrincipalResolver">
  <bean class="org.jasig.cas.authentication.principal.
  UsernamePasswordCredentialsToPrincipalResolver" />
 </property>
 <property name="filter" value="(uid=%u)" />
 <property name="principalAttributeName" value="uid" />
 <property name="searchBase" value="dc=sungardhe,dc=com" />
```

```
  <property name="contextSource" ref="contextSource" />
  <property name="attributeRepository">
    <ref bean="attributeRepository" />
  </property>
</bean>
```

Note

Enter specific information for you configuration. For example, modify `property name ="searchBase"` to specify the correct searchBase entry for your LDAP. Change `dc=sungardhe` to reflect your environment.

4. By default, CAS provides two handlers. Remove the second handler (highlighted below) from the configuration.

```
<bean class="org.jasig.cas.authentication.handler.support.
HttpBasedServiceCredentialsAuthenticationHandler"
p:httpClient-ref="httpClient" />

<bean class="org.jasig.cas.authentication.
handler.support.SimpleTestUsernamePasswordAuthentication
Handler" />
```

5. Add the following bean inside `list` in the `authenticationHandlers` property.

```
<bean class="org.jasig.cas.adaptors.ldap.BindLdapAuthentication
Handler" >
  <property name="filter" value="uid=%u" />
  <property ‘searchBase"
  value="ou=users,dc=sungardhe,dc=com" />
  <property name="contextSource" ref="contextSource" />
</bean>
```

6. Modify bean `userDetailsService` to define an LDAP user to support CAS administration functions. This user must already exist in LDAP. Specify the user ID and password of the CAS administrative user with the default CAS administration role name (`ROLE_ADMIN`).

   6.1. For CAS 3.2.1.1, modify bean `userDetailsService` as follows.

```
<bean id="userDetailsService"
class="org.acegisecurity.userdetails.memory.InMemoryDaoImpl">
<property name="userMap">
   <value>triddle=123456,ROLE_ADMIN</value>
  </property>
</bean>
```

   6.2. For CAS 3.3.1, modify bean `userDetailsService` as follows.

```
<bean id="userDetailsService"
class="org.springframework.security.userdetails.memory.InMemory
DaoImpl">
```

```
  <property name="userMap">
    <value>triddle=123456,ROLE_ADMIN</value>
  </property>
</bean>
```

6.3. For CAS 3.4.2.1, modify bean `userDetailsService` as follows.

```
<sec:user-service id="userDetailsService">
  <sec:user name="triddle" password="123456"
authorities="ROLE_ADMIN" />
</sec:user-service>
```

7. Replace the `attributeRepository` bean as follows. This bean is used to specify the attributes that must be fetched from LDAP.

Note

Modify `property name="baseDN value=”ou=users, dc=sungardhe,dc=com”` to reflect your LDAP configuration. ■

7.1. For CAS 3.2.1.1 and CAS 3.3.1, replace the `attributeRepository` bean with the following bean.

```
<bean id="attributeRepository"
  class="org.jasig.services.persondir.support.ldap.
  LdapPersonAttributeDao">
  <property name="contextSource" ref="contextSource" />
  <property name="baseDN" value="ou=users,dc=sungardhe,
  dc=com" />
  <property name="query" value="(uid={0})" />
  <property name="ldapAttributesToPortalAttributes">
  <map>
    <entry value="UDC_IDENTIFIER" key="cn" />
    <entry value="uid" key="uid"/>
  </map>
  </property>
</bean>
```

7.2. For CAS 3.4.2.1, replace the `attributeRepository` bean with the following bean.

```
<bean id="attributeRepository"
  class="org.jasig.services.persondir.support.ldap.
  LdapPersonAttributeDao">
  <property name="contextSource" ref="contextSource" />
  <property name="baseDN" value="ou=users,dc=sungardhe,
  dc=com" />
  <property name="requireAllQueryAttributes" value="true" />
  <!--
  Attribute mapping between principal (key) and LDAP (value)
  names used to perform the LDAP search. By default, multiple
  search criteria are ANDed together. Set the queryType property
  to change to OR.
```

```
  -->

  <property name="queryAttributeMapping">
    <map>
      <entry key="username" value="uid" />
    </map>
  </property>
  <property name="resultAttributeMapping">
    <map>
      <!-- Mapping between LDAP entry attributes (key) and
      Principal's (value) -->
      <entry key="cn" value="UDC_IDENTIFIER"/>
      <entry key="uid" value="uid" /
    </map>
  </property>
</bean>
```

8. Replace the `contextSource` bean as follows. This bean is used to specify the LDAP properties.

Note

Modify the LDAP properties url, userDn, and password to reflect your LDAP configuration. ■

8.1. For CAS 3.2.1.1 and CAS 3.3.1, replace the `contextSource` bean with the following bean.

```
<bean id="contextSource" class="org.jasig.cas.adaptors.ldap.
util.AuthenticatedLdapContextSource">
  <property name="url" value="ldap://localhost:389" />
  <property name="userDn" value="cn=Manager,dc=sungardhe,
  dc=com"/>
  <property name="password" value="u_pick_it"/>
</bean>
```

8.2. For CAS 3.4.2.1, replace the `contextSource` bean with the following bean.

```
<bean id="contextSource" class="org.springframework.ldap.
core.support.LdapContextSource">
  <property name="pooled" value="false"/>
  <property name="urls">
    <list>
      <value>ldap://localhost:389</value>
    </list>
  </property>
  <property name="userDn" value="cn=Manager,dc=sungardhe,
  dc=com"/>
  <property name="password" value="u_pick_it"/>
  <property name="baseEnvironmentProperties">
    <map>
      <entry>
        <key>
```

```
        <value>java.naming.security.authentication</value>
      </key>
      <value>simple</value>
    </entry>
  </map>
 </property>
</bean>
```

9. Modify bean serviceRegistryDao as follows.

```
<bean id="serviceRegistryDao"
class="org.jasig.cas.services.InMemoryServiceRegistryDaoImpl" >
 <property name="registeredServices">
  <list>
   <bean class="org.jasig.cas.services.
   RegisteredServiceImpl">
    <property name="id" value="2" />
    <property name="name" value="IMAPS" />
    <property name="description" value="Only Allows HTTPS
    Urls" />
    <property name="serviceId" value="imaps://**" />
   </bean>
   <bean class="org.jasig.cas.services.
   RegisteredServiceImpl">
    <property name="id" value="3" />
    <property name="name" value="IMAP" />
    <property name="description" value="Only Allows IMAP
    Urls" />
    <property name="serviceId" value="imap://**" />
   </bean>
   <bean class="org.jasig.cas.services.
   RegisteredServiceImpl">
    <property name="id" value="1" />
    <property name="name" value="HTTPS" />
    <property name="description" value="Only Allows HTTPS
    Urls" />
    <property name="serviceId" value="https://**"
    <property name="allowedAttributes">
     <list>
      <value>UDC_IDENTIFIER</value>
     </list>
    </property>
   </bean>
   <bean class="org.jasig.cas.services.
   RegisteredServiceImpl">
    <property name="id" value="0" />
    <property name="name" value="HTTP" />
    <property name="description" value="Only Allows HTTP
    Urls" />
    <property name="serviceId" value="http://**" />
    <property name="allowedAttributes">
    <list>
```

```
        <value>UDC_IDENTIFIER</value>
      </list>
      </property>
    </bean
  </list>
 </property>
</bean>
```

Note

In the default deployment, volatile data is cleared when the application restarts. If you want to persist the data, use steps 10 through 21 to make the necessary configuration changes.

10. Modify bean `serviceRegistryDao` as follows.

```
<bean id="serviceRegistryDao"
class="org.jasig.cas.services.JpaServiceRegistryDaoImpl"
     p:entityManagerFactory-ref="entityManagerFactory" />
```

11. Add the following bean definitions under the `beans` root element.

```
<bean id="entityManagerFactory" class="org.spring
framework.orm.jpa.LocalContainerEntityManagerFactoryBean">
 <property name="dataSource" ref="dataSource"/>
 <property name="jpaVendorAdapter">
  <bean class="org.springframework.
  orm.jpa.vendor.HibernateJpaVendorAdapter">
   <property name="generateDdl" value="true"/>
   <property name="showSql" value="true" />
  </bean>
 </property>
 <property name="jpaProperties">
  <props>
   <prop key="hibernate.dialect">org.hibernate.dialect.
   OracleDialect</prop>
   <prop key="hibernate.hbm2ddl.auto">update</prop>
  </props>
 </property>
</bean>

<bean id="transactionManager"
class="org.springframework.orm.jpa.JpaTransactionManager">
<property name="entityManagerFactory"
ref="entityManagerFactory"/>
</bean>

<tx:annotation-driven transaction-manager=
"transactionManager"/>

<bean id="dataSource"
 class="org.apache.commons.dbcp.BasicDataSource"
 p:driverClassName="oracle.jdbc.driver.OracleDriver"
 p:url="jdbc:oracle:thin:@localhost:1521:ORCL"
```

```
p:password="tiger"
p:username="scott" />
```

Note

For the bean `id="dataSource"`, modify the URL, password, and username to reflect your environment. The password and username must be a valid Oracle password and username. You can use the `ssomgr` Oracle account that was created for the Banner Identity Gateway to store CAS managed services information. ■

12. Add the `tx:` namespace to the `beans` root element tag:

    12.1. For CAS 3.2.1.1 and CAS 3.3.1, add the namespace as follows:

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:p="http://www.springframework.org/schema/p"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-
2.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.0.xsd">
```

    12.2. For CAS 3.4.2.1, add the namespace as follows:

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:p="http://www.springframework.org/schema/p"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">
```

13. Save and close `deployerConfigContext.xml`.

14. Open `WEB-INF/cas.properties` and make sure the following entry is present and uncommented:

```
database.hibernate.dialect=org.hibernate.dialect.OracleDialect
```

15. Add the following property in `WEB-INF/cas.properties`.

```
ticket.cleaner.database.platform=SQL92
```

16. Save and close `cas.properties`.

17. Open `WEB-INF/spring-configuration/ticketRegistry.xml`.

18. Replace the entire file contents with one of the following.

18.1. For CAS 3.2.1.1 and CAS 3.3.1, replace the entire file contents as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:p="http://www.springframework.org/schema/p"
 xsi:schemaLocation="http://www.springframework.org/schema/
 beans http://www.springframework.org/schema/beans/spring-
 beans-2.0.xsd">
 <description>
  Configuration for the default TicketRegistry which stores
  the tickets in-memory and cleans them out at specified
  intervals.
 </description>

 <bean id="ticketRegistry"
 class="org.jasig.cas.ticket.registry.JpaTicketRegistry">
 <constructor-arg index="0" ref="entityManagerFactory" />
 </bean>

 <bean id="ticketRegistryCleaner"
 class="org.jasig.cas.ticket.registry.support.DefaultTicket
 RegistryCleaner"
 p:ticketRegistry-ref="ticketRegistry"/>

 <bean id="ticketRegistryCleanerJobDetail"
 class="org.springframework.scheduling.quartz.Method
 InvokingJobDetailFactoryBean"
 p:targetObject-ref="ticketRegistryCleaner"
 p:targetMethod="clean"/>

 <bean id="periodicTicketRegistryCleanerTrigger"
 class="org.springframework.scheduling.quartz.SimpleTrigger
 Bean"
 p:jobDetail-ref="ticketRegistryCleanerJobDetail"
 p:startDelay="20000"
 p:repeatInterval="1800000"/>

</beans>
```

18.2. For CAS 3.4.2.1, replace the entire file contents as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:p="http://www.springframework.org/schema/p"
 xsi:schemaLocation="http://www.springframework.org/schema/
 beans http://www.springframework.org/schema/beans/
 spring-beans-2.0.xsd">
 <description>
  Configuration for the JPA based TicketRegistry
```

```
    which stores the tickets in the db and cleans them out at
    specified intervals.
  </description>
  <bean id="ticketRegistry"
    class="org.jasig.cas.ticket.registry.JpaTicketRegistry">
    <constructor-arg index="0" ref="entityManagerFactory" />
  </bean>
  <bean id="ticketRegistryCleaner"
    class="org.jasig.cas.ticket.registry.support.
    DefaultTicketRegistryCleaner"
    p:ticketRegistry-ref="ticketRegistry"
    p:lock-ref="cleanerLock"/>
  <bean id="cleanerLock"
    class="org.jasig.cas.ticket.registry.support.
    JdbcLockingStrategy"
    p:uniqueId="${host.name}"
    p:platform="${ticket.cleaner.database.platform}"
    p:applicationId="cas-ticket-registry-cleaner"
    p:dataSource-ref="dataSource"/>
  <bean id="ticketRegistryCleanerJobDetail"
    class="org.springframework.scheduling.quartz.
    MethodInvokingJobDetailFactoryBean"
    p:targetObject-ref="ticketRegistryCleaner"
    p:targetMethod="clean"/>
  <bean id="periodicTicketRegistryCleanerTrigger"
    class="org.springframework.scheduling.quartz.
    SimpleTriggerBean"
    p:jobDetail-ref="ticketRegistryCleanerJobDetail"
    p:startDelay="20000"
    p:repeatInterval="1800000"/>
</beans>
```

19. Save and close `ticketRegistry.xml`.

20. For CAS 3.2.1.1 and CAS 3.3.1, ignore this step. Configuration is complete.

    For CAS 3.4.2.1, log in to the Oracle database using the Oracle account specified in step 11.

21. For CAS 3.2.1.1 and CAS 3.3.1, ignore this step. Configuration is complete.

    For CAS 3.4.2.1, execute the following DDL statements to create the table that CAS requires for persistence:

```
CREATE TABLE LOCKS (
   APPLICATION_ID VARCHAR(50) NOT NULL,
   UNIQUE_ID VARCHAR(50) NULL,
   EXPIRATION_DATE TIMESTAMP NULL
);
ALTER TABLE LOCKS ADD CONSTRAINT LOCKS_PK
PRIMARY KEY (APPLICATION_ID);
COMMIT;
```

Step 9 Modify WEB-INF/classes/default_views.properties

Note

Skip this step if you are using the /samlValidate service. ■

Use the following steps to implement the CAS validation success and failure views.

1. Open `WEB-INF/classes/default_views.properties`.

2. Add the following lines:

```
## Banner Applications Views
bannerAccountServiceSuccessView.(class)=com.sghe.cas.view.
BannerAccountSuccessResponseView
bannerAccountServiceFailureView.(class)=com.sghe.cas.view.
BannerAccountFailureResponseView
```

3. Save and close `default_views.properties`.

Step 10 Deploy application

Use the following steps to deploy the application so the CAS server can provide service for /bannerValidate.

1. Create the `cas.war` distribution by including all the files and directories under `SGHE_CAS_HOME`.

2. Change to the `SGHE_CAS_HOME\cas-server-<version>` directory.

3. Use the jar command to create the `cas.war` file:

```
jar -cvf ..\cas.war .
```

4. Deploy the application to any servlet container.

5. Ensure that SSL is enabled in your servlet container.

If you are using Tomcat, copy `cas.war` to the Tomcat installation `webapps` directory. Tomcat automatically deploys the application.

Step 11 Test CAS Web application

Use the following steps to test the CAS Web application.

1. Use the following URL to access the CAS login page:

```
https://<host>:<port>/<cas context root>
```

For example, `<cas context root>` could be `cas`. An example of how this is pre-configured with the integrated Luminis Platform 5.x CAS instance would be `cas-web`.

2. Test the login page by specifying the CAS administration user that was defined in step 8. This must be an existing LDAP user. (The documentation example defined a CAS administration user named *triddle* with the password *123456*.)

Step 12 Modify Tomcat (if needed)

If the test results in an error and you deployed the `cas.war` application to Tomcat 6, the following Tomcat changes are needed for CAS.

Windows

1. Add the following line to `catalina.bat` in the Tomcat `installation/bin` directory:

   ```
   set JAVA_OPTS="-Dorg.apache.jasper.compiler.
   Parser.STRICT_QUOTE_ESCAPING=false"
   ```

2. Restart Tomcat to implement the change.

Windows service

1. Select Start > All Programs > Apache Tomcat 6.0 > Configure Tomcat. The Apache Tomcat Properties window is displayed,

2. Select the Java tab.

3. Enter the following command in the Java Options window:

   ```
   -Dorg.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING=false
   ```

4. Restart the Tomcat Windows service to implement the change.

Unix

1. Add the following line to `catalina.sh` in the Tomcat `installation/bin` directory:

   ```
   JAVA_OPTS="-Dorg.apache.jasper.compiler.Parser.
   STRICT_QUOTE_ESCAPING=false":$JAVA_OPTS
   ```

2. Restart Tomcat to implement the change.

Step 13 Define CAS managed services

In the default configuration, all http and https URLs are protected. If you want to protect specific URLs, you can define the following CAS managed services:

- One managed service protects the CAS managed services URL (/<cas context root>/services). For example, <cas context root> could be cas. An example of how this is pre-configured with the integrated Luminis Platform 5.x CAS instance would be cas-web.
- Another managed service protects the SSO Manager URL.

Note

Luminis Platform 5.x requires additional service definitions. Refer to the *Luminis Platform Banner Integration Setup Guide* for Luminis Platform 5.x examples of the CAS managed services.

Use the following steps to define the CAS managed services.

1. Launch a browser and go to the CAS server management page, which is at a URL similar to this:

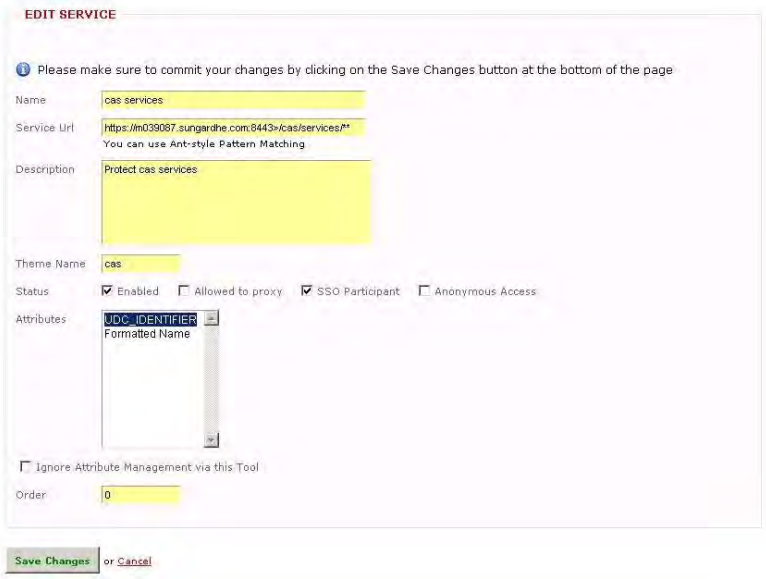
   https://<CAS server>:<port>/cas/services/manage.html

2. Enter a valid administrator user name and password (obtained from the CAS administrator). The Services Management page is displayed.

3. Click the Add New Service tab in the left corner of the page.

4. Add a new service by entering the following values:

Name	*cas services*
Service URL	*http(s)://<cas-server>:<port>/cas/services/**
Description	*Protect cas services*
Theme Name	*cas*
Status	Select *Enabled* and *SSO Participant*
Attributes	Select *UDC_IDENTIFIER*

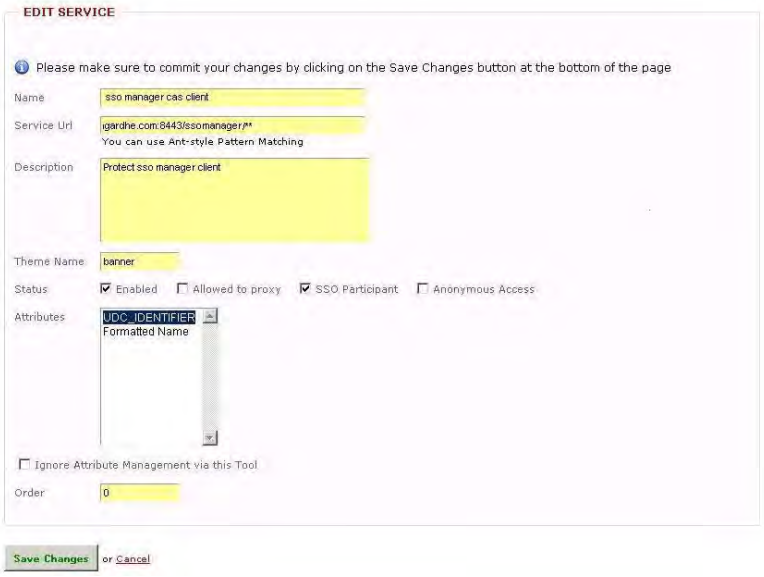


5. Click **Save Changes**. The CAS server is now protecting the `/cas/services` URL.

6. Click the Add New Service tab in the left corner of the page.

7. Add a new service by entering the following values:

Name	*sso manager cas client*
Service URL	*http(s)://<sso-manager-server>:<port>/ssomanager/**
Description	*Protect sso manager client*
Theme Name	*banner*
Status	Select *Enabled* and *SSO Participant*
Attributes	Select *UDC_IDENTIFIER*



8. Click **Save Changes**. The CAS server is now protecting the SSO Manager URL.

Configuration of Self-Service Banner for CAS

Once the CAS server is configured to support BEIS, you can configure Self-Service Banner (SSB) to participate in a CAS single sign on (SSO) session. This is accomplished with the SSO Manager, a BEIS component that provides an SSO gateway for Internet-native Banner and Self-Service Banner.

Single sign on makes it easier for users to navigate from a portal to Self-Service Banner where they can manage Banner data. When using the SSO Manager for SSO, SSB is protected by a central access manager that authenticates users. The central access manager can be JA-SIG Central Authentication Service (CAS) or a third-party access manager such as Oracle® Access Manager. When using the Central Authentication Service (CAS) for single sign on, SSB is accessed through the following URL:

```
http(s)://<host>:<port>/ssomanager/c/SSB
```

Configuration of Internet-native Banner for CAS

Once the CAS server is configured to support BEIS, you can configure Internet-native Banner (INB) to participate in a CAS single sign on (SSO) session. This is accomplished with the SSO Manager, a BEIS component that provides an SSO gateway for Internet-native Banner and Self-Service Banner.

When using the SSO Manager for SSO, INB is protected by a central access manager that authenticates users. The central access manager can be JA-SIG Central Authentication Service (CAS) or a third-party access manager such as Oracle® Access Manager. When using the Central Authentication Service (CAS) for single sign on, INB is accessed through the following URL:

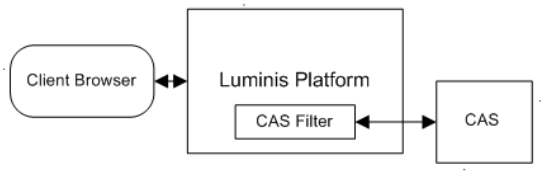
```
http(s)://<host>:<port>/ssomanager/c/INB
```

Configuration of Luminis Platform for CAS

Note

This section applies to configuring Luminis Platform 4.x for CAS-based SSO using Banner Enterprise Identity Services. If you have installed Luminis Platform 5.x, you can skip this section because Luminis Platform 5.x is already configured to participate in CAS-based SSO.

Once the CAS server is configured to support BEIS, you can configure Luminis Platform to participate in a CAS single sign on (SSO) session.



The following steps must be completed for Luminis Platform to communicate with CAS:

- Step 1, “Configure Luminis Platform with CAS server information”
- Step 2, “Add Luminis service to CAS server”

Step 1 Configure Luminis Platform with CAS server information

A script located in the `$CP_ROOT/bin/banneridm` directory handles the CAS configuration for Luminis Platform. The `banneridm` script has the following commands:

```
banneridm setcashost -t host-name -r port
banneridm importcascert certificate-file [-p password]
banneridm enable
banneridm disable
banneridm -h
```

The options in these commands have the following parameters:

Parameter	Description
`setcashost`	Sets CAS server host and port configuration properties
`-t host-name`	Sets host name where the CAS server is running
`-r port`	Sets port where the CAS server is running
`importcascert`	Imports the CAS certificate into Luminis Platform
`certificate-file`	Sets CAS server certificate file name to install into Luminis Platform
`-p password`	Sets Java keystore password. Defaults to the keystore password specified when Luminis Platform was installed.
`enable`	Turns on the Luminis Platform connection
`disable`	Turns off the Luminis Platform connection
`-h`	Displays help

Before the `banneridm` script can be run, the following CAS server information is needed from the CAS server administrator:

CAS server host	Looks something like *https://slctests12n.sct.com*
CAS server port	Looks something like *8843*
CAS server SSL certificate chain	X509 certificate that is imported using the Java keytool

Use the following steps to configure Luminis Platform for CAS.

1. Log in to the Luminis Platform server with administrative privileges.
2. Open a command window (Cygwin on Windows servers).
3. Use the following command to set the `setcashost` option:

   ```
   banneridm setcashost -t host-name -r port
   ```

   This command sets the configuration values used by the other options.
4. Enter the following command to install the CAS server certificate:

   ```
   banneridm importcascert certificate -file <-p password>
   ```
5. Use the following command to enable Luminis Platform to use a CAS service for authentication:

   ```
   banneridm enable
   ```
6. Restart Luminis Platform using the following commands:

   ```
   stopcp
   startcp
   ```

You can use the disable command to return Luminis Platform to the original install status.

Step 2 Add Luminis service to CAS server

You must add a Luminis service to the list of services that the CAS server protects. This can be done by the CAS server administrator or by the Luminis Platform administrator with a valid administrator username and password supplied by the CAS server administrator. Use the following steps to add a Luminis service.

1. Launch a browser and go to the CAS server management page. The CAS server management page is at a URL that looks like this:

   ```
   https://<CAS server>:<port>/<version>/services/manage.html
   ```
2. Enter a valid administrator user name and password (obtained from CAS administrator). The Services Management page is displayed.
3. Click the Add New Service tab in the left corner of the page.

4. Add a new service by entering the following values:

Name	*Luminis*
Service URL	*http(s)://<Luminis server name>:<port>/sso/***
Description	*Luminis Platform*
Theme Name	Not used, but a value must be entered
Status	Check *Enabled, Allowed to proxy, SSO Participant*
Attributes	Select *uid, UDC_IDENTIFIER*

5. Click **Save Changes**. The CAS server is now configured to communicate with the Luminis Platform server.

Configuration of Banner Workflow for CAS

You must first configure the CAS server to support Banner Workflow (see "Configuration of the CAS server" on page D-2). Then you must configure Banner Workflow for CAS (see Chapter 14, "Banner Workflow Configuration").

Configuration of Banner Document Management Suite for CAS

You must first configure the CAS server to support BEIS (see "Configuration of the CAS server" on page D-2). Then you must configure Banner Document Management Suite for CAS (see Chapter 15, "Banner Document Management Suite Configuration").

4 Country View Road
Malvern, Pennsylvania 19355
United States of America
www.sungardhe.com