Oracle® Fusion Middleware

WebCenter Sites Installation Guide

11*g* Release 1 (11.1.1.8.0)

E29632-04

December 2016

Oracle Fusion Middleware WebCenter Sites Installation Guide, 11*g* Release 1 (11.1.1.8.0)

E29632-04

Contents

6 Procedures for Installing WebCenter Sites

7 WebCenter Sites Post-Installation Steps

Part II Installing Oracle WebCenter Sites: Satellite Server

8 Overview of Satellite Server

9 Procedures for Installing Remote Satellite Servers

10 Satellite Server Post-Installation Steps

Part III Internationalizing Oracle WebCenter Sites

11 Configuring Your Environment for Multi-Language Operations

Part IV Installing Oracle WebCenter Sites: Community-Gadgets

12 Overview of Community-Gadgets

18 Verifying and Implementing Community-Gadgets

19 Adding Sample Gadgets

Part V Installing the Community Blogs Module

20 Procedures for Installing the Community Blogs Module

Part VI Installing Oracle WebCenter Sites: Site Capture

21 Overview of WebCenter Sites: Site Capture

22 Configuring the Site Capture Application Server

23 Procedures for Installing Site Capture

24 Enabling Publishing-Triggered Site Capture

Part VII Installing Oracle WebCenter Sites: Analytics

25 Overview of Analytics Architecture

26 Prerequisites for Installing Analytics

27 Procedures for Installing Analytics

28 Tuning Analytics Configuration Parameters

29 Configuring Visitor Detection

30 Verifying Your Analytics Installation

Part VIII Installing Oracle WebCenter Sites: Content Integration Platform

31 Installing Content Integration Platform for File Systems and Microsoft SharePoint

32 Installing Content Integration Platform for EMC Documentum

Part IX Reference: Backing Up and Recovering

33 Backup and Recovery

Preface

This guide contains information about installing Oracle WebCenter Sites, Satellite Server, Community-Gadgets, Community Blogs Module, Site Capture, Analytics, and content integration platform. It also includes procedures to configure environment for multi-lingual operations as well as back up and recover WebCenter Sites.

Audience

This guide is intended for installation engineers with experience installing and configuring enterprise-level software, including databases, database drivers, application servers, web servers, and LDAP servers.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents for the current release:

- *Oracle WebCenter Sites Certification Matrix*
- *Oracle WebCenter Sites Release Notes*
- *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*

- *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*
- *Oracle Fusion Middleware WebCenter Sites Developer's Guide*

Graphics in This Guide

Graphics in this guide are screen captures of dialog boxes and similar windows that you will interact with during the installation or configuration process. The graphics are presented to help you follow the installation and configuration processes. They are not intended to be sources of information such as parameter values, options to select, and product version numbers.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Guide

The *Oracle Fusion Middleware WebCenter Sites Installation Guide* consolidates installation documents that were once available in separate books. This guide provides information about installing and configuring Oracle WebCenter Sites on Oracle WebLogic, Apache Tomcat, and IBM WebSphere application servers. It also provides detailed procedures for installing other WebCenter Sites applications. Additionally, it includes information about configuring your environment for multi-language operations, as well as backing up and recovering your environment.

This book includes the following parts:

- Part I, "Installing Oracle WebCenter Sites"
- Part II, "Installing Oracle WebCenter Sites: Satellite Server"
- Part III, "Internationalizing Oracle WebCenter Sites"
- Part IV, "Installing Oracle WebCenter Sites: Community-Gadgets"
- Part V, "Installing the Community Blogs Module"
- Part VI, "Installing Oracle WebCenter Sites: Site Capture"
- Part VII, "Installing Oracle WebCenter Sites: Analytics"
- Part VIII, "Installing Oracle WebCenter Sites: Content Integration Platform"
- Part IX, "Reference: Backing Up and Recovering"

Part I

Installing Oracle WebCenter Sites

Part I contains the following chapters:

1

Overview of the WebCenter Sites Installation Process

This chapter provides information that will help you prepare to install WebCenter Sites on the following application servers: Oracle WebLogic Server, Apache Tomcat, and IBM WebSphere, in standalone and clustered environments.

The following environments are supported:

- Single Server Environment: This is an environment consisting of a single WebCenter Sites installation that may or may not have a front-end web server.
- Multiple Server Load Balanced environment: This is an environment consisting of multiple WebCenter Sites installations that share the same database and shared directory, and use a front-end load balancer. The application servers where the WebCenter Sites applications are deployed are standalone servers and do not belong to a cluster.
- Multiple Server Cluster environment: This is an environment consisting of multiple WebCenter Sites installations that share the same database and shared directory, and use a front-end load balancer. The application servers on which the WebCenter Sites applications are deployed belong to the same cluster. The cluster may be configured to support session persistence. Only horizontal clustering is supported for this environment.

This guide does not cover the following topics, as they fall outside of its scope:

- LDAP integration. A summary is available in Section 7.4, "Integrating with LDAP (Optional)." For information about this topic, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.
- Integrating Oracle Access Manager (OAM) with WebCenter Sites. For information about this topic, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

This chapter contains the following sections:

- Section 1.1, "Installation Quick Reference"
- Section 1.2, "Next Step"

1.1 Installation Quick Reference

Before installing WebCenter Sites, you will install and configure its supporting software on each development, content management, and delivery system on which you plan to use WebCenter Sites. You will then run the WebCenter Sites installer once for each server for each development, content management, and delivery system.

During the installation process for development and content management, you will have the option to install sample sites and sample content. When the installation process ends, you will complete post-installation steps to verify your WebCenter Sites systems and if necessary, configure additional support, such as OAM or LDAP.

The steps below summarize the installation process of configuring WebCenter Sites supporting software, installing WebCenter Sites, verifying its functionality, and completing post-installation steps. Keep the steps handy as a quick reference to installation procedures and to which chapters provide detailed instructions.

Complete the steps below for all deployments.

1. **Complete the Prerequisite Steps**

   In this step, you will prepare your environment for the installation process by completing a number of steps that also include setting up a supported database, installing a supported application server, and if necessary for your requirements, installing a supported web server. For detailed information, see Chapter 2, "Prerequisites for Installing WebCenter Sites."

2. **Set Up the Application Server**

   In this step, you will do the following:

   a. Configure your application server, as described in one of the following sections, specific to your application server:

      – Section 3.2, "Step 1: Configuring WebLogic Application Server"

      – Section 4.2, "Step 1: Configuring Tomcat Application Server"

      – Section 5.2, "Step 1: Configuring WebSphere Application Server"

   b. If you installed a web server, integrate it with the application server:

      – Section 3.3, "Step 2: Integrating Oracle WebLogic Server with a Supported Web Server"

      – Section 4.3, "Step 2: Integrating Tomcat Application Server with a Supported Web Server"

      – Section 5.3, "Step 2: Integrating WebSphere Application Server with a Supported Web Server"

3. **Install WebCenter Sites**

   For each application server configured in the environment, install WebCenter Sites by running the GUI installer or silent installer. For instructions, see Chapter 6, "Procedures for Installing WebCenter Sites."

   There will be a point during an installation when the installer will prompt you to perform a number of installation actions. At this point you will have to manually deploy the WebCenter Sites web application (on WebLogic and WebSphere) and follow the remaining steps to complete the installation. (Instructions are available in Section 6.5, "Deploying the WebCenter Sites and CAS Web Applications.")

4. **Complete the Post-Installation Steps**

   Complete the steps that are appropriate for your system. Refer to Chapter 7, "WebCenter Sites Post-Installation Steps" for full instructions and to the list below for a summary.

   a. WebCenter Sites contains a modified version of the Microsoft XML Parser (`MSXML.jar` in the `WEB-INF/lib` directory). If a different version of the Microsoft XML Parser is referenced in the WebCenter Sites `CLASSPATH` environment variable, you must change the path to refer to the version used by WebCenter Sites; otherwise, WebCenter Sites will fail when parsing. For information, see Section 7.1, "Loading the XML Parser."

   b. If you wish to add trusted URLs for CAS redirects, follow the steps in Section 7.2, "Specifying Trusted URLs for CAS."

   c. Verify the WebCenter Sites installation by logging in to Oracle WebCenter Sites as the general administrator and accessing the WebCenter Sites Admin interface and Contributor interface. For instructions, see Section 7.3, "Verifying the WebCenter Sites Installation."

   d. If you need to perform LDAP integration, complete the steps in Section 7.4, "Integrating with LDAP (Optional)."

   e. If you wish to replace CAS with Oracle Access Manager (OAM), follow instructions Section 7.5, "Integrating Oracle Access Manager (OAM) with WebCenter Sites (Optional)."

5. Install remote Satellite Server as described in Chapter 9, "Procedures for Installing Remote Satellite Servers."

6. **Set Up WebCenter Sites for Its Business Purpose**

   Once the entire installation is completed and verified, set up WebCenter Sites for its business purpose. For instructions, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

1.2 Next Step

Continue to Chapter 2, "Prerequisites for Installing WebCenter Sites."

2

Prerequisites for Installing WebCenter Sites

This chapter contains the following sections:

- Section 2.1, "Before You Begin"
- Section 2.2, "Next Step"

2.1 Before You Begin

- Refer to the following documentation for various information about installing WebCenter Sites:
  - Refer to the *Oracle WebCenter Sites Certification Matrix* to select a platform on which to install WebCenter Sites. You will install and configure components of the platform as described in this guide.

    Only selected information regarding the configuration of third-party products is given in this guide. For detailed information and to obtain the latest e-fixes, patches, and service packs, refer to third-party product vendors' documentation and release notes.
  - Refer to *Oracle WebCenter Sites Release Notes* for last-minute information concerning the WebCenter Sites installation process and related procedures.
  - Refer to the guide named *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software* for instructions on installing supported databases, application servers, and web servers, and for procedures on integrating with software required by WebCenter Sites.
- If you plan to run the GUI WebCenter Sites installer, ensure that your systems are capable of displaying the installer GUI. The installer will not work in text mode.

  You can install and deploy WebCenter Sites by either running the GUI installer or installing silently. When you run the GUI installer, a graphical interface guides you through the installation process, prompting you to enter information and select options as required. It also provides access to online help. To install silently, first generate an `omii.ini` file by running the GUI installer until the Settings Summary screen and then exiting. The silent installer uses this file to install WebCenter Sites.

  You will run the WebCenter Sites installer on all systems in your environment. There are two system types: content management or development (whose installation processes are identical) and delivery. Content management and development systems run in the same mode but are used for different purposes.

Note: Keep in mind the following:

- The system type cannot be changed once you select the type and click **Next** (GUI installation), or you start running the silent installer.
- The installation process does not install all user interfaces on delivery systems, only a limited version of the WebCenter Sites Admin interface to enable the management of select features is installed in delivery.
- The names of the systems in your WebCenter Sites environment might differ from the names used in this document. Typically, the content management system is also called "staging," and the delivery system is also called "production."
- The WebCenter Sites installer includes the installation of CAS. By default, CAS will be installed on the primary server. If you will be running CAS on servers other than the primary server, deploy the CAS application and copy the required files at the midway point (when the installer pauses).
- If you plan to use OAM or another single sign-on provider instead of CAS, install WebCenter Sites to work with CAS. Once you have verified the installation, switch to the new single sign-on provider.

- Remove older versions of the Java Development Kit (JDK) from the `CLASSPATH` and `PATH` environment variables.

  WebCenter Sites requires a JDK, the product will not operate correctly with a Java Runtime Environment.

  Note: For clustered installations, ensure date-time in all nodes keep synchronized. In general, this can be achieved by configuring your system clock to synchronize with a time authority using Network Time Protocol (NTP).

- Create the directory into which you are installing WebCenter Sites. The directory name (`<sites_install_dir>` in this guide) and path cannot contain spaces or non-ASCII characters, and the application server must be able to read from and write to the directory. In a cluster environment, the path for this directory must be the same for all WebCenter Sites servers.
- For clustered installations, ensure you have created a shared file system directory that all cluster members can read from and write to; the directory name and path cannot contain spaces. This directory must be mounted as the same path on all cluster member systems.
- Create a directory for WebCenter Sites temporary files. This directory should be unique for each WebCenter Sites server.
- For all types of installations, add the `JDK/bin` directory to the path variable.
- Install a supported database.

  WebCenter Sites requires access to a database specifically configured for WebCenter Sites. Refer to the *Oracle WebCenter Sites Certification Matrix* for the list of supported databases (and other third-party components).

Before installing any other of WebCenter Sites supporting software, complete the following steps:

1. Install the database management system.

   For instructions, refer to the product vendor's documentation.

2. Create and configure a database for WebCenter Sites.

   For instructions, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

   Note that database configuration is identical across different application servers. Refer to the correct chapter to create and configure the database of your choice.

   Note that the database configuration is identical across different cluster servers.

- Install a supported application server on each server that will contain a WebCenter Sites installation. For more information, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.
- Install a supported web server as your environment requires. For more information, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

2.2 Next Step

Your next step is to prepare your application server for the WebCenter Sites installation process. For instructions, see one of the following chapters:

- Chapter 3, "Preparing to Install WebCenter Sites on Oracle WebLogic Application Server"
- Chapter 4, "Preparing to Install WebCenter Sites on Apache Tomcat Application Server"
- Chapter 5, "Preparing to Install WebCenter Sites on IBM WebSphere Application Server"

3

Preparing to Install WebCenter Sites on Oracle WebLogic Application Server

This chapter contains information about configuring WebLogic Server to support your WebCenter Sites web application.

This document contains the following sections:

- Section 3.1, "Quick Reference for Setting Up WebLogic Application Server"
- Section 3.2, "Step 1: Configuring WebLogic Application Server"
- Section 3.3, "Step 2: Integrating Oracle WebLogic Server with a Supported Web Server"

> **Note:** Before starting the procedures in this chapter, ensure you have completed the prerequisite steps described in Chapter 2, "Prerequisites for Installing WebCenter Sites."

3.1 Quick Reference for Setting Up WebLogic Application Server

This section contains the following topics:

- Section 3.1.1, "Summary of Steps"
- Section 3.1.2, "Paths and Directories Used in This Chapter"
- Section 3.1.3, "Start/Stop Commands for WebLogic Server"

3.1.1 Summary of Steps

Complete the steps below for each development, content management, and delivery environment on which you have installed the application server.

Step 1: Configuring WebLogic Application Server

1. Create a WebLogic domain, as shown in Section 3.2.1, "Creating a WebLogic Domain." This step requires you to create a domain with an admin server.
2. Configure the WebLogic domain as shown in Section 3.2.2, "Configuring the WebLogic Domain." This step requires you to:
   a. Create a machine for each physical server that will be running a WebLogic server. For instructions, see Section 3.2.2.1, "Creating Machines."
   b. If you are configuring a cluster environment, create and configure a cluster as described in Section 3.2.2.2, "Creating and Configuring a Cluster."

c. Create and configure one or more WebLogic managed servers. For instructions, see Section 3.2.2.3, "Creating and Configuring Managed Servers."

d. Create and configure a data source as described in Section 3.2.2.4, "Creating and Configuring a Data Source."

e. Set Java options, as described in Section 3.2.2.5, "Setting Java Options."

f. Set classpath, as described in Section 3.2.2.6, "Setting the Classpath."

g. If you are configuring a multi-server environment, transfer any remote managed servers to their remote physical servers, as described in Section 3.2.2.7, "Transfer Remote Managed Servers."

Step 2: Integrating WebLogic Application Server with a Supported Web Server

If your environment requires a web server, you have the option to integrate WebLogic application servers with an Oracle, Apache, or IIS web server, as described in:

- Section 3.3.1, "Configuring the Oracle HTTP Server"
- Section 3.3.2, "Configuring the Apache HTTP Server"
- Section 3.3.3, "Configuring the IIS Web Server"

3.1.2 Paths and Directories Used in This Chapter

Table 3–1 lists paths and directories used in this chapter.

Table 3–1 Paths and Directories Used in Configuring WebLogic Application Server

Name	Description
`<sites_install_dir>`	Path to the directory where WebCenter Sites is installed; the path does not include the name of the WebCenter Sites application.
`<shared_dir>`	Path to the shared folder on the given system; the path includes the name of the shared folder.
`<sites_deploy_home>`	The projected path of the exploded WebCenter Sites web application. The WebCenter Sites installer uses a specified deployment directory (`<deploy_home>`) and a specified WebCenter Sites application name (`<sites_app_name>`), and explodes the web application into the path `<deploy_home>/<sites_app_name>`.
`<weblogic_home>`	Path to the directory where WebLogic is installed. The path includes the name of the directory.
`<domain_home>`	Path to the WebLogic domain. The path includes the domain name.
`<apache_home>`	Path to the directory where the Apache web server is installed. The path includes the name of the directory.

3.1.3 Start/Stop Commands for WebLogic Server

This section lists commands that are used in this guide for managing WebLogic Server.

Note: We assume that you are using a UNIX-based system. For Windows-based installations, the commands end in either `.cmd` or `.bat` extensions.

- Start the admin server:

  ```
  <domain_home>/bin/startWebLogic.sh
  ```

- Stop the admin server:

  ```
  <domain_home>/bin/stopWebLogic.sh
  ```

- Start the node manager:

  ```
  <weblogic_home>/wlserver_10.3/server/bin/startNodeManager.sh
  ```

- Start a managed server:

  ```
  <domain_home>/bin/startManagedWebLogic.sh <managed_server_name> http://<admin_
  server_host>:<admin_port>
  ```

- Stop a managed server:

  ```
  <domain_home>/bin/stopManagedWebLogic.sh <managed_server_name> http://<admin_
  server_host>:<admin_port>
  ```

3.2 Step 1: Configuring WebLogic Application Server

To configure the WebLogic application server, you will complete the following steps:

- Section 3.2.1, "Creating a WebLogic Domain"
- Section 3.2.2, "Configuring the WebLogic Domain"

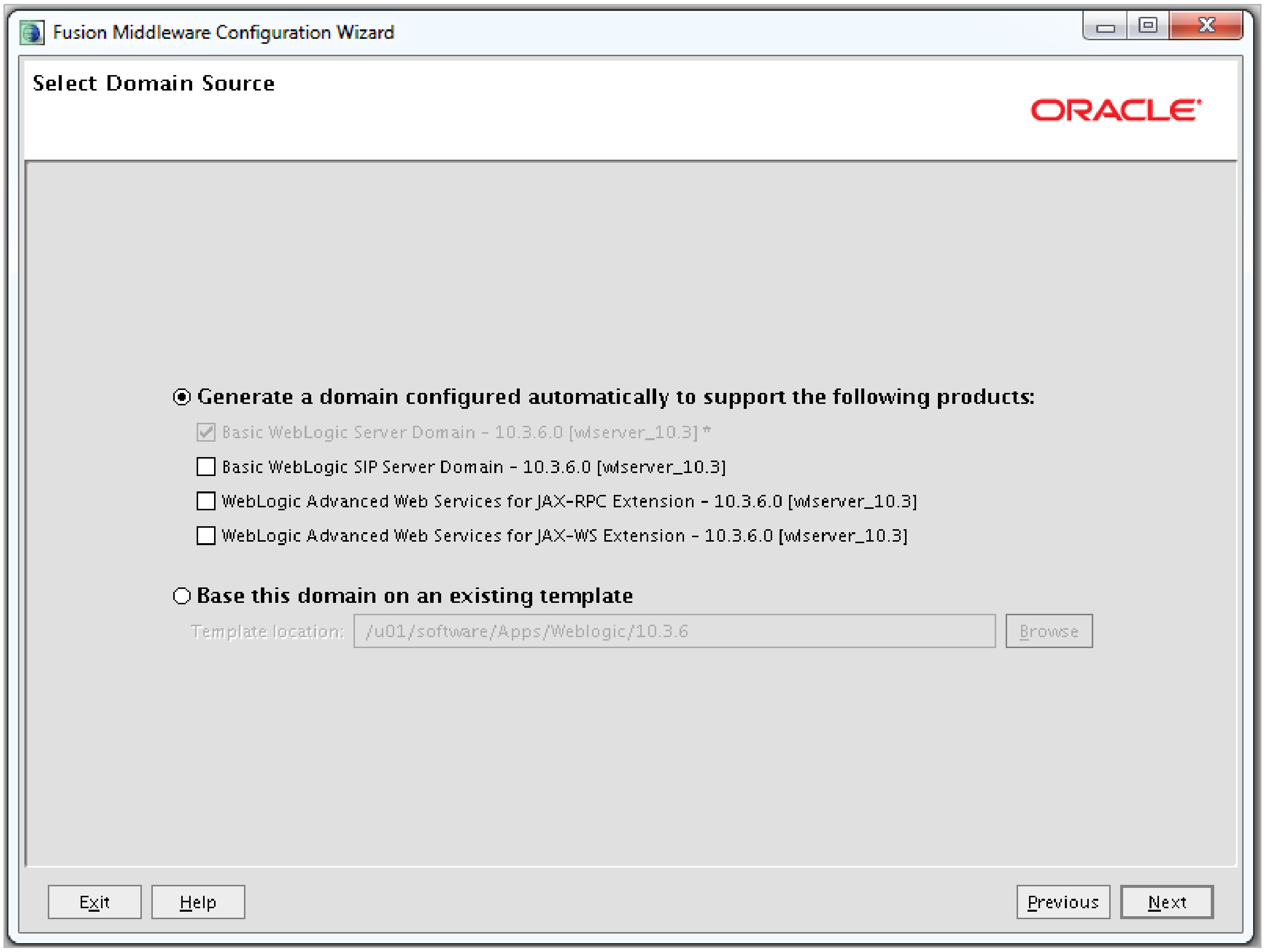
3.2.1 Creating a WebLogic Domain

In the steps below, you will be creating a WebLogic domain and configuring an admin server. In a multi-server non-clustered environment, you may create a single domain and administer all local and remote servers from a single administration console, or create a domain for each standalone server. In a clustered environment, create the domain on the primary cluster member.

To create and configure a WebLogic Server domain

1. Change directories to `<weblogic_home>/wlserver_10.3/common/bin` and run `config.sh` (`.cmd` for Windows).
2. Create a domain:
   a. In the Welcome screen, select **Create a new WebLogic domain** and click **Next**.
   b. In the Select Domain Source screen, select **Generate a domain configured automatically to support Basic WebLogic Server Domain** (Figure 3–1) and click **Next**.

Figure 3–1 Domain Source Selection



Fusion Middleware Configuration Wizard

Select Domain Source

● Generate a domain configured automatically to support the following products:

☑ Basic WebLogic Server Domain - 10.3.6.0 [wlserver_10.3] *

□ Basic WebLogic SIP Server Domain - 10.3.6.0 [wlserver_10.3]

□ WebLogic Advanced Web Services for JAX-RPC Extension - 10.3.6.0 [wlserver_10.3]

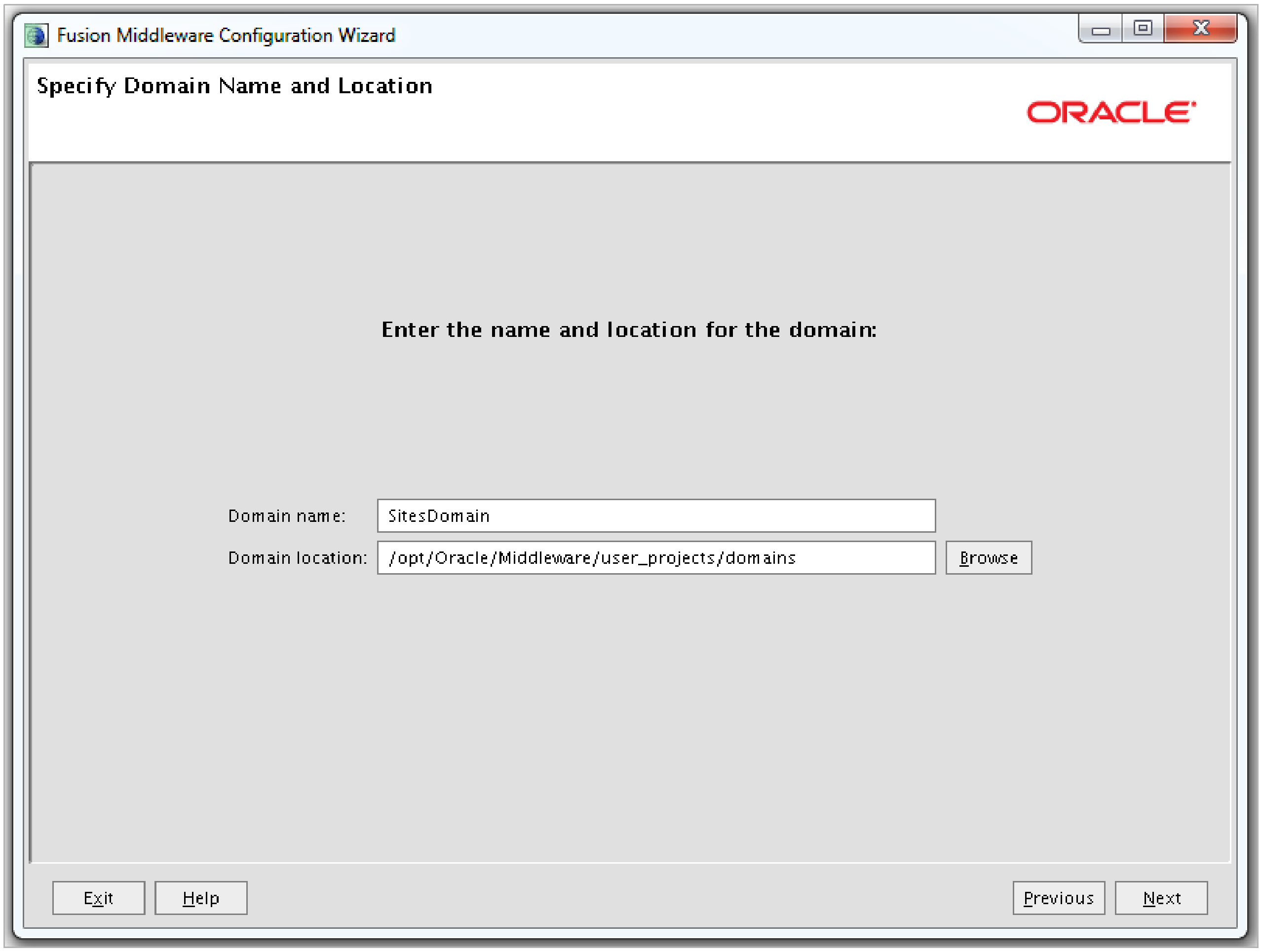
□ WebLogic Advanced Web Services for JAX-WS Extension - 10.3.6.0 [wlserver_10.3]

○ Base this domain on an existing template

Template location: /u01/software/Apps/Weblogic/10.3.6

c. In the Specify Domain Name and Location screen, enter the name and path of the WebLogic domain (Figure 3–2). Click **Next**.

Figure 3–2 Domain Name and Location



Fusion Middleware Configuration Wizard

Specify Domain Name and Location

Enter the name and location for the domain:

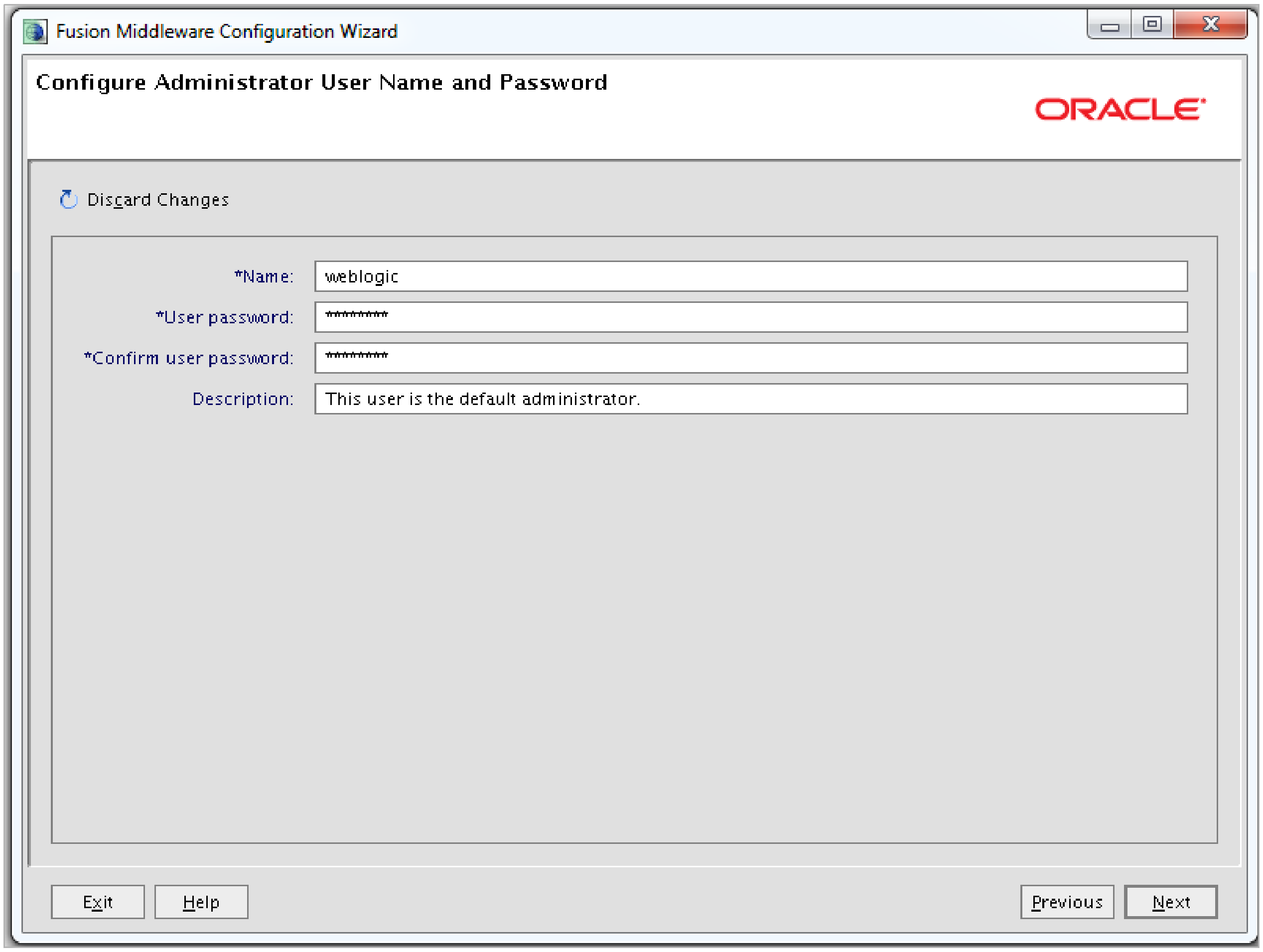
Domain name: SitesDomain

Domain location: /opt/Oracle/Middleware/user_projects/domains

Note: The path to the domain will be referred to throughout this guide as <domain_home>.

d. In the Configure Administrator Username and Password screen, enter a user name and password for the WebLogic domain (Figure 3–3). Click **Next**.

Figure 3–3 Administrator User Name and Password



Fusion Middleware Configuration Wizard
Configure Administrator User Name and Password
ORACLE
Discard Changes
Exit
Help
Previous
Next

*Name: weblogic

*User password: *********

*Confirm user password: *********

Description: This user is the default administrator.

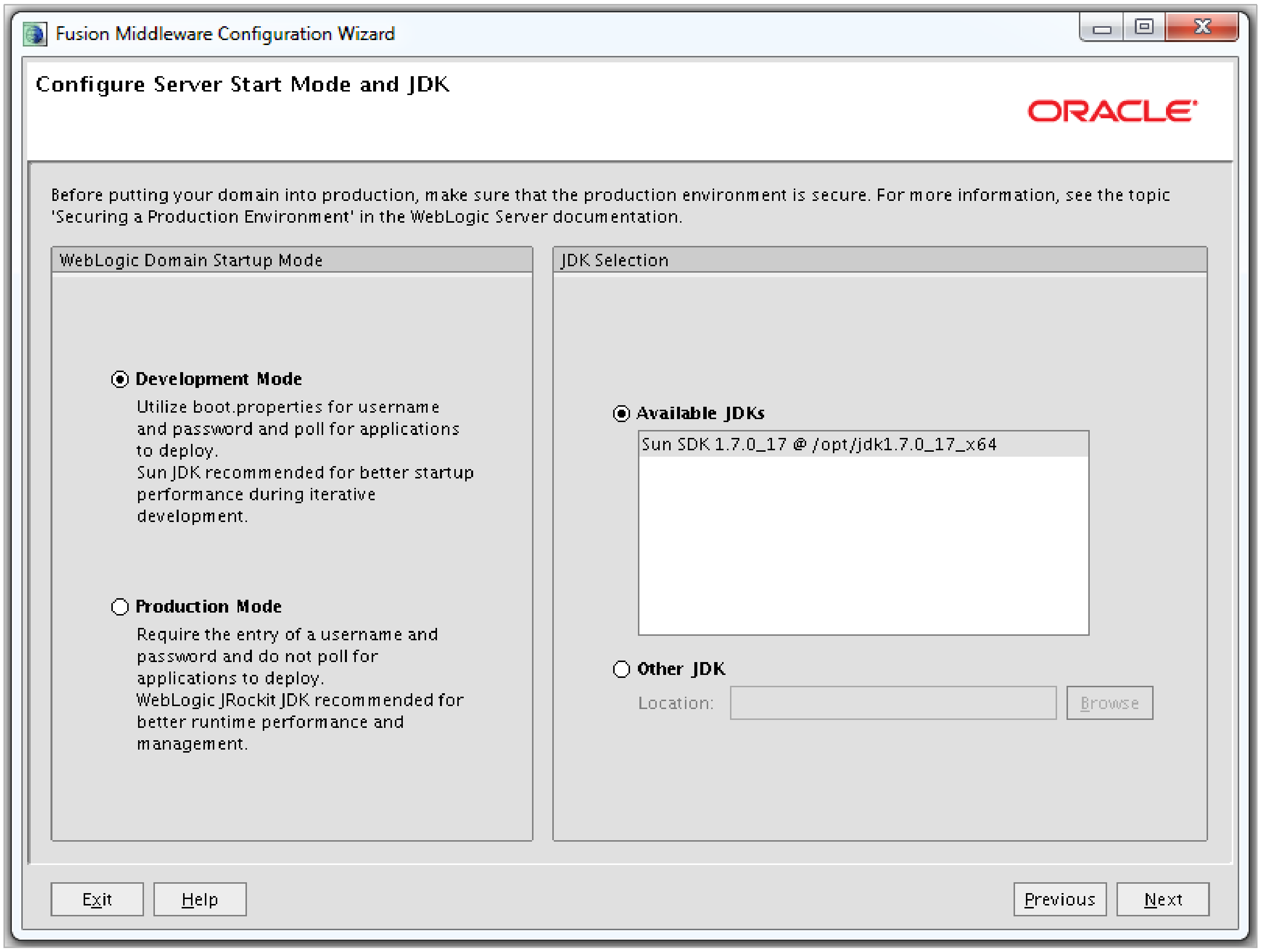
e. In the Configure Server Start Mode and JDK screen, select Development Mode and the preferred SDK (in the **WebLogic Supplied JDKs** list box). See Figure 3–4. For production environments, the domain will be changed to Production Mode later in this guide in Section 6.7, "Switching WebLogic Application Server to Production Mode (Delivery Systems Only)." Click **Next**.

Note: If a JRockit JDK is used, then you must create an optimization file containing the following:

```
{ match: [
"net/sf/ehcache/Cache.get*","com/fatwire/cache/ehcache/EhCacheObjec
tCache._
get*","com/fatwire/cache/ehcache/EhCacheObjectCache.get*","net/sf/e
hcache/Cache.isExpired*" ], hotspot : { enable : false },}
```

This file needs to be copied to each physical server where a WebLogic server is run.

Figure 3–4 Server Start Mode and JDK Configuration



Fusion Middleware Configuration Wizard

Configure Server Start Mode and JDK

Before putting your domain into production, make sure that the production environment is secure. For more information, see the topic 'Securing a Production Environment' in the WebLogic Server documentation.

WebLogic Domain Startup Mode

● Development Mode

Utilize boot.properties for username and password and poll for applications to deploy.
Sun JDK recommended for better startup performance during iterative development.

○ Production Mode

Require the entry of a username and password and do not poll for applications to deploy.
WebLogic JRockit JDK recommended for better runtime performance and management.

JDK Selection

● Available JDKs

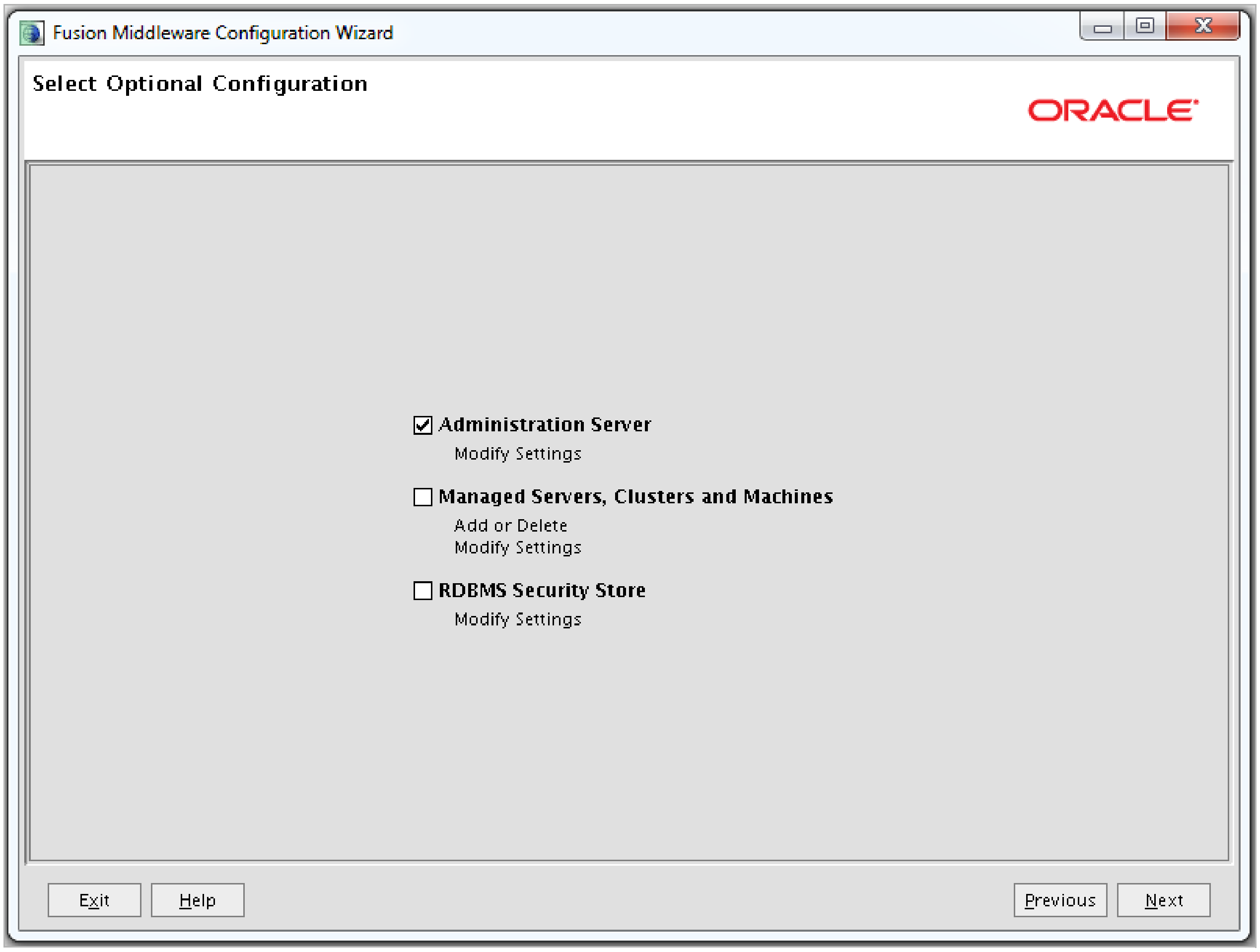
Sun SDK 1.7.0_17 @ /opt/jdk1.7.0_17_x64

○ Other JDK

Location:

f. In the Select Optional Configuration screen, select **Administration Server** (Figure 3–5). Managed Servers, Clusters, and Machines will be created and configured later in the guide. Click **Next**.

Figure 3–5 Optional Configuration



Select Optional Configuration

☑ Administration Server
Modify Settings

□ Managed Servers, Clusters and Machines
Add or Delete
Modify Settings

□ RDBMS Security Store
Modify Settings

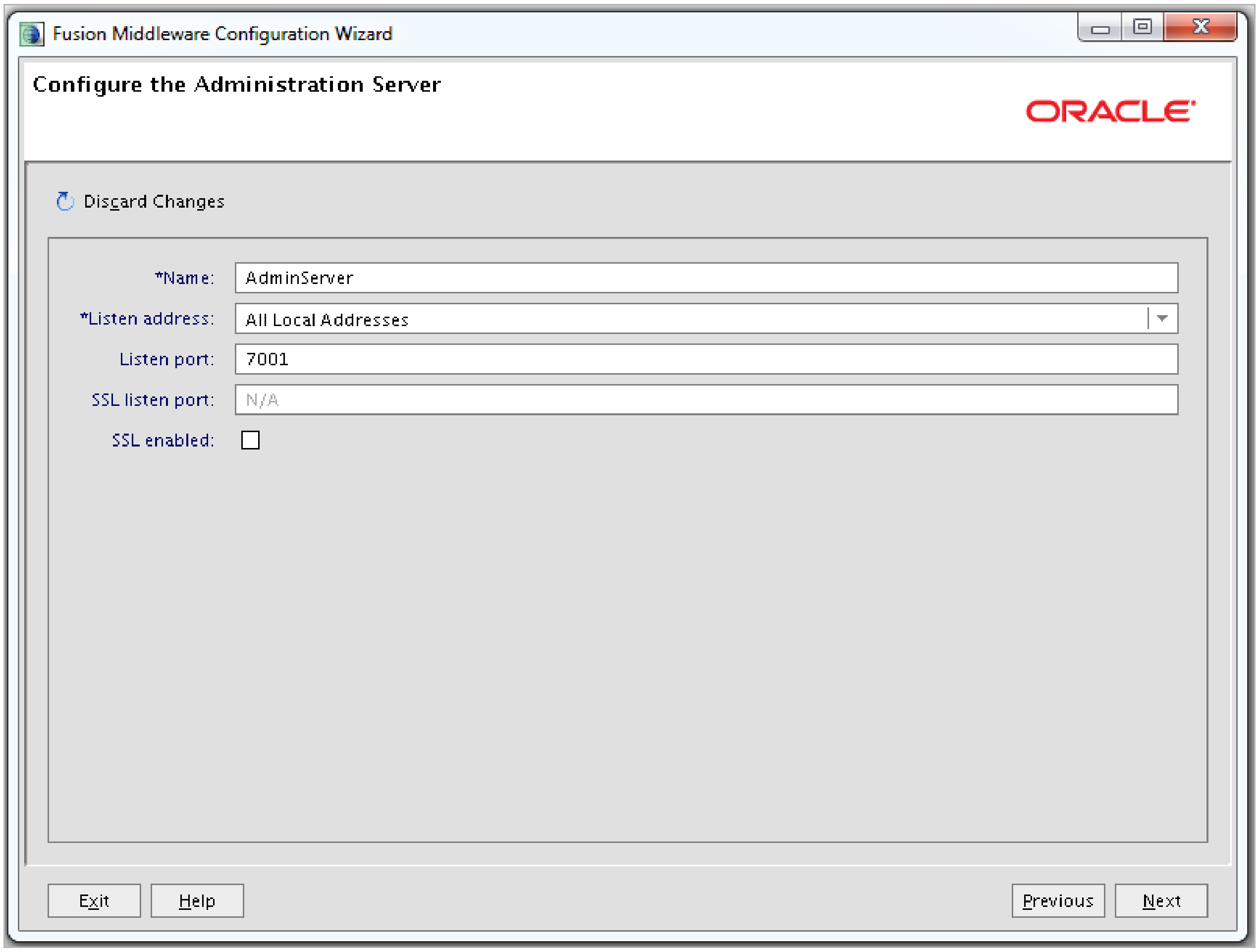
3. Configure the admin server (Figure 3–6):

   a. Enter the admin server's name, address, and port.

   b. If you want to use SSL, select **SSL enabled** and include the port for it to run on (recommended for Delivery).

   c. Click **Next**.

Figure 3–6 Administration Server Configuration



Fusion Middleware Configuration Wizard
Configure the Administration Server
ORACLE
Discard Changes
Exit
Help
Previous
Next

*Name: AdminServer

*Listen address: All Local Addresses

Listen port: 7001

SSL listen port: N/A

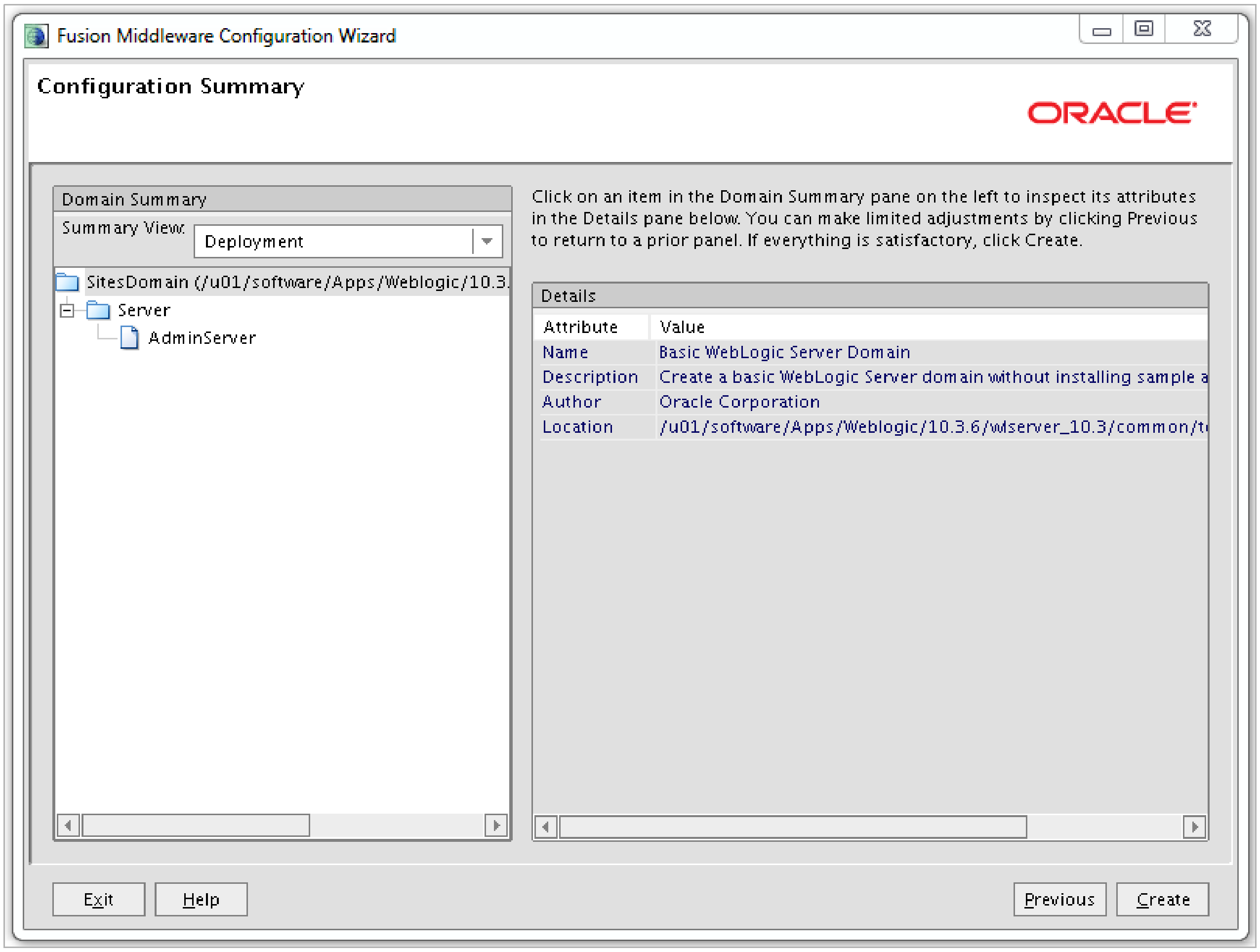
SSL enabled: □

Note: Throughout this guide, the values that you entered will be referred to as `<admin_server_host>` and `<admin_port>`.

---

4. Review the domain configuration (Figure 3–7) and click **Create**.

Figure 3–7 Configuration Summary



Configuration Summary

Domain Summary

Summary View: Deployment

- SitesDomain (/u01/software/Apps/Weblogic/10.3.
  - Server
    - AdminServer

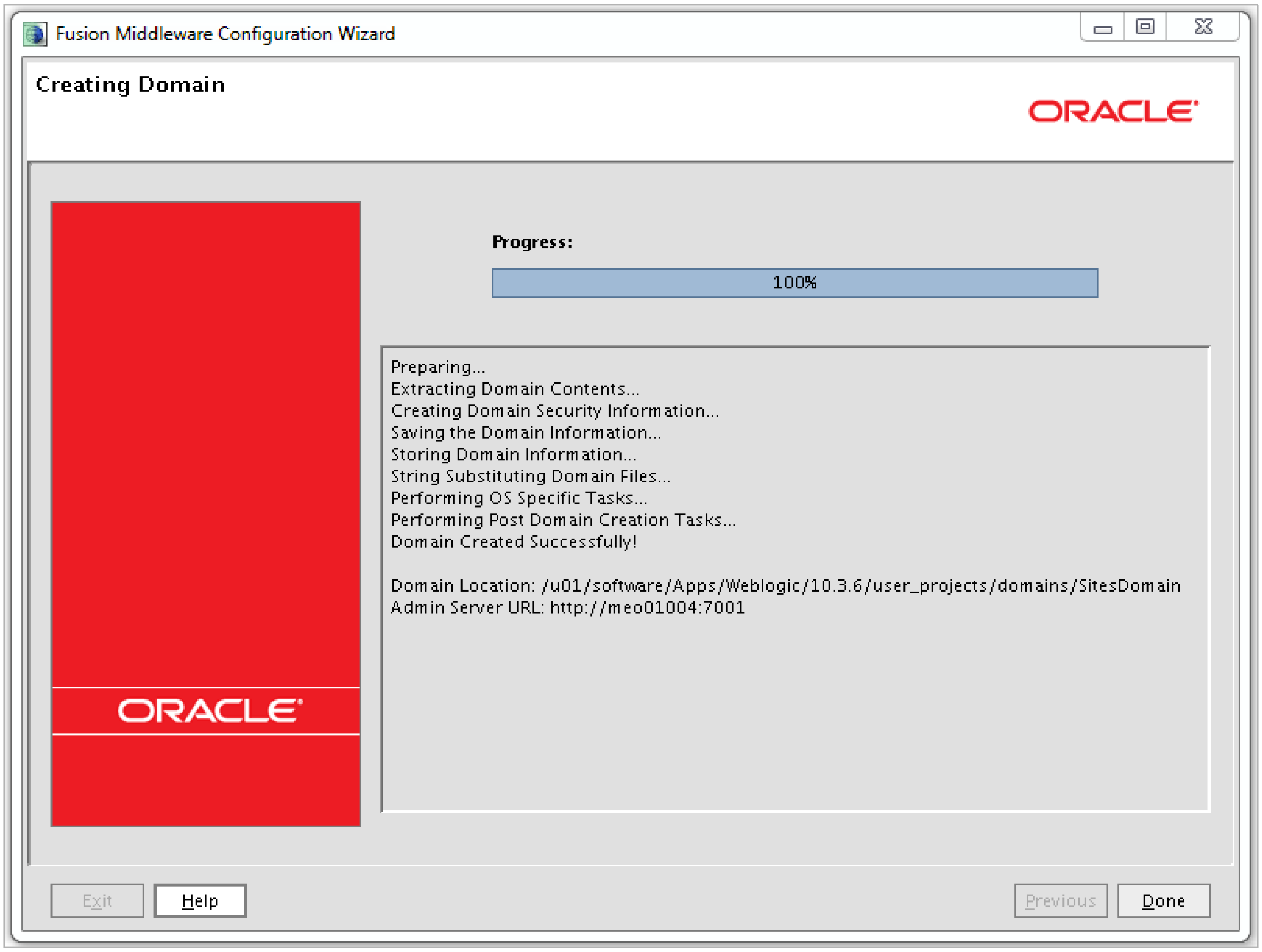
Click on an item in the Domain Summary pane on the left to inspect its attributes in the Details pane below. You can make limited adjustments by clicking Previous to return to a prior panel. If everything is satisfactory, click Create.

Details

Attribute	Value
Name	Basic WebLogic Server Domain
Description	Create a basic WebLogic Server domain without installing sample a
Author	Oracle Corporation
Location	/u01/software/Apps/Weblogic/10.3.6/wlserver_10.3/common/t

5. The domain installation begins (Figure 3–8). When the installation is complete, click **Done**.

Figure 3–8 Domain Creation



```
Preparing...
Extracting Domain Contents...
Creating Domain Security Information...
Saving the Domain Information...
Storing Domain Information...
String Substituting Domain Files...
Performing OS Specific Tasks...
Performing Post Domain Creation Tasks...
Domain Created Successfully!

Domain Location: /u01/software/Apps/Weblogic/10.3.6/user_projects/domains/SitesDomain
Admin Server URL: http://meo01004:7001
```

3.2.2 Configuring the WebLogic Domain

In this section, you will complete the steps listed below:

- Section 3.2.2.1, "Creating Machines"
- Section 3.2.2.2, "Creating and Configuring a Cluster"
- Section 3.2.2.3, "Creating and Configuring Managed Servers"
- Section 3.2.2.4, "Creating and Configuring a Data Source"
- Section 3.2.2.5, "Setting Java Options"
- Section 3.2.2.6, "Setting the Classpath"
- Section 3.2.2.7, "Transfer Remote Managed Servers"

3.2.2.1 Creating Machines

A machine is a logical representation of a physical server that hosts one or more WebLogic servers. The Administration Server uses a machine definition with Node Manager to start remote servers.

To create machines:

1. Start the administration server.

```
<domain_home>/bin/startWebLogic.sh
```

2. Log in to the WebLogic Server Administration Console:

   ```
   http://<admin_server_host>:<admin_port>/console
   ```

3. In the tree on the left, expand **Environment** and click **Machines**.
4. For each physical server that will have a WebLogic server running complete the following steps:
   1. Click **New**.
   2. Enter a **Name**.
   3. Select an OS.
   4. Click **Next**.
   5. If you want to use a secure listener, select **SSL**, otherwise select **Plain**.
   6. Enter a **Listen Address**.
   7. Modify the port as needed.
   8. Click **Finish**.

3.2.2.2 Creating and Configuring a Cluster

In a cluster environment, all WebLogic servers belong to a cluster, and communication between the servers is managed by the application server.

1. From the Administration Console, expand **Environment** and click **Clusters**.
2. Create a cluster:
   - a. Click **New**.
   - b. Enter a cluster name.
   - c. Select **Multicast**.
   - d. Modify the **Multicast Address** as needed.
   - e. Enter an unused port for **Multicast Port**.
   - f. Click **OK**.
3. (Optional) Enable Session Persistence:
   - a. Click the newly created cluster and select the **Replication** tab.
   - b. For Cross-cluster Replication Type, select **MAN (Synchronous) HTTP Session State Replication**.
   - c. Leave **Remote Cluster Address** blank.
   - d. For Replication Channel, enter a name for the channel to be used for replication traffic. This name should be unique for each cluster in the domain. Later in the guide, a network channel with this name will be created for every server in the cluster.
   - e. Click **Save**.
4. Continue to Section 3.2.2.3, "Creating and Configuring Managed Servers."

3.2.2.3 Creating and Configuring Managed Servers

For each server that will be managed by this Administration Console (all servers in a cluster environment), complete the steps below:

To create and configure managed servers:

1. From the Administration Console, expand **Environment** and click **Servers**.
2. Create a Managed Server:
   a. Click **New**.
   b. Enter a **Server Name**, **Server Listen Address**, and **Server Listen Port**.
   c. If the server is part of a single-server or multi-server load balanced environment, select **No, this is a stand-alone server**. If the server is part of a cluster environment, select **Yes, make this server a member of an existing cluster** and select the name of the cluster created in Section 3.2.2.2, "Creating and Configuring a Cluster."
   d. Click **Next**.
   e. Click **Finish**.
3. Associate the server with a Machine:
   a. Click the newly created server.
   b. For Machine, select the machine corresponding to the physical server this WebLogic server will be running on.
   c. Click **Save**.
4. In order to manipulate a server and its deployments using the `weblogic.Deployer` and `weblogic.Admin` utilities, tunneling must be enabled.

   Enable Tunneling:
   a. Click the **Protocols** tab.
   b. Select the **Enable Tunneling** checkbox.
   c. Click **Save**.
5. Create the Replication Channel:

   > **Note:** If this is not a cluster environment or session persistence will not be used, skip this step.

   a. From the **Protocols** tab, click the **Channels** tab.
   b. Click **New**.
   c. Enter the name of the replication channel specified in step 3 in Section 3.2.2.2, "Creating and Configuring a Cluster."
   d. For Protocol, select **http**.
   e. Click **Next**.
   f. Leave the addresses blank and the default ports.
   g. Click **Finish**.

3.2.2.4 Creating and Configuring a Data Source

This section explains how to create a data source for any of the supported databases using the Administration Console.

1. (DB2 Only) Add the IBM DB2 Driver jar files to the domain classpath.

   a. Stop the administration server.

   ```
   <domain_home>/bin/stopWebLogic.sh
   ```

   Note: `db2jcc4.jar` is not supported.

   ---

   b. Copy the `db2jcc.jar` and `db2jcc_license_cu.jar` from DB2 to a location that can be added to the domain classpath.

   c. Open `<domain_home>/bin/setDomainEnv.sh` in a text editor and search for the following line:

   ```
   # ADD EXTENSIONS TO CLASSPATHS
   ```

   d. Add the following line after it:

   ```
   PRE_CLASSPATH="<path_to_db2jcc.jar>:<path_to_db2jcc_license_cu.jar>:${PRE_
   CLASSPATH}"
   ```

   e. Save the changes.

   Note: In order to create a DB2 data source, you must first download and apply WebLogic Patches 13536596 and 13870996.

   ---

   f. Start the administration server.

2. From the Administration Console, expand **Services** and click **Data Sources**.

3. Create a Data Source:

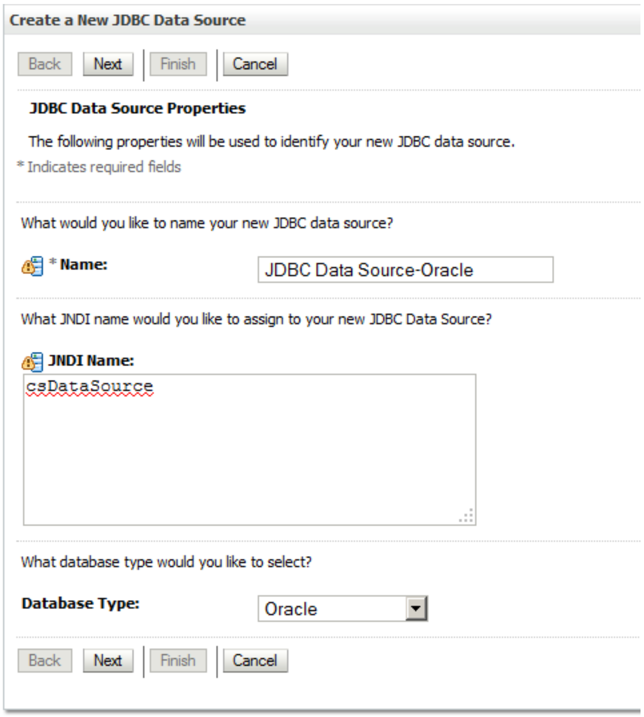
   a. Click **New** and select **Generic Data Source**.

   b. For JNDI name, enter a name. Throughout this guide, the value entered will be referred to as `<jndi_name>`.

   c. For Database Type, select one of the supported databases: Oracle, MS SQL Server, or DB2.

   d. Click **Next**.

Figure 3–9 New JDBC Data Source - JNDI Name and Database Type



Create a New JDBC Data Source

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

* Indicates required fields

What would you like to name your new JDBC data source?

* Name: JDBC Data Source-Oracle

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

JNDI Name: csDataSource

What database type would you like to select?

Database Type: Oracle

e. For Database Driver, select the option below that corresponds to the type chosen for step c:

```
Oracle - Oracle's Driver (Thin) for Instance connections;  Versions:9.0.1
and later
MS SQL Server - Oracle's MS SQL Server Driver (Type 4) Versions:7.0 and
later
DB2 - Other (IBM DB2 Universal JDBC Type 4 Driver will be used)
```

Note: XA Data Sources are not supported by WebCenter Sites.

---

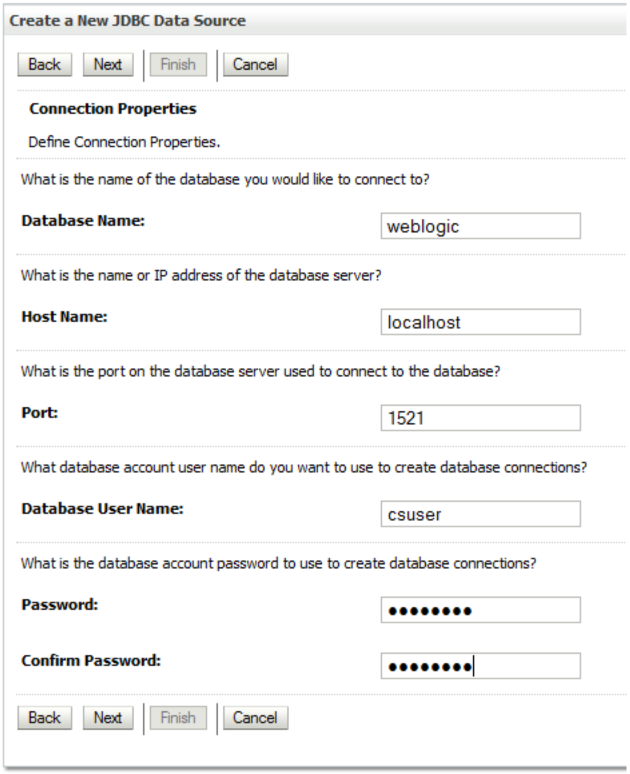
f. Click **Next**.

g. Check that Supports Global Transactions and One-Phase Commit are selected, and click **Next**.

h. For Connection properties, enter values for Database Name, Host Name, Port, Database User Name, and Password.

For DB2, only User Name and Password can be entered.

Figure 3–10 New JDBC Data Source - Connection Properties



Create a New JDBC Data Source

Connection Properties

Define Connection Properties.

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

Database Name: weblogic

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

Host Name: localhost

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

Port: 1521

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

Database User Name: csuser

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

Password: ••••••••

Confirm Password: ••••••••

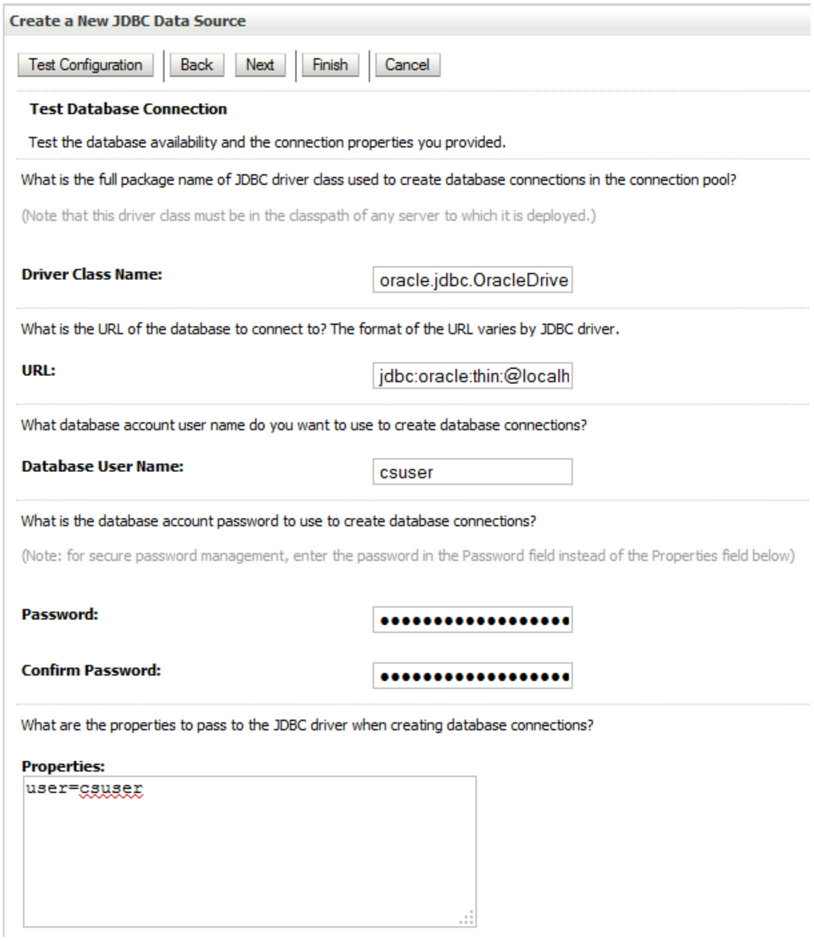
i. Click **Next**.

j. (DB2 Only) For Driver Class Name, enter com.ibm.db2.jcc.DB2Driver. For URL, enter the following:

```
jdbc:db2://<db_host>:<db_port>/<db_name>. For Properties, enter user=<db_
user_name>
```

k. Verify that the data source information is correct, and click **Test Configuration**. If the test fails, review your data source information and try again. When the test succeeds, click **Next**.

Figure 3–11 Data Source Information Verification



Create a New JDBC Data Source

Test Database Connection

Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool?

(Note that this driver class must be in the classpath of any server to which it is deployed.)

Driver Class Name: oracle.jdbc.OracleDrive

What is the URL of the database to connect to? The format of the URL varies by JDBC driver.

URL: jdbc:oracle:thin:@localh

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

Database User Name: csuser

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

(Note: for secure password management, enter the password in the Password field instead of the Properties field below)

Password: ••••••••••••••••••

Confirm Password: ••••••••••••••••••

What are the properties to pass to the JDBC driver when creating database connections?

Properties:

```
user=csuser
```

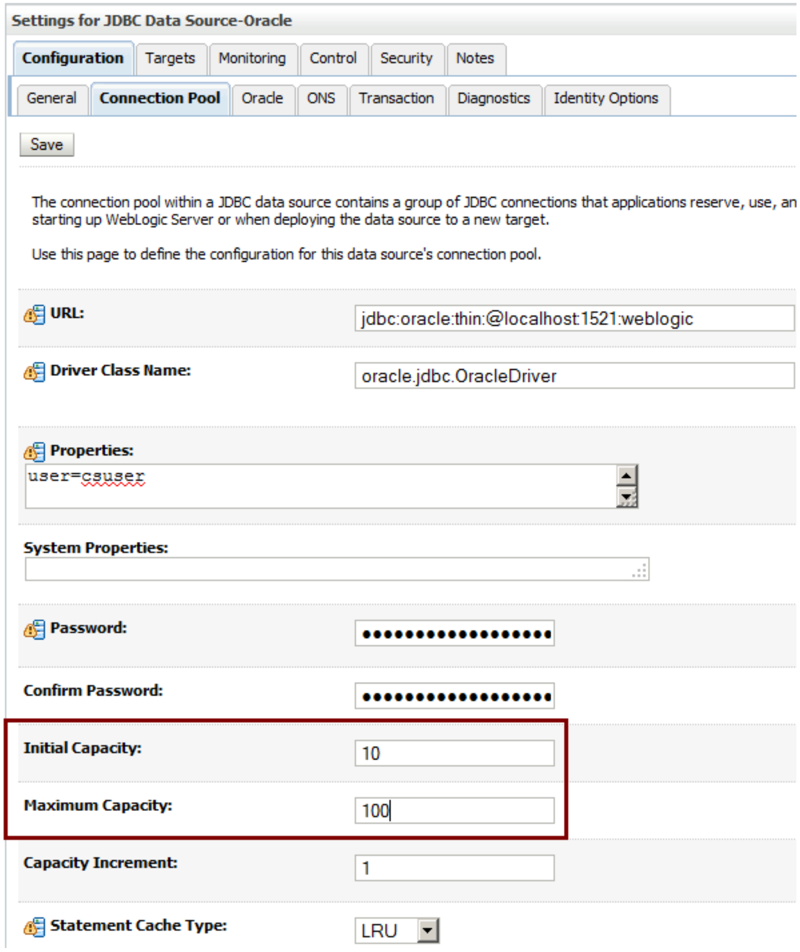
- **l.** Select the administration server as well as all managed servers that require access to the data source. In a cluster environment, select the cluster instead of selecting individual managed servers.
- **m.** Click **Finish**.

4. Configure the Connection Pool Size:

- **a.** Click the newly created data source.
- **b.** Click the **Connection Pool** tab.
- **c.** For Initial Capacity, enter `10` and for Maximum Capacity, enter `100`.

Note: These are recommended values that may be changed as your environment requires.

Figure 3–12 Configuration - Connection Pool



The connection pool within a JDBC data source contains a group of JDBC connections that applications reserve, use, an starting up WebLogic Server or when deploying the data source to a new target.

Use this page to define the configuration for this data source's connection pool.

URL: jdbc:oracle:thin:@localhost:1521:weblogic

Driver Class Name: oracle.jdbc.OracleDriver

Properties:

```
user=csuser
```

System Properties:

Password: ••••••••••••••••••

Confirm Password: ••••••••••••••••••

Initial Capacity: 10

Maximum Capacity: 100

Capacity Increment: 1

Statement Cache Type: LRU

d. Click **Save**.

3.2.2.5 Setting Java Options

This section explains how to set memory arguments as well as add Java options required by WebCenter Sites.

> **Note:** If a JRockit JDK will be used, you must create an optimization file containing the following:
>
> ```
> { match: [ "net/sf/ehcache/Cache.get*",
> "com/fatwire/cache/ehcache/EhCacheObjectCache._
> get*","com/fatwire/cache/ehcache/EhCacheObjectCache.get*",
> "net/sf/ehcache/Cache.isExpired*" ],
> hotspot : { enable : false },}
> ```

1. Stop the administration server.

2. Set Memory Arguments

   a. Open `<domain_home>/bin/setDomainEnv.sh` in a text editor and search for the following:

```
WLS_MEM_ARGS
```

b. Set the value for `WLS_MEM_ARGS_64BIT` or `WLS_MEM_ARGS_32BIT` for both Sun and JRockit. The recommended value is `-Xms2048m -Xmx16834m`.

c. Save the changes.

3. Add the required Java options to WebCenter Sites:

a. Open `<domain_home>/bin/startWebLogic.sh` in a text editor and search for the following line (located below the **Start Derby** section):

```
JAVA_OPTIONS="${SAVE_JAVA_OPTIONS}"
```

b. Modify the line by adding the following Java options:

- `-Dfile.encoding=UTF-8`

  This option specifies UTF-8 file encoding.

- `-Dnet.sf.ehcache.enableShutdownHook=true`

  This option enables the Ehcache shutdown hook.

- `-Djava.net.preferIPv4Stack=true`

  This option specifies the use of IPv4 addresses.

- `-Duser.timezone=UTC`

  This option is used to set the time zone for the JVM. It is recommended to set the value to `UTC` across all WebCenter Sites installations. However, you can set any preferable time zone as long as this value is the same for all WebCenter Sites installations.

- `-Djava.io.tmpdir=<path_to_sites_tmpdir>`

  This option specifies which temporary directory the JVM uses. The value should be the path to the directory created in Section 2.1, "Before You Begin."

- `(JRockit JDK only) -XX:+UnlockDiagnosticVMOptions`
  `-XX:OptFile=<path_to_optimization_file>`

  These options are required when using a JRockit JDK. They signal the JVM to use the optimization file that was created in Section 3.2.1, "Creating a WebLogic Domain."

Note: If `Shift-JIS` characters will be used, add the following option: `-Dcs.useJavaURLDecoder=false`

After making the above changes, the modified line should look as follows:

```
JAVA_OPTIONS="-Dfile.encoding=UTF-8
-Dnet.sf.ehcache.enableShutdownHook=true -Djava.net.preferIPv4Stack=true
-Duser.timezone=UTC -Djava.io.tmpdir=<path_to_sites_tmpdir> ${SAVE_JAVA_
OPTIONS}"
```

c. Save the changes.

3.2.2.6 Setting the Classpath

This section explains how to add paths required for WebCenter Sites to the application server classpath. To ensure a class is loaded first, it is added to the PRE_CLASSPATH variable.

1. Add the required paths to CLASSPATH for WebCenter Sites.

   a. Open <domain_home>/bin/startWebLogic.sh in a text editor and search for the following line (located below the **Start Derby** section):

   ```
   CLASSPATH="${SAVE_CLASSPATH}"
   ```

   b. Modify the line by adding the following path:

   ```
   <sites_install_dir>/bin
   ```

   This path contains the required WebCenter Sites and CAS configuration files. Without this path, the CAS web application will not start and the System Tools node in the Admin tab of the WebCenter Sites Admin interface will have reduced functionality.

   c. Save the changes.

2. Add the required paths to PRE_CLASSPATH for WebCenter Sites.

   a. Open wlserver_10.3/server/bin/startNodeManager.sh in a text editor and search for the following line:

   ```
   # Get PRE and POST environment
   if [ ! -z "${PRE_CLASSPATH}" ]; then
     CLASSPATH="PRE_CLASSPATH="<sites_deploy_
   home>/WEB-INF/lib/commons-lang-2.4.jar:${PRE_
   CLASSPATH}"${CLASSPATHSEP}${CLASSPATH}"
   ```

   b. Add the following line after it:

   ```
   PRE_CLASSPATH="<sites_deploy_home>/WEB-INF/lib/commons-lang-2.4.jar:${PRE_
   CLASSPATH}"
   ```

   This JAR is located in the lib directory of the exploded WebCenter Sites web application. For the WebCenter Sites Contributor interface to load properly, this JAR must be loaded to the classpath prior to the other versions of its classes.

   Note: This changes affects all domains.

   c. Save the changes.

3.2.2.7 Transfer Remote Managed Servers

This section explains how to transfer any remote managed servers that were created using the administration console, to the physical servers they will be running on.

1. Make sure that all servers in the domain are shutdown.
2. Create a template from the domain that was created on the primary server. Run the following command from <weblogic_home>/wlserver10.3/common/bin:

   ```
   ./pack.sh -managed=true -domain=<domain_home> -template=<template_creation_
   location>/<template_name>.jar -template_name="<template_name>"
   For example:
   ./pack.sh -managed=true -domain=/opt/Oracle/Middleware/user_
   ```

```
projects/domains/SitesDomain -template=/opt/Downloads/SitesDomainTemplate.jar
-template_name="SitesDomainTemplate"
```

3. For each remote server, complete the following steps:

   a. Copy the newly created template to the remote server.

      Note: The remote server should already have WebLogic server installed.

   b. Unpack the managed server and required domain files from the copied template.

      The domain will be created in the same location as the primary server.

   c. Run the following command from `<weblogic_home>/wlserver10.3/common/bin` directory:

      ```
      ./unpack.sh -domain=<domain_home> -template=<template_location>/<template
      name>.jar
      ```

      For example:

      ```
      ./unpack.sh -domain=/opt/Oracle/Middleware/user_
      projects/domains/SitesDomain -template=/opt/Downloads/SitesDomain.jar
      ```

   d. Copy the `<domain_home>/bin/startWebLogic.sh` and `<domain_home>/bin/setDomainEnv.sh` files from the primary server to the `<domain_home>/bin` directory of the remote server. The modified files are not included in the template.

   e. Modify the files and make any changes to reflect differences in the paths or options added in Section 3.2.2.5, "Setting Java Options" and Section 3.2.2.6, "Setting the Classpath."

   f. Save the changes.

3.3 Step 2: Integrating Oracle WebLogic Server with a Supported Web Server

This chapter provides guidelines for integrating WebLogic Server with your choice of supported web servers.

This chapter contains the following sections:

- Section 3.3.1, "Configuring the Oracle HTTP Server"
- Section 3.3.2, "Configuring the Apache HTTP Server"
- Section 3.3.3, "Configuring the IIS Web Server"

3.3.1 Configuring the Oracle HTTP Server

Follow these steps to configure Oracle HTTP Server with the `mod_wl_ohs` plugin for use with WebLogic.

1. Open `<ohs_instance>/config/OHS/<server_name>/mod_wl_ohs.conf` in a text editor and add the following before the commented Location tag:

   Single Server Environment

```
<IfModule mod_weblogic.c>
   WebLogicHost {weblogic_host}
   WebLogicPort {weblogic_port}
</IfModule>

<Location /{sites_context_root}>
  SetHandler weblogic-handler
</Location>

<Location /cas>
  SetHandler weblogic-handler
</Location>
```

Multi Server Environment

```
<IfModule mod_weblogic.c>
    WebLogicCluster {weblogic_host1}:{weblogic_port1},{weblogic_
host2}:{weblogic_port2}
</IfModule>

<Location /{sites_context_root}>
  SetHandler weblogic-handler
</Location>

<Location /cas>
  SetHandler weblogic-handler
</Location>
```

2. Save the changes.
3. Restart the web server.

3.3.2 Configuring the Apache HTTP Server

This section explains how to configure Apache 2.2.x HTTP server with the `mod_wl` plugin to use this plugin with WebLogic.

To configure the mod_wl plug-in

1. Ensure that `mod_so.c` is enabled. Run `<apache_home>/bin/apachectl -l`. If `mod_so.c` is not in the list that is printed, you must rebuild your Apache with the `-enable-module=so` option.
2. Copy the `mod_wl_22.so` file from `<weblogic_home>/wlserver_10.3/server/plugin/<os_type>/<os_version>` to `<apache_home>/modules`.
3. Open `<apache_home>/config/httpd.conf` in a text editor and add the following at the end of the file:

   Single Server Environment

```
<IfModule mod_weblogic.c>
  WebLogicHost {weblogic_host}
  WebLogicPort {weblogic_port}
</IfModule>

<Location /{sites_context_root}>
  SetHandler weblogic-handler
</Location>

<Location /cas>
  SetHandler weblogic-handler
```

```
</Location>
```

Multi Server Environment

```
LoadModule weblogic_module modules/mod_wl_22.so
```

```
<IfModule mod_weblogic.c>
  WebLogicCluster {weblogic_host1}:{weblogic_port1},{weblogic_host2}:{weblogic_
port2}
</IfModule>

<Location /{sites_context_root}>
  SetHandler weblogic-handler
</Location>

<Location /cas>
  SetHandler weblogic-handler
</Location>
```

4. Save the changes.
5. Verify the syntax of the `httpd.conf` file with the following command:

   ```
   <apache_home>/bin/apachectl -t
   ```

6. Restart the web server.

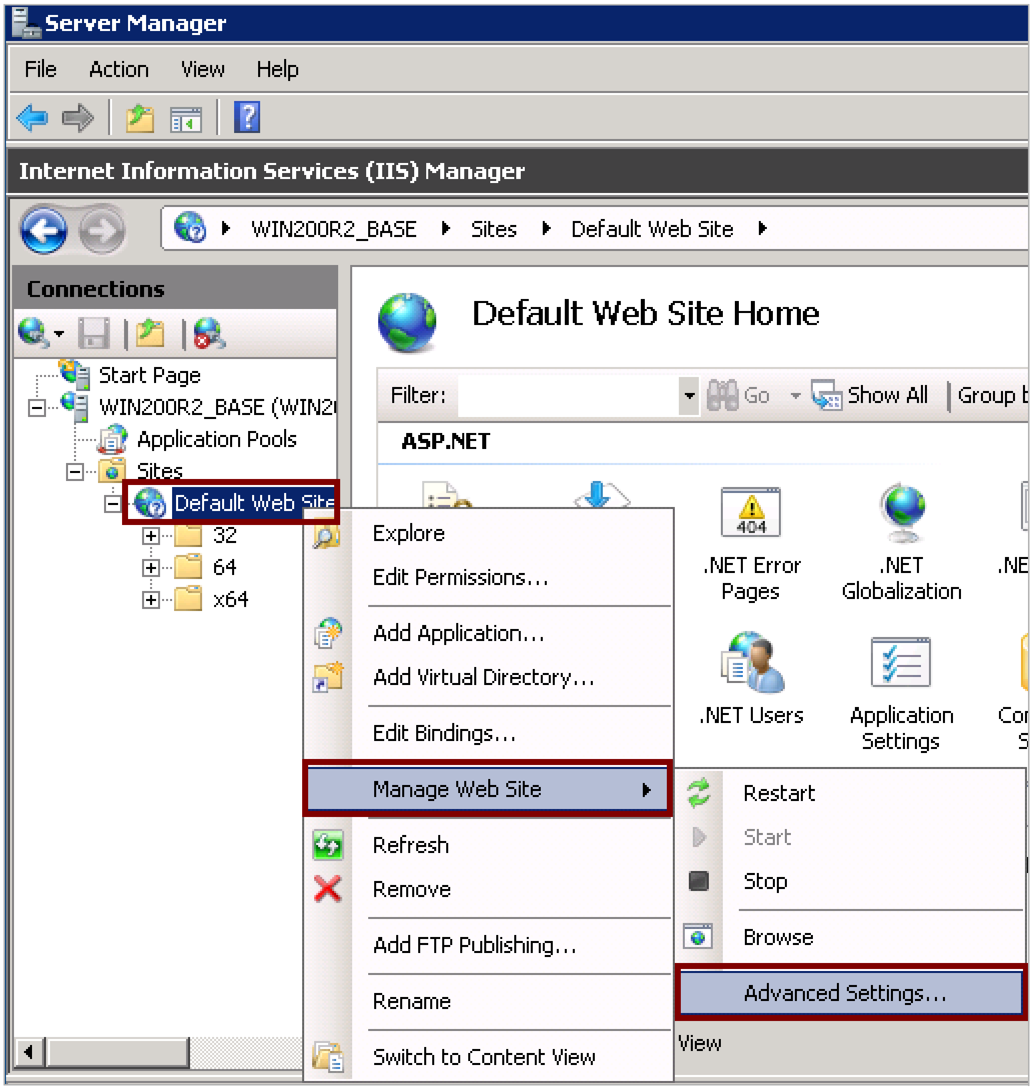
3.3.3 Configuring the IIS Web Server

This section explains how to configure IIS versions 7.0 and later with the IIS plugin to use this plugin with the WebLogic server.

To configure the IIS plug-in for IIS 7.0 and later versions

1. Go to the `<wl_home>/wlserver_10.3/server/plugin/` directory and copy the correct OS version and type plugin folder to the server where you installed IIS (referred to as the `<IIS_plugin_dir>` directory). For example: `C:\inetpub\win\x64`.
2. Access the IIS Manager. In the Connections navigation pane, expand **root server**, then expand **Sites**.
   a. Right-click the Default Web Site node and select **Manage Web Site** then select **Advanced Settings** (Figure 3–13).

Figure 3–13 Connections - Manage Web Site



b. In the Advanced Settings window, change the value of the **Physical Path** field to the `<IIS_plugin_dir>` directory (from step 1 on page 3-23). For example, `C:\inetpub\win\x64`. Then, click **OK**.

3. In the Default Web Site Home screen, double-click **ISAPI Filters**.

   a. Right-click anywhere in the ISAPI Filters screen and click **Add** from the context menu.

      The Add ISAPI Filter window opens.

   b. Populate the following fields with the appropriate information for your system. For example:

      – Filter name: `wlforward`

      – Executable: `C:\inetpub\win\x64\iisforward.dll`

   c. Click **OK**.

4. Click **Default Web Site** to return to the Default Web Site Home screen. In the Default Web Site Home screen, double-click **Handler Mappings**.

a. In the Handler Mappings screen, right-click anywhere and select **Add Script Map** from the context menu.

The Add Script Map window opens.

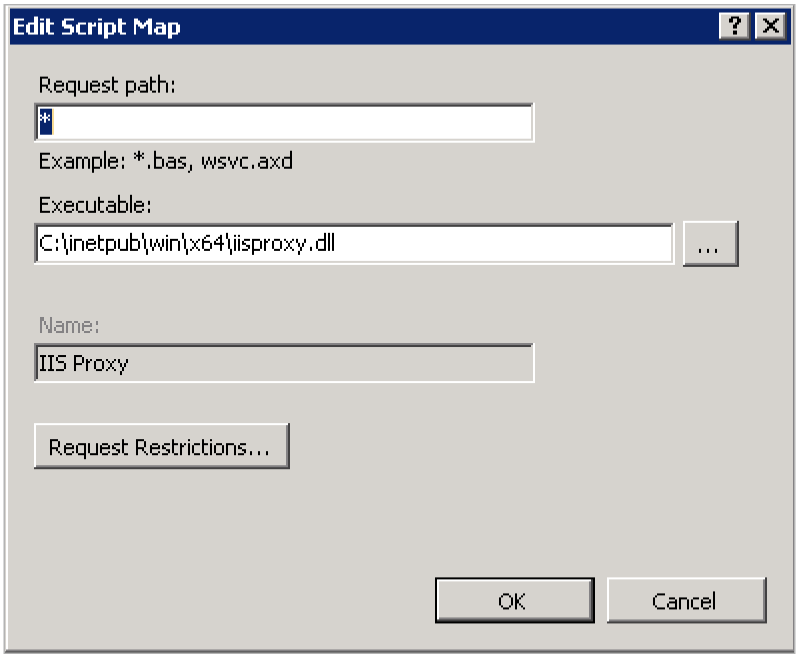
b. Fill in the following fields with the appropriate information for your system (Figure 3–14). For example:

– Requested Path: `*`

– Executable: `C:\inetpub\win\x64\iisproxy.dll`

– Name: `IISProxy`

Figure 3–14 Edit Script Map Dialog Box



Edit Script Map

Request path: *

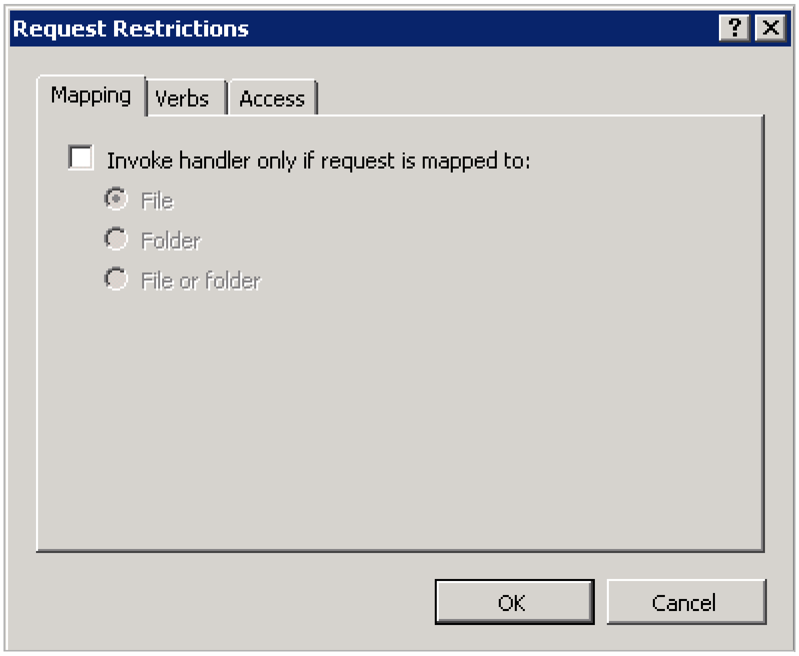
Example: *.bas, wsvc.axd

Executable: C:\inetpub\win\x64\iisproxy.dll

Name: IIS Proxy

c. Click **Requested Restrictions**. In the Request Restrictions window, deselect the **Invoker handler mapping** option (if checked), as shown in Figure 3–15.

Figure 3–15 Request Restrictions Dialog Box



☐ Invoke handler only if request is mapped to:

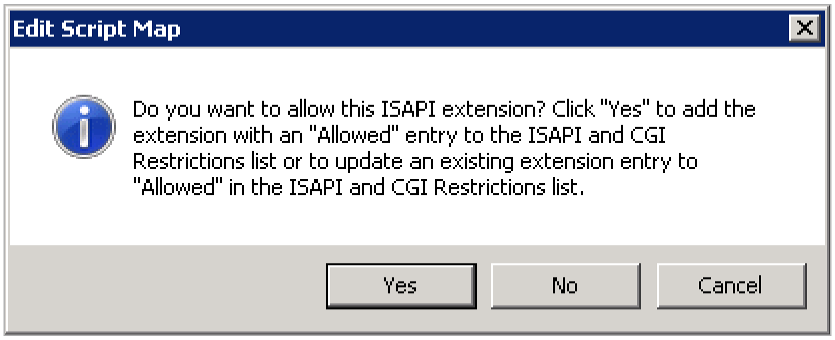
● File

○ Folder

○ File or folder

d. Click **Yes** (Figure 3–16).

Figure 3–16 Edit Script Map Dialog Box



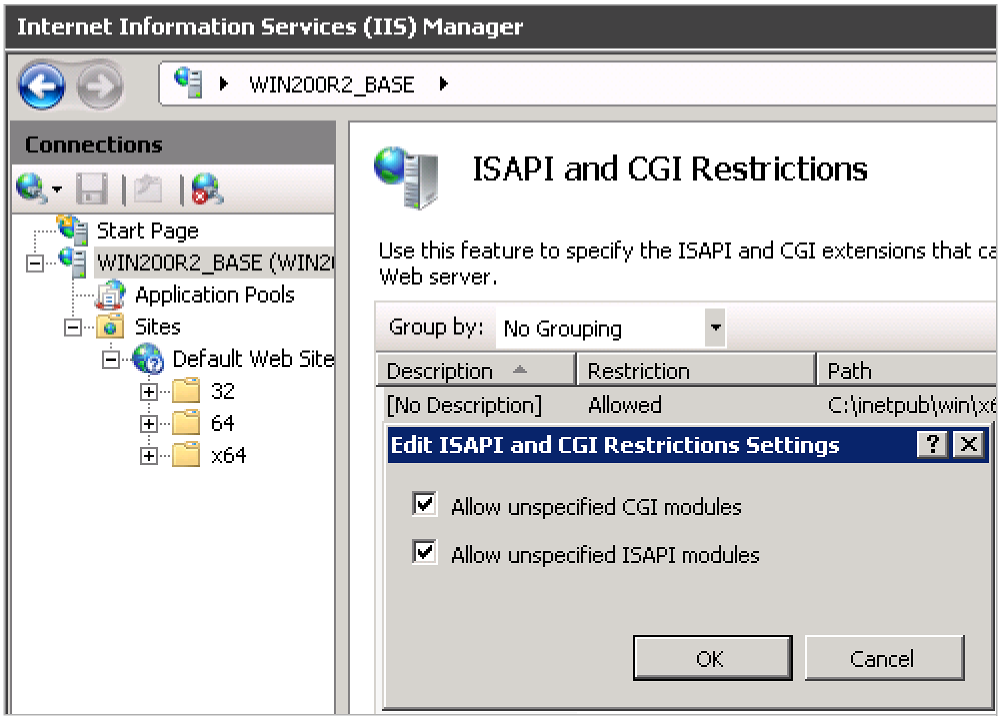
Do you want to allow this ISAPI extension? Click "Yes" to add the extension with an "Allowed" entry to the ISAPI and CGI Restrictions list or to update an existing extension entry to "Allowed" in the ISAPI and CGI Restrictions list.

5. Click the **root server** node to access the IIS Home screen, then double-click **ISAPI and CGI Restrictions**.

   a. In the ISAPI and CGI Restrictions screen, right-click anywhere and select **Edit Feature Settings** from the context menu.

   b. In the Edit ISAPI and CGI Restrictions Settings window, check both the **Allow unspecified CGI modules** option and the **Allow unspecified ISAPI modules** option (Figure 3–17).

Figure 3–17 Edit ISAPI and CGI Restrictions Settings Dialog Box



Internet Information Services (IIS) Manager

WIN200R2_BASE

ISAPI and CGI Restrictions

Use this feature to specify the ISAPI and CGI extensions that ca
Web server.

Group by: No Grouping

Description	Restriction	Path
[No Description]	Allowed	C:\inetpub\win\x6

Edit ISAPI and CGI Restrictions Settings

☑ Allow unspecified CGI modules

☑ Allow unspecified ISAPI modules

c. Click **OK**.

6. In the **Physical Path** folder (`<IIS_plugin_dir>` directory, from step 1 on page 3-23), create a file named `iisproxy.ini` and populate it with the appropriate information for your system.

   - For a non-clustered environment, use the following example:

     ```
     WebLogicHost=<hostname>
     WebLogicPort=<port>
     ConnectRetrySecs=5
     WLForwardPath=/<cs_context_root>,/<cas_context_root>
     ```

   - For a clustered environment, use the following example:

     ```
     WebLogicCluster=<member1_hostname>:<member1_port>,
     <member2_hostname>:<member2_port>,<membern_hostname>:
     <membern_port>
     ConnectRetrySecs=5
     ConnectTimeoutSecs=25
     WLForwardPath=/<cas_context_root>,/<cas_context_root>
     ```

     For detailed information about these parameters, refer to the vendor documentation.

     Note: Only include `/cas` if both Sites and CAS are running on the same server, if CAS is located on a separate server then you will need new entries for it that are unique to that environment.

7. Start your IIS Server and Default Web Site.

8. You are now ready to install WebCenter Sites. For instructions, continue to Chapter 6, "Procedures for Installing WebCenter Sites."

4

Preparing to Install WebCenter Sites on Apache Tomcat Application Server

This chapter contains information about configuring Tomcat to support your WebCenter Sites web application.

This chapter contains the following sections:

- Section 4.1, "Quick Reference for Setting Up Tomcat Application Server"
- Section 4.2, "Step 1: Configuring Tomcat Application Server"
- Section 4.3, "Step 2: Integrating Tomcat Application Server with a Supported Web Server"

4.1 Quick Reference for Setting Up Tomcat Application Server

This section contains the following topics:

- Section 4.1.1, "Summary of Steps"
- Section 4.1.2, "Paths and Directories Used in This Chapter"
- Section 4.1.3, "Start/Stop Commands for Tomcat"

4.1.1 Summary of Steps

You will complete the steps below as necessary for your requirements for each environment: development, content management, and delivery.

Step 1: Configuring Tomcat Application Server

1. Modify the default Tomcat ports as shown in Section 4.2.1, "Modifying Tomcat Server Ports."
2. Create and configure a data source as shown in Section 4.2.2, "Creating and Configuring a Data Source."
3. Set Java options as described in Section 4.2.3, "Setting Java Options."
4. Set the classpath as described in Section 4.2.4, "Setting the Classpath."
5. Set the URI encoding character set, as described in Section 4.2.5, "Configuring URI Encoding."
6. If the server is a part of a multi-server environment, set the `jvmRoute` attribute as described in Section 4.2.7, "Configuring a Tomcat Server for Load Balancing."

7. If the Tomcat server is part of a cluster environment, configure the server for clustering as described in Section 4.2.8, "Configuring a Tomcat Server for Clustering."
8. If the server will be running over SSL, configure the Tomcat server for SSL as described in Section 4.2.9, "Configuring a Tomcat Server for SSL."

Step 2: Integrating Tomcat Application Server with a Supported Web Server

This step is required if:

- You are creating a clustered environment that will be load balanced through Apache Web Server.
- You wish to place an apache WebServer in front of Tomcat.

Steps for integrating Tomcat with Apache Web Server are given in Section 4.3, "Step 2: Integrating Tomcat Application Server with a Supported Web Server."

4.1.2 Paths and Directories Used in This Chapter

Table 4–1 lists the paths and directories used in this chapter.

Table 4–1 Paths and Directories Used in Configuring Tomcat Application Server

Name	Description
`<sites_install_dir>`	Path to the WebCenter Sites installation directory.
`<tomcat_home>`	Path to the Tomcat installation directory.
`<shared_dir>`	Path to the shared folder on the given system. The path includes the name of the shared folder.

4.1.3 Start/Stop Commands for Tomcat

This section lists commands for starting and stopping Tomcat instances.

> **Note:** All commands require that `CATALINA_HOME` and `JAVA_HOME` are set to the proper directories.

To start the Tomcat instance

Run the following command:

- In UNIX:

```
<tomcat_home>/bin/startup.sh
```

- In Windows:

```
<tomcat_home>/bin/startup.bat
```

To stop the Tomcat instance

Run the following command:

- In UNIX:

```
<tomcat_home>/bin/shutdown.sh -force
```

- In Windows:

```
<tomcat_home>/bin/shutdown.bat
```

4.2 Step 1: Configuring Tomcat Application Server

In this section, you will complete the following steps:

- Section 4.2.1, "Modifying Tomcat Server Ports"
- Section 4.2.2, "Creating and Configuring a Data Source"
- Section 4.2.3, "Setting Java Options"
- Section 4.2.4, "Setting the Classpath"
- Section 4.2.5, "Configuring URI Encoding"
- Section 4.2.7, "Configuring a Tomcat Server for Load Balancing"
- Section 4.2.8, "Configuring a Tomcat Server for Clustering"
- Section 4.2.9, "Configuring a Tomcat Server for SSL"

> **Note:** Procedures in this section use UNIX commands.

4.2.1 Modifying Tomcat Server Ports

This section explains how to modify the default ports used by Tomcat. In order to run more than one Tomcat server on the same physical server, these ports must be changed to avoid port conflicts.

1. Shutdown the Tomcat server.
2. Open the `<tomcat_home>/conf/server.xml` file in a text editor and modify the following ports:
   - Shutdown port (default `8005`)

     This port is used when shutting down the Tomcat server.
   - HTTP port (default `8080`)

     This is the main port from which web applications will be accessible.
   - Redirect port (default `8443`)

     This port needs to be modified in both the HTTP and AJP connectors.
   - AJP port (default `8009`)

     This port can be used for communicating with an Apache web server.
   - (SSL only) SSL HTTP port (default `8443`)

     If you will be using SSL, change this port to the same value used in step n.
3. Save the changes.

4.2.2 Creating and Configuring a Data Source

1. Choose the data source information that corresponds to the database you will be using and place the required JAR files in the `<tomcat_home>/lib` directory. Data source information in Table 4–2 will be used to create the data source in the next step.

Table 4–2 Data Source Information

Database Driver	Parameters	Values
JTDS (third-party driver)	■ `DriverClass` ■ Required `.jar` files ■ URL	■ `net.sourceforge.jtds.jdbcx.JtdsDataSource` ■ `jtds-1.2.5 jar` ■ `jdbc:jtds:sqlserver://<server>:<dbport>/<dbname>` Example: `jdbc:jtds:sqlserver://127.0.0.1:1433/contentDB`
DB2	■ `DriverClass` ■ Required `.jar` files ■ URL	■ `com.ibm.db2.jcc.DB2Driver` ■ `db2jcc.jar, db2cc_license_cu.jar` **Note**: The `db2jcc4.jar` file is not supported. Installation will fail if the `db2jcc4.jar` file is used. ■ `jdbc:db2://<hostname>:<dbport>/<dbname>` Example: `jdbc:db2://127.0.0.1:50001/contentDB`
Oracle Thin driver	■ `DriverClass` ■ Required `.jar` files ■ URL	■ `oracle.jdbc.driver.OracleDriver` ■ `ojdbc6.jar` ■ `jdbc:oracle:thin:@//<hostname>:1521/<dbname>` Example: `jdbc:oracle:thin:@//127.0.0.1:1521/contentDB`

2. Open the `<tomcat_home>/conf/server.xml` file in a text editor and add the following XML code inside the `Host` tag, replacing any parameterized values:

```
<Context path="/<sites_context_root>" docBase="<sites_context_root>"
reloadable="true" crossContext="true">
          <Resource name="<jndi_name>" auth="Container"
type="javax.sql.DataSource"
                    maxActive="100" maxIdle="30"
                    username="<db_user_name>"
                    password="<db_user_password>"
                    driverClassName="<db_driver_class>"
                    url="<db_url>"/>
        </Context>
```

Sample data source for an Oracle Database:

```
<Context path="/cs" docBase="cs" reloadable="true" crossContext="true">
          <Resource name="csDataSource" auth="Container"
type="javax.sql.DataSource"
                    maxActive="100" maxIdle="30"
                    username="csuser"
                    password="password"
                    driverClassName="oracle.jdbc.driver.OracleDriver"
                    url="jdbc:oracle:thin:@//127.0.0.1:1521/contentDB"/>
        </Context>
```

Note: In the example above the maximum allowed database connections is set to `100`. While this is the recommended value, it may be changed depending on environment requirements.

3. Save the changes.

4.2.3 Setting Java Options

This section explains how to set memory arguments as well as add Java options required by WebCenter Sites.

1. Open `<tomcat_home>/bin/setenv.sh` in a text editor and add the line `CATALINA_OPTS="<java_opts>"` where `<java_opts>` is a space-separated list of the following options:

   - `-Xms2048m -Xmx16834m`

     These options set the minimum and maximum memory heap sizes. These are recommended sizes, but may be changed depending on environment requirements.

   - `-XX:MaxPermSize=192m`

     This option sets the maximum size of the permanent generation space. This is the recommended size, but may be changed depending on environment requirements.

   - `-Dfile.encoding=UTF-8`

     This option specifies UTF-8 file encoding.

   - `-Dnet.sf.ehcache.enableShutdownHook=true`

     This option enables the `Ehcache` shutdown hook.

   - `-Djava.net.preferIPv4Stack=true`

     This option specifies the use of IPv4 addresses.

   - `-Duser.timezone=UTC`

     This option is used to set the time zone for the JVM. It is recommended to set the value to `UTC` across all WebCenter Sites installations. However, you can set any preferable time zone as long as this value is the same for all WebCenter Sites installations. So, the complete `CATALINA_OPTS` should look like:

     ```
     CATALINA_OPTS="-Xms2048m -Xmx16834m -XX:MaxPermSize=192m
     -Dfile.encoding=UTF-8
     -Dnet.sf.ehcache.enableShutdownHook=true -Djava.net.preferIPv4Stack=true
     -Duser.timezone=UTC"
     ```

   Note: If `Shift-JIS` characters will be used, add the following option: `-Dcs.useJavaURLDecoder=false`

2. Save the changes.

4.2.4 Setting the Classpath

This section explains how to add paths required by WebCenter Sites to the application server classpath.

1. Open `<tomcat_home>/bin/setenv.sh` in a text editor and add the following line: `CLASSPATH="<paths>"` where `<paths>` is a colon (:) separated list of the following paths:

   a. `<sites_install_dir>/bin`

This path contains the required WebCenter Sites and CAS configuration files. Without this file, the CAS web application will not start and the **System Tools** node in the **Admin** tab of the WebCenter Sites Admin interface will have reduced functionality.

b. `$JAVA_HOME/lib/tools.jar`

This JAR file is located in the JDK.

2. Save the changes.

4.2.5 Configuring URI Encoding

This section explains how to set the character set for URI encoding to UTF-8.

1. Open `<tomcat_home>/conf/server.xml` in a text editor and add the `URIEncoding` attribute to the `HTTP Connector` tag as follows:

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" URIEncoding="UTF-8" />
```

2. Save the changes.

4.2.6 Configuring AJP Connectors

This section explains how to set encoding configuration for one or both HTTP and AJP connectors on the application server.

1. Ensure all the language and file encoding configuration parameters are set to be UTF-8 throughout the entire infrastructure for multilingual and special character recognition.
2. This includes the db, system variables, futuretense.ini file and the connectors as defined in the application server -including the AJP connector as needed, described in the Oracle Fusion Middleware Online Documentation Library 11g Release 1 (11.1.1.8) for WebCenter Sites installation and administration guides.
3. The above circumstance (Tomcat also using Apache webserver) was resolved by the following:

   Add `URIEncoding="UTF-8"` in the HTTP and AJP Connectors in `$CATALINA_BASE/conf/server.xml.`

4.2.7 Configuring a Tomcat Server for Load Balancing

This section explains how to configure a Tomcat server so that it may be a part of a load balancing environment.

1. Open `<tomcat_home>/conf/server.xml` in a text editor and add the `jvmRoute` attribute to the `Engine` tag as follows:

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
```

2. Modify the value of `jvmRoute` so that it is unique from all other Tomcat servers that will be load balanced.
3. Save the changes.

4.2.8 Configuring a Tomcat Server for Clustering

This section explains how to configure a Tomcat server to be part of a cluster.

1. Open `<tomcat_home>/conf/server.xml` in a text editor and add the following XML code as the first thing inside the `Engine` tag:

```
<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"
                 channelSendOptions="8">
          <Manager className="org.apache.catalina.ha.session.DeltaManager"
                   expireSessionsOnShutdown="false"
                   notifyListenersOnReplication="true"/>
          <Channel className="org.apache.catalina.tribes.group.GroupChannel">
            <Membership
className="org.apache.catalina.tribes.membership.McastService"
                        address="228.0.0.4"
                        port="45564"
                        frequency="500"
                        dropTime="3000"/>
            <Receiver
className="org.apache.catalina.tribes.transport.nio.NioReceiver"
                      address="auto"
                      port="4000"
                      autoBind="100"
                      selectorTimeout="5000"
                      maxThreads="6"/>
            <Sender
className="org.apache.catalina.tribes.transport.ReplicationTransmitter">
              <Transport
className="org.apache.catalina.tribes.transport.nio.PooledParallelSender"/>
            </Sender>
            <Interceptor
className="org.apache.catalina.tribes.group.interceptors.TcpFailureDetector"/>
            <Interceptor
className="org.apache.catalina.tribes.group.interceptors.MessageDispatch15Inter
ceptor"/>
          </Channel>
          <Valve className="org.apache.catalina.ha.tcp.ReplicationValve"
                 filter=""/>
          <Valve
className="org.apache.catalina.ha.session.JvmRouteBinderValve"/>
          <Deployer className="org.apache.catalina.ha.deploy.FarmWarDeployer"
                    tempDir="/tmp/war-temp/"
                    deployDir="/tmp/war-deploy/"
                    watchDir="/tmp/war-listen/"
                    watchEnabled="false"/>
          <ClusterListener
className="org.apache.catalina.ha.session.JvmRouteSessionIDBinderListener"/>
          <ClusterListener
className="org.apache.catalina.ha.session.ClusterSessionListener"/>
        </Cluster>
```

2. Modify the Membership address or port as necessary:

   - Each cluster must have a unique Membership address/port combination to avoid message conflicts.
   - All members of a cluster must have the same values for Membership address and port.

3. Modify the Receiver port or `autoBind` as necessary.

   The receiver will automatically avoid port conflicts between port and port plus `autoBind` by binding to a free port. Modify either of these values as necessary to restrict the range of ports the receiver will use.

4. Save the changes.

4.2.9 Configuring a Tomcat Server for SSL

1. Generate a certificate:

   ```
   keytool -genkey -alias tomcat -keyalg RSA -keystore
   <tomcat_home>/keys/.keystore
   ```

   a. Set the `-keystore` parameter to the location where you want the generated keys to be stored.

   b. When prompted for the keystore password, you may use your own password, or the default `changeit`. If a custom password is entered, a value in the `server.xml` file will need to be changed (as explained in step 2).

   c. Enter your first and last name, name of organization, city, state, and country code when prompted, then click **Yes** to confirm.

   d. At the next prompt, "Enter key password for `<tomcat>` (RETURN if same as keystore password)," press **Enter**, as the key password **must** be the same as the keystore password.

2. Edit `<tomcat_home>/conf/server.xml`, uncomment the SSL section, and add the following attributes to the SSL connector:

   - `keystoreType="JKS"`

     This attribute is set to JKS for Java Keystore (the format produced by Java's keytool).

   - `keystoreFile="<tomcat_home>/keys/.keystore"`

     This attribute is set to the path where the `.keystore` file was created and the same path used in step 1.

   - `keystorePass="<new_password>"`

     This attribute is needed only if the keystore password used in step 1 is not `changeit`, and should be set to the custom password used.

3. Save the changes.

4. After starting the server, point your browser to `https://<tomcat_host>:<ssl_port>`.

   If your system works correctly, you will be prompted to accept the certificate. When you accept the certificate, the Tomcat index page is displayed.

4.3 Step 2: Integrating Tomcat Application Server with a Supported Web Server

This section provides guidelines for integrating Tomcat Server with your choice of supported web servers.

4.3.1 Configuring Apache HTTP Server

This section explains how to configure Apache 2.2.x or Apache 2.4.x HTTP server with the `mod_proxy_ajp` plugin to use this plugin with Tomcat.

1. Ensure that mod_so.c is enabled. Run <apache_home>/bin/apachectl -l. If mod_so.c is not in the list that is printed, you must rebuild your Apache with the -enable-module=so option.
2. Open <apache_home>/config/httpd.conf in a text editor and add the following at the end of the file:

   Single Server Environment

   ```
   ProxyPass /cas ajp://<tomcat_server_host>:<ajp_port>/cas
   ProxyPassReverse /cas ajp://<tomcat_server_host>:<ajp_port>/cas
   ProxyPass /<sites_context_root> ajp://<tomcat_server_host>:<ajp_port>/<sites_
   context_root>
   ProxyPassReverse /<sites_context_root> ajp://<tomcat_server_host>:<ajp_
   port>/<sites_context_root>
   ```

   Multi Server Environment

   ```
   <Location /balancer-manager>
       SetHandler balancer-manager
       Order Deny,Allow
       Deny from none
       Allow from all
   </Location>

   <Proxy balancer://cluster>
       BalancerMember ajp://<tomcat_server_host1>:<ajp_port1> loadfactor=1
   route=<jvm_route1>
       BalancerMember ajp://<tomcat_server_host2>:<ajp_port2> loadfactor=1
   route=<jvm_route2>
       ProxySet stickysession=JSESSIONID
   </Proxy>

   ProxyPass /cas balancer://cluster/cas
   ProxyPassReverse /cas balancer://cluster/cas
   ProxyPass /<sites_context_root> balancer://cluster/<sites_context_root>
   ProxyPassReverse /<sites_context_root> balancer://cluster/<sites_context_root>
   ```

   > **Note:** In this example, the balancer-manager allows all access. For increased security you may want to modify the values of "Deny from" and "Allow from" using the Apache documentation as reference.
   >
   > If you will be installing the AVI Sports sample site, add timeout=180 at the end of each BalancerMember line.

3. Save the changes.
4. Verify the syntax of the httpd.conf file with the following command:

   ```
   <apache_home>/bin/apachectl -t
   ```

5. Restart the web server.

5

Preparing to Install WebCenter Sites on IBM WebSphere Application Server

The chapter shows you how to configure WebSphere Application Server (WAS) for WebCenter Sites.

This chapter contains the following sections:

- Section 5.1, "Quick Reference for Setting Up WebSphere Application Server"
- Section 5.2, "Step 1: Configuring WebSphere Application Server"
- Section 5.3, "Step 2: Integrating WebSphere Application Server with a Supported Web Server"

> **Note:** Before starting the procedures in this chapter, ensure you have completed pre-requisite steps described in Chapter 2, "Prerequisites for Installing WebCenter Sites."

5.1 Quick Reference for Setting Up WebSphere Application Server

This section contains the following topics:

- Section 5.1.1, "Summary of Steps"
- Section 5.1.2, "Paths and Directories Used in Configuring WebSphere Application Server"
- Section 5.1.3, "Start/Stop Commands for WebSphere Application Server"

5.1.1 Summary of Steps

Complete the steps below for each development, content management, and delivery environment on which WebCenter Sites will be installed.

Step 1: Configuring WebSphere Application Server

1. Create the deployment manager and application server profiles as described in Section 5.2.1, "Create Required Profiles."
2. If you are configuring a cluster environment, create a cluster as described in Section 5.2.2, "Create a Cluster."
3. If you are configuring a non-cluster environment, create application servers as described in Section 5.2.3, "Create Servers."
4. Configure application servers as described in Section 5.2.4, "Configure Servers."

This step requires you to:

a. Configure web container properties for application servers as described in Section 5.2.4.1, "Configuring Web Container Properties."

b. Configure session management for application servers as described in Section 5.2.4.2, "Configuring Session Management."

c. Set the classpath and Java options as described in Section 5.2.4.3, "Setting the Classpath and Java Options."

5. Add host aliases to the virtual host for the new servers as described in Section 5.2.5, "Create Host Aliases for the Virtual Host."

6. Configure the WAS environment to connect to the WebCenter Sites database as described in Section 5.2.6, "Configuring a WAS Environment for Database Communications."

   This step required you to:

   a. Create J2C authentication data as described in Section 5.2.6.1, "Creating J2C Authentication Data."

   b. Create a JDBC provider as described in Section 5.2.6.2, "Creating a JDBC Provider."

   c. Create a data source as described in Section 5.2.6.3, "Creating a Data Source."

Step 2: Integrating WebSphere Application Server with a Supported Web Server

If your environment required a web server, you have the option to integrate WebSphere application servers with IBM HTTP Server or an Apache web server. For information, see Section 5.3, "Step 2: Integrating WebSphere Application Server with a Supported Web Server."

5.1.2 Paths and Directories Used in Configuring WebSphere Application Server

Table 5–1 lists paths and directories that are used in configuring WebSphere application server.

Table 5–1 Paths and Directories Used in Configuring WebSphere Application Server

Name	Description
`<sites_install_dir>`	Path to the directory where WebCenter Sites is installed. The path includes the name of the directory.
`<shared_dir>`	Path to the WebCenter Sites shared file system directory. The path includes the name of the shared directory.
`<WAS_home>`	Path to the directory where WebSphere Application Server is installed. The path includes the name of the directory.
`<dmgr_host>`	The host name or IP address of the Deployment Manager host.
`<dmgr_port>`	The port number on which the Deployment Manager console is listening for connections.
`<dmgr_profile>`	The name of the Deployment Manager profile.
`<dmgr_soap_port>`	The number of the Simple Object Access Protocol port of the Deployment Manager.
`<server_name>`	The name of the WAS server.
`<appsrv_profile>`	The name of the application server profile.

Table 5–1 (Cont.) Paths and Directories Used in Configuring WebSphere Application

Name	Description
`<appsrv_node>`	The name of the application server node.
`<ibm_http_home>`	Path to the directory where IBM HTTP Server is installed. The path includes the name of the directory.
`<plugin_root>`	The path to the plug-in directory of the web server. The path includes the name of the directory.

5.1.3 Start/Stop Commands for WebSphere Application Server

This section lists the commands for starting and stopping WAS components.

This section contains the following topics:

- Section 5.1.3.1, "Deployment Manager"
- Section 5.1.3.2, "Node Agent"
- Section 5.1.3.3, "Application Server"

5.1.3.1 Deployment Manager

Note: The default Deployment Manager profile name is `Dmgr01`.

Start and stop commands are the following:

- To start:
  - On Windows:

    ```
    <WAS_home>\bin\startManager.bat -profileName <dmgr_profile>
    ```

  - On UNIX:

    ```
    <WAS_home>/bin/startManager.sh -profileName <dmgr_profile>
    ```

- To stop:
  - On Windows:

    ```
    <WAS_home>\bin\stopManager.bat -profileName <dmgr_profile>
    ```

  - On UNIX:

    ```
    <WAS_home>/bin/stopManager.sh -profileName <dmgr_profile>
    ```

5.1.3.2 Node Agent

Note: The default name of the first application server profile created is `AppSrv01`.

Start and stop commands are the following:

- To start:
  - On Windows

```
<WAS_home>\bin\startNode.bat -profileName <appsrv_profile>
```

  - On UNIX

```
<WAS_home>/bin/startNode.sh -profileName <appsrv_profile>
```

- To stop:
  - On Windows:

```
<WAS_home>\bin\stopNode.bat -profileName <appsrv_profile>
```

  - On UNIX:

```
<WAS_home>/bin/stopNode.sh -profileName <appsrv_profile>
```

5.1.3.3 Application Server

Note: The default server name is `server1`.

The default name of the first application server profile created is `AppSrv01`.

Start and stop commands are the following:

- To start:
  - On Windows:

```
<WAS_home>\bin\startServer.bat <server_name> -profileName <appsrv_profile>
```

  - On UNIX:

```
<WAS_home>/bin/startServer.sh <server_name> -profileName <appsrv_profile>
```

- To stop:
  - On Windows:

```
<WAS_home>\bin\stopServer.bat <server_name> -profileName <appsrv_profile>
```

  - On UNIX:

```
<WAS_home>/bin/stopServer.sh <server_name> -profileName <appsrv_profile>
```

5.2 Step 1: Configuring WebSphere Application Server

This section shows you how to create a WAS instance using the command line.

Note: On Windows, the names of the command-line tools used in this section end with `.bat` instead of `.sh`. Remember to make the necessary substitution when executing the commands on a Windows system.

This section contains the following steps:

- Section 5.2.1, "Create Required Profiles"

- Section 5.2.2, "Create a Cluster"
- Section 5.2.3, "Create Servers"
- Section 5.2.4, "Configure Servers"
- Section 5.2.5, "Create Host Aliases for the Virtual Host"
- Section 5.2.6, "Configuring a WAS Environment for Database Communications"

5.2.1 Create Required Profiles

Following the steps given in this section, you will be creating the deployment manager and application server profiles as required by your environment. In a multi-server non-clustered environment, you may use a single deployment manager to administer all local and remote servers from a single administration console, or use a deployment manager for each physical server. In a clustered environment, use a deployment manager on the primary cluster member only.

Note: Make sure the host name is configured for the system so the nodes in each profile can be reached remotely.

1. Delete unused profiles.

   In case you want to cleanup profiles that are no longer in use, complete the following steps:

   a. List existing profiles

      From the `<WAS_home>/bin` directory of the WAS installation you want to check, run the following command:

      ```
      ./manageprofiles.sh -listProfiles
      ```

      By default, a deployment manager profile is prefixed with `Dmgr`, while an application server profile is prefixed with `AppSrv`.

   b. Choose profiles from the list that you want to delete and proceed to step c.

   c. Delete profiles

      For each profile selected in step b, run the following command:

      ```
      ./manageprofiles.sh -delete -profileName <profile_name>
      ```

   d. Delete the leftover profile directories, `<WAS_home>/profiles/<profile_name>`.

2. Create a Deployment Manager profile:

   If you are not using a deployment manager created by the installer, create one now. The command below will create a cell (`<dmgr_host>Cell#`) or top-level organizational unit for the environment, and a node (`<dmgr_host>CellManager#`) or mid-level organizational unit for the deployment manager profile.

   From the `<WAS_home>/bin` directory of the primary server, run the following command:

   ```
   ./manageprofiles.sh -create -profileName <dmgr_profile> -profilePath <WAS_
   home>/profiles/<dmgr_profile> -templatePath <WAS_home>/profileTemplates/dmgr
   -isDefault -defaultPorts
   ```

Note: If you want to enable admin security, append the following options:

```
-enableAdminSecurity true -adminUserName <username> -adminPassword
<password>
```

3. Start the deployment manager.

   ```
   <WAS_home>/bin/startManager.sh -profileName <dmgr_profile>
   ```

4. Create Application Server profiles.

   For each physical server, you will create either a managed or a standalone application server profile. If the application server profile will be local to the deployment manager, it will be managed, and if it is remote, it will be standalone. The commands in the following steps will create a node (`<appsrv_host>Node#`) for each profile.

   a. Create a managed application server profile.

      From the `<WAS_home>/bin` directory of the server in which the profile will exist, run the following command:

      ```
      ./manageprofiles.sh -create -profileName <appsrv_profile> -profilePath
      <WAS_home>/profiles/<appsrv_profile> -templatePath <WAS_
      home>/profileTemplates/managed -isDefault
      ```

   b. Create a standalone application server profile.

      ```
      ./manageprofiles.sh -create -profileName <appsrv_profile> -profilePath
      <WAS_home>/profiles/<appsrv_profile> -templatePath <WAS_
      home>/profileTemplates/default -isDefault
      ```

   Note: The ports used for these profiles will be the next available set of ports based on existing application server profiles on the server.

5. Federate Application Server Profiles with the Deployment Manager.

   The following command will associate the node of the application server profile with the cell that is managed by the deployment manager.

   For each application server profile created in step 3, run the following command from the `<WAS_home>/bin` directory of the server where the profile exists:

   ```
   ./addNode.sh <dmgr_host> <dmgr_soap_port> -profileName <appsrv_profile>
   ```

   Note: The default value for `<dmgr_soap_port>` is `8879`.

6. Start the Node Agents.

   For each application server profile, run the following command from the server on which this profile is located:

   ```
   <WAS_home>/bin/startNode.sh -profileName <appsrv_profile>
   ```

5.2.2 Create a Cluster

In a cluster environment, all WebSphere servers belong to a cluster, and communication between the servers is managed by the application server.

1. Log in to the WebSphere Integrated Solutions administration console:

   ```
   http://<dmgr_host>:<dmgr_port>/admin
   ```

   The default deployment manager port is `9060`.

2. In the tree on the left, expand **Servers**, expand **Clusters**, and click **WebSphere** application server clusters.

3. Create a Cluster:

   a. Click **New**.

   b. Enter a cluster name.

   c. For an environment using session persistence, select **Configure HTTP session memory-to-memory replication**.

   d. Click **Next**.

   e. Create cluster member servers:

      Repeat the following steps for each application server cluster member.

      a. Enter a name for the cluster member.

      b. Select the node where the cluster member will be located.

      c. For the first cluster member, click **Next**, otherwise click **Add Member**.

      d. Click **Next**.

   f. Click **Finish**.

4. Save the changes.

5.2.3 Create Servers

In a non-cluster environment create servers using the following steps:

1. From the administration console, under **Servers**, expand **Server Types**, and click **WebSphere application servers**.

2. Create Servers.

   Repeat the following steps for each application server in the environment:

   a. Click **New**.

   b. Select the node where the server will be located.

   c. Enter a server name.

   d. Click **Next**.

   e. Select the default template and click **Next**.

   f. Select **Generate Unique Ports** and click **Next**.

   g. Click **Finish**.

   h. Click on the newly created server.

   i. Under **Communications**, expand **Ports**.

   j. Note the following ports (They will be added later to the virtual host.):

      – `WC_defaulthost`

      – `WC_defaulthost_secure`

      – `SIP_DEFAULTHOST`

      – `SIP_DEFAULTHOST_SECURE`

3. Save the changes.

5.2.4 Configure Servers

In this section, you will complete the following steps:

- Section 5.2.4.1, "Configuring Web Container Properties"
- Section 5.2.4.2, "Configuring Session Management"
- Section 5.2.4.3, "Setting the Classpath and Java Options"

5.2.4.1 Configuring Web Container Properties

This section explains how to configure web container properties required for WebCenter Sites.

1. From the administration console, under **Servers**, expand **Server Types**, and click **WebSphere application servers**.

2. Add Web Container Properties.

   For each application server, complete the following steps:

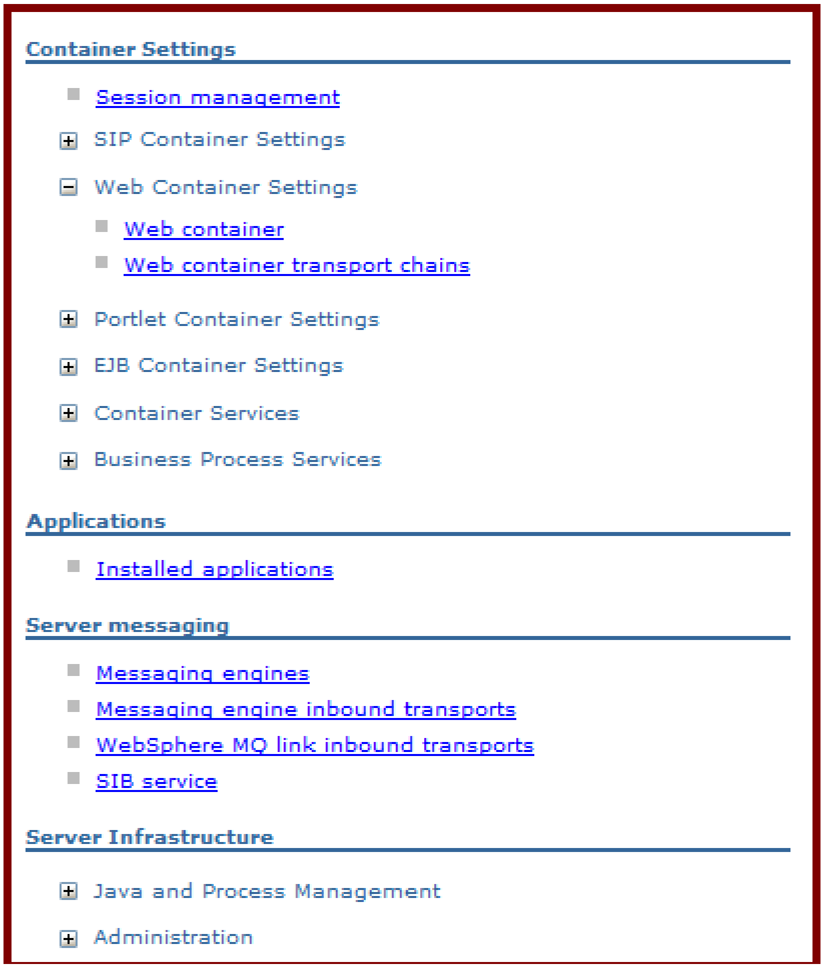
   a. Select the server and expand the **Web Container Settings** node.

   b. In the left-hand pane, expand **Servers**, then expand **Server Types** and click **Application Servers**.

      A list of configured servers is displayed.

   c. Select the application server instance you created for WebCenter Sites (for example, select **server1**), and expand the **Web Container Settings** node (Figure 5–1).

Figure 5–1 Web Container Settings Node



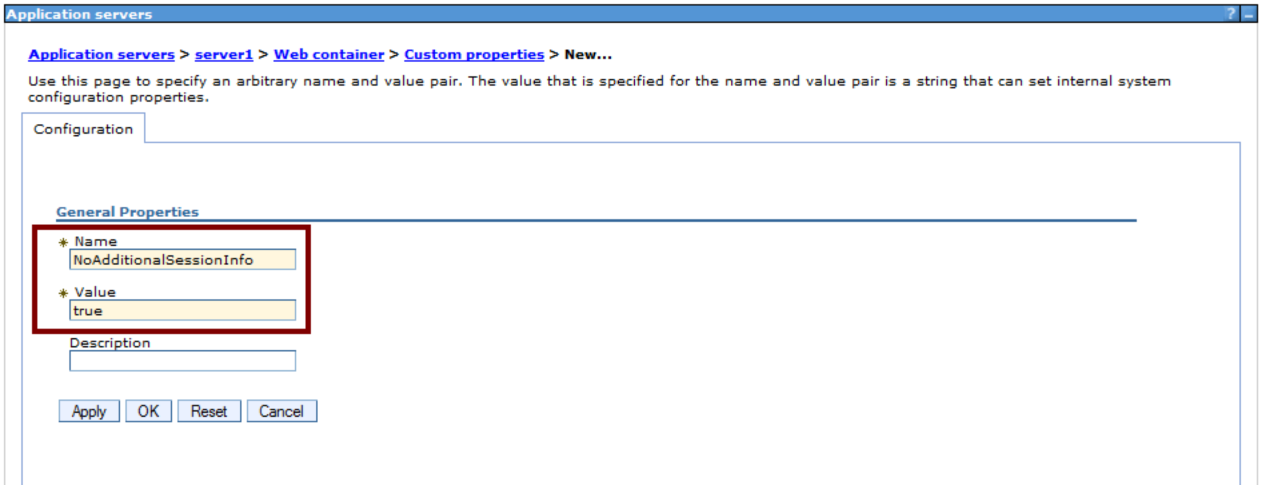
d. Below the Web Container Settings node, click **Web container**.

e. Below the Web container node, click **Custom Properties** and then click **New** (Figure 5–2).

– In the **Name** field, enter `NoAdditionalSessionInfo`.

– In the **Value** field, enter `true`.

Figure 5–2 Custom Properties



Application servers > server1 > Web container > Custom properties > New...

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.

Configuration

General Properties

* Name: NoAdditionalSessionInfo

* Value: true

Description:

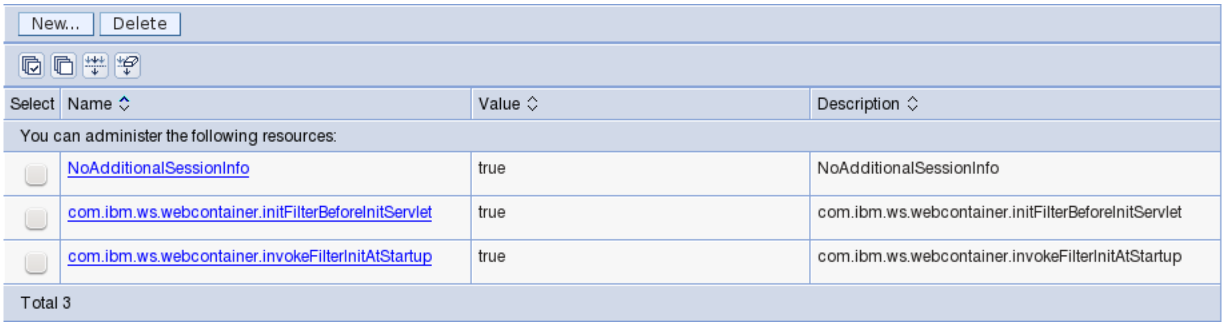
f. Click **Save**. The changes synchronize to all nodes (Figure 5–3).

g. Add the following two properties by following steps e and f:

```
com.ibm.ws.webcontainer.initFilterBeforeInitServlet=true
com.ibm.ws.webcontainer.invokeFilterInitAtStartup=true
```

Web container properties should look something like Figure 5–3.

Figure 5–3 Web Container Properties



Select	Name	Value	Description
You can administer the following resources:			
□	NoAdditionalSessionInfo	true	NoAdditionalSessionInfo
□	com.ibm.ws.webcontainer.initFilterBeforeInitServlet	true	com.ibm.ws.webcontainer.initFilterBeforeInitServlet
□	com.ibm.ws.webcontainer.invokeFilterInitAtStartup	true	com.ibm.ws.webcontainer.invokeFilterInitAtStartup
Total 3			

5.2.4.2 Configuring Session Management

This section explains how to configure session management properties required for WebCenter Sites.

1. From the administration console, under **Servers**, expand **Server Types**, and click **WebSphere application servers**.

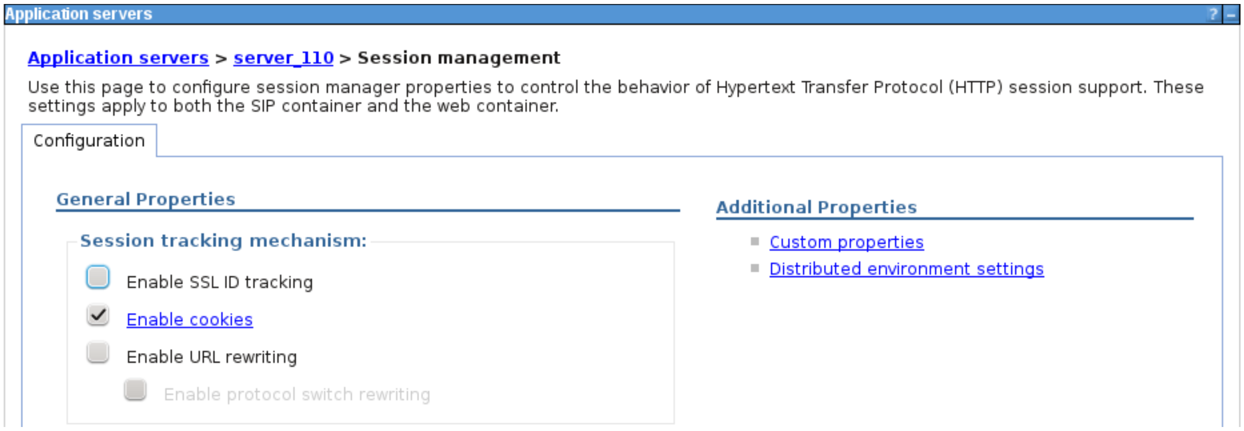
2. Configure Session Management Properties.

   For each application server, complete the following steps:

   a. Click the server and under **Container Settings**, click **Session management**.

   b. Under **General Properties**, click **Enable cookies** (Figure 5–4).

Figure 5–4 Session tracking mechanism: Enable cookies



Application servers

Application servers > server_110 > Session management

Use this page to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support. These settings apply to both the SIP container and the web container.

Configuration

General Properties

Session tracking mechanism:

□ Enable SSL ID tracking

☑ Enable cookies

□ Enable URL rewriting

□ Enable protocol switch rewriting

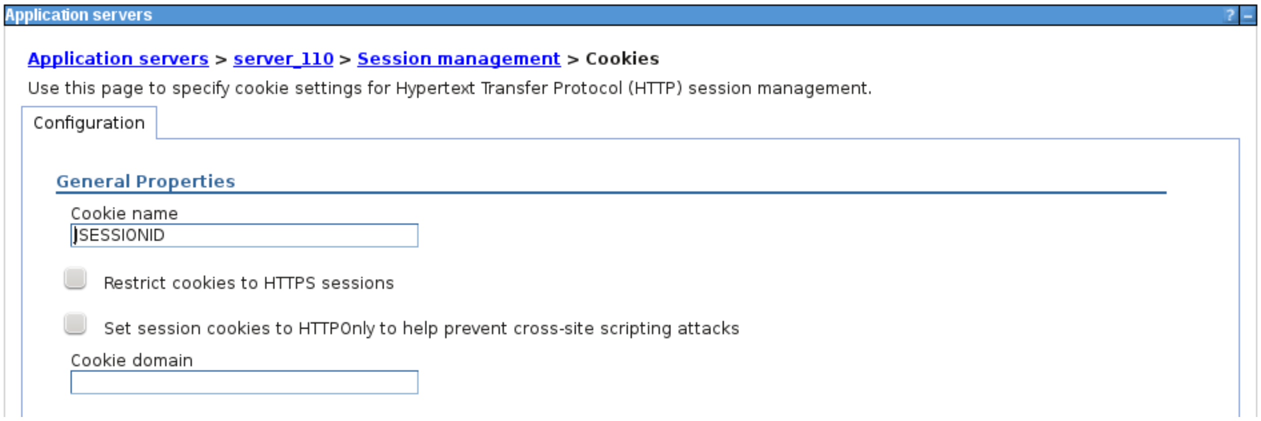
Additional Properties

- Custom properties
- Distributed environment settings

c. Deselect the **Set session cookies to HTTPOnly to help prevent cross-sites scripting** checkbox (Figure 5–5).

This is required to run the tree applet in the WebCenter Sites Admin interface.

Figure 5–5 Specify cookie settings



Application servers

Application servers > server_110 > Session management > Cookies

Use this page to specify cookie settings for Hypertext Transfer Protocol (HTTP) session management.

Configuration

General Properties

Cookie name
JSESSIONID

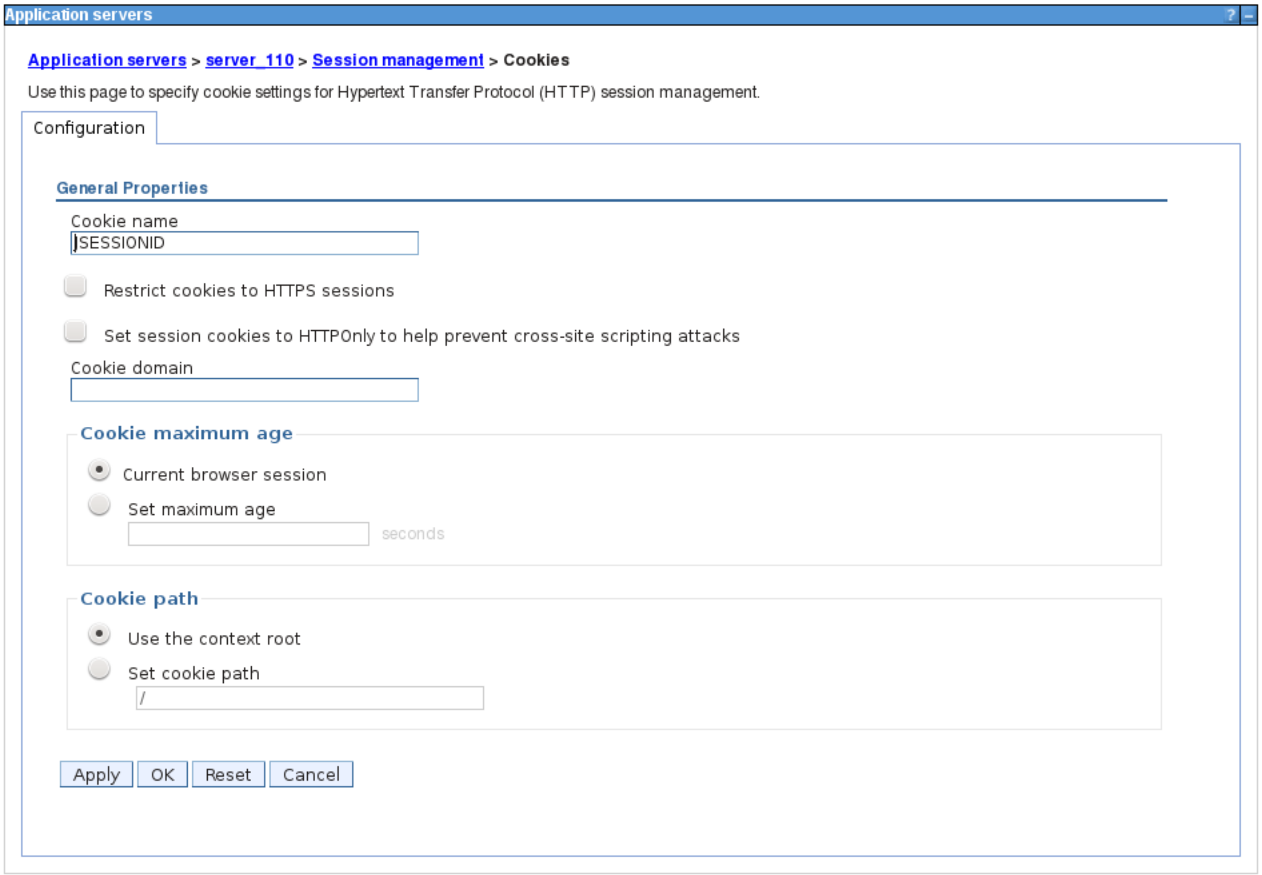
□ Restrict cookies to HTTPS sessions

□ Set session cookies to HTTPOnly to help prevent cross-site scripting attacks

Cookie domain

d. For **Cookie path**, select **Use the context root** (Figure 5–6).

Figure 5–6 Set cookie path



Application servers

Application servers > server_110 > Session management > Cookies

Use this page to specify cookie settings for Hypertext Transfer Protocol (HTTP) session management.

Configuration

General Properties

Cookie name
JSESSIONID

□ Restrict cookies to HTTPS sessions

□ Set session cookies to HTTPOnly to help prevent cross-site scripting attacks

Cookie domain

Cookie maximum age

● Current browser session

○ Set maximum age
seconds

Cookie path

● Use the context root

○ Set cookie path
/

e. Click **Apply** and **OK** to save the changes.

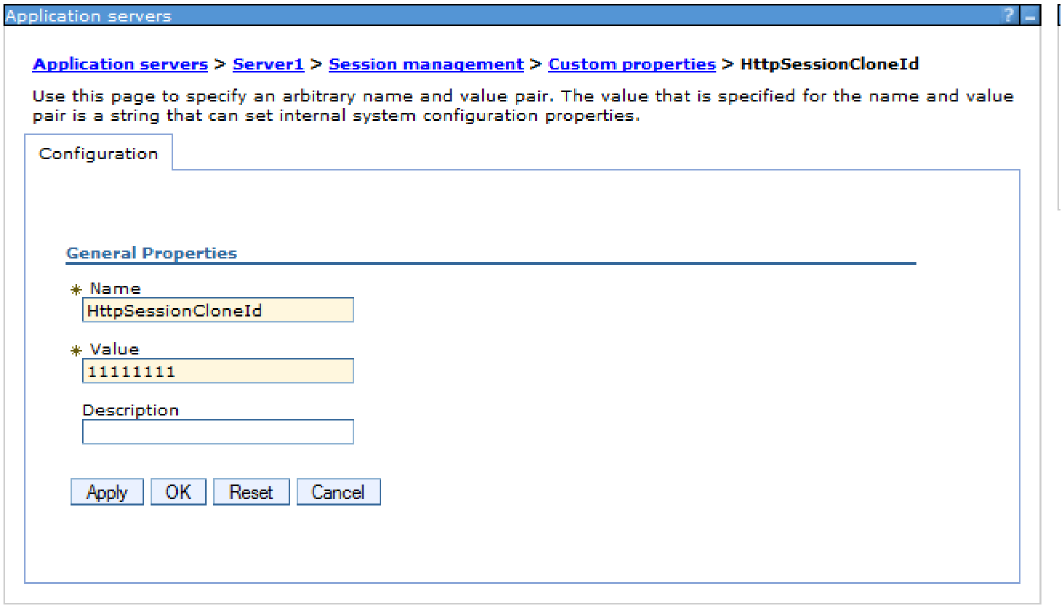
f. Under **Additional Properties**, click **Custom properties**.

g. Click **New**.

h. In the **Name** field, enter `HttpSessionCloneId`.

i. In the **Value** field, enter an 8 or 9 character alphanumeric Id (Figure 5–7). In a cluster environment, this Id is used to identify the cluster member in which a session is located. This Id must be unique for all cluster members.

Figure 5–7 HttpSessionCloneId



Application servers > Server1 > Session management > Custom properties > HttpSessionCloneId

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.

General Properties

* Name
HttpSessionCloneId

* Value
11111111

Description

j. Click **OK** and save the changes.

5.2.4.3 Setting the Classpath and Java Options

This section explains how to set memory arguments as well as add the required Java options for WebCenter Sites.

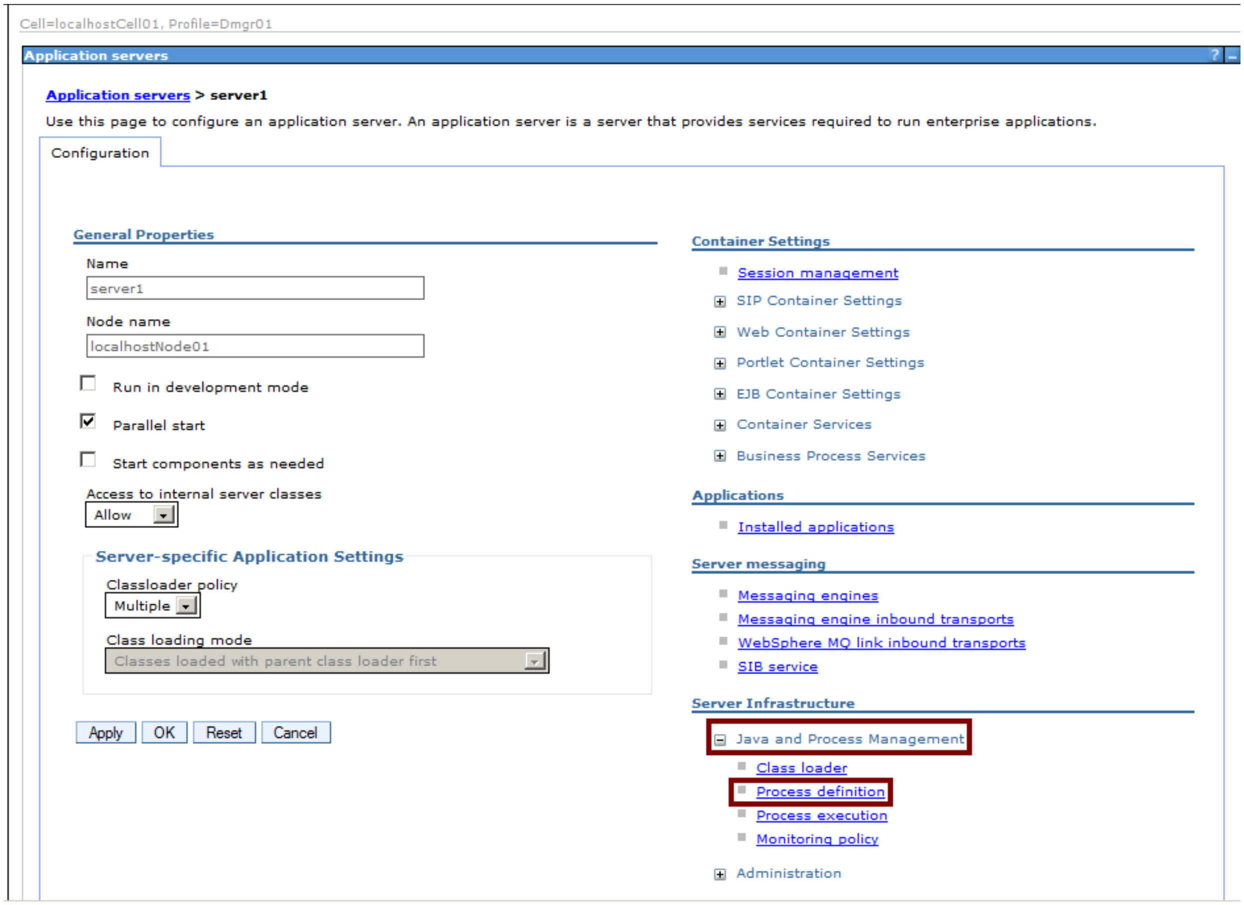
1. From the administration console, under **Servers**, expand **Server Types**, and click **WebSphere application servers**.

2. Set the Classpath and Java Options.

   For each application server, complete the following steps:

   a. Click the server and under **Server Infrastructure**, expand **Java and Process Management** and click **Process definition** (Figure 5–8).

Figure 5–8 Java and Process Management - Process Definition



Cell=localhostCell01, Profile=Dmgr01

Application servers

Application servers > server1

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Configuration

General Properties

Name

server1

Node name

localhostNode01

□ Run in development mode

☑ Parallel start

□ Start components as needed

Access to internal server classes

Allow

Server-specific Application Settings

Classloader policy

Multiple

Class loading mode

Classes loaded with parent class loader first

b. Under **Additional Properties**, click **Java Virtual Machine**.

c. Set the Classpath.

In the box under **Classpath**, enter the following path: `<sites_install_dir>/bin`

d. Set **Memory Arguments**.

– Enter a value in the box under **Initial heap size**. The recommended value is `512`.

– Enter a value in the box under **Maximum heap size**. The recommended value is `2048`.

e. Set Java Options.

In the box under **Generic JVM arguments**, add the following Java options separated by a space:

– `-Dfile.encoding=UTF-8`

This option specifies UTF-8 file encoding.

– `-Dclient.encoding.override=UTF-8`

This option specifies UTF-8 URI encoding.

– `-Dnet.sf.ehcache.enableShutdownHook=true`

This option enables the Ehcache shutdown hook.

- `-Djava.net.preferIPv4Stack=true`

  This option specifies the use of IPv4 addresses.

- `-Duser.timezone=UTC`

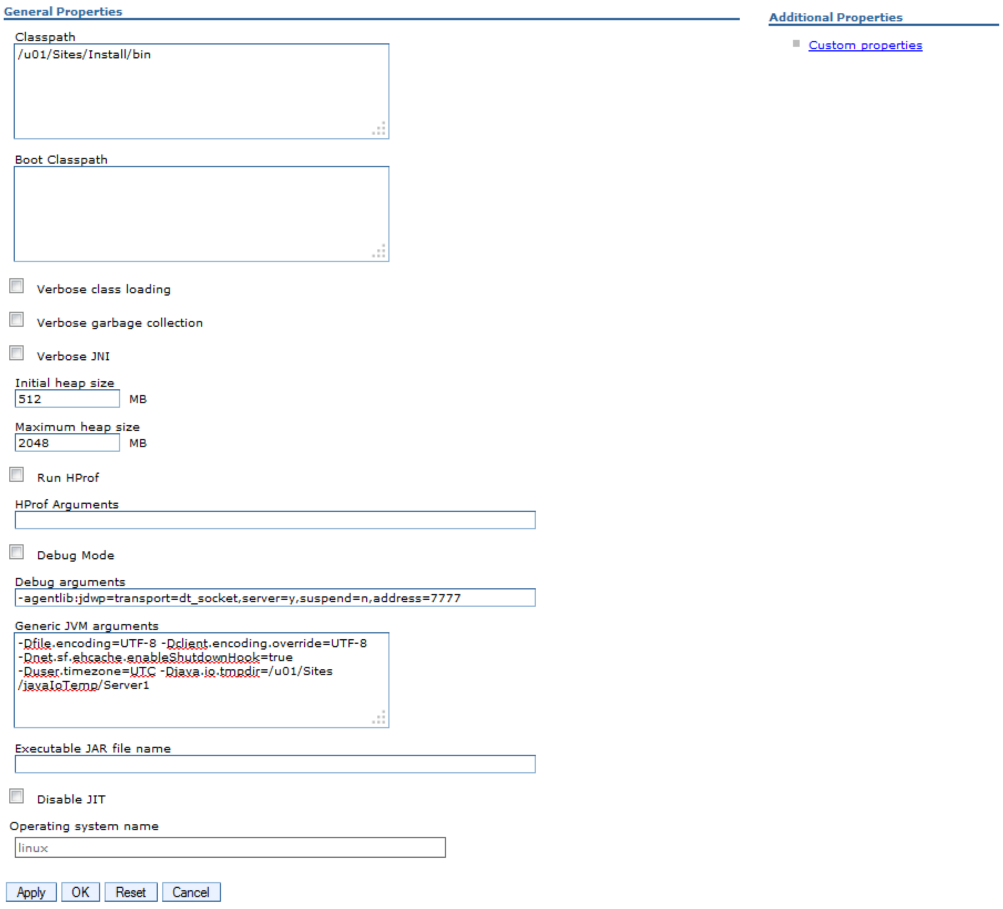
  This option is used to set the time zone for the JVM. It is recommended to set the value to `UTC` across all WebCenter Sites installations. However, any preferable time zone may be used as long as this value is the same for all WebCenter Sites installations.

- `-Djava.io.tmpdir=<path_to_sites_tmpdir>`

  This option specifies which temporary directory the JVM uses. The value (Figure 5–9) should be the path to the directory created in Section 2.1, "Before You Begin."

Note: If `Shift-JIS` characters will be used, add the following option: `-Dcs.useJavaURLDecoder=false`

Figure 5–9 Generic JVM Arguments



General Properties

Classpath: /u01/Sites/Install/bin

Boot Classpath:

□ Verbose class loading

□ Verbose garbage collection

□ Verbose JNI

Initial heap size: 512 MB

Maximum heap size: 2048 MB

□ Run HProf

HProf Arguments:

□ Debug Mode

Debug arguments: -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777

Generic JVM arguments:

```
-Dfile.encoding=UTF-8 -Dclient.encoding.override=UTF-8
-Dnet.sf.ehcache.enableShutdownHook=true
-Duser.timezone=UTC -Djava.io.tmpdir=/u01/Sites
/javaIoTemp/Server1
```

Executable JAR file name:

□ Disable JIT

Operating system name: linux

f. Click **OK** and save the changes.

5.2.5 Create Host Aliases for the Virtual Host

This section explains how to add host aliases to the virtual host for the new servers that were created.

1. From the administration console, expand **Environment**, and click **Virtual hosts**.
2. Click **default_host**.
3. Under **Additional Properties**, click **Host Aliases**.
4. Compare the list of ports that were noted in step 2 of Section 5.2.3, "Create Servers" with the existing host aliases. For each unique port that does not have a host alias, do the following:
   a. Click **New**.
   b. For **Host Name**, enter `*`.
   c. For **Port**, enter the port number.
   d. Click **OK**.
5. Save the changes.

5.2.6 Configuring a WAS Environment for Database Communications

This section explains how to configure the WAS environment to communicate with the database for WebCenter Sites.

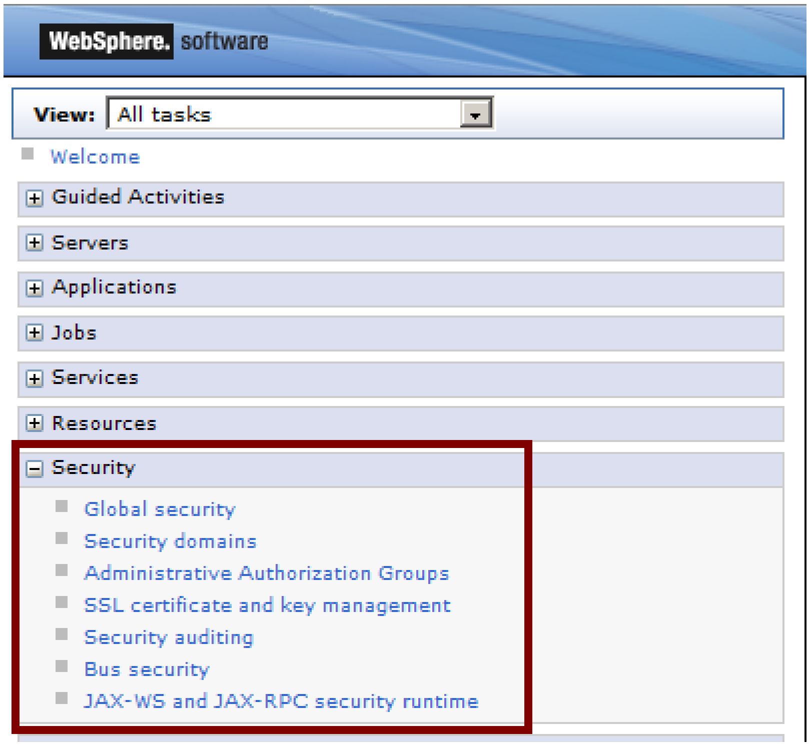
- Section 5.2.6.1, "Creating J2C Authentication Data"
- Section 5.2.6.2, "Creating a JDBC Provider"
- Section 5.2.6.3, "Creating a Data Source"

5.2.6.1 Creating J2C Authentication Data

The J2C authentication data is a list of identities and passwords that can be used for securing connections such as JDBC connections.

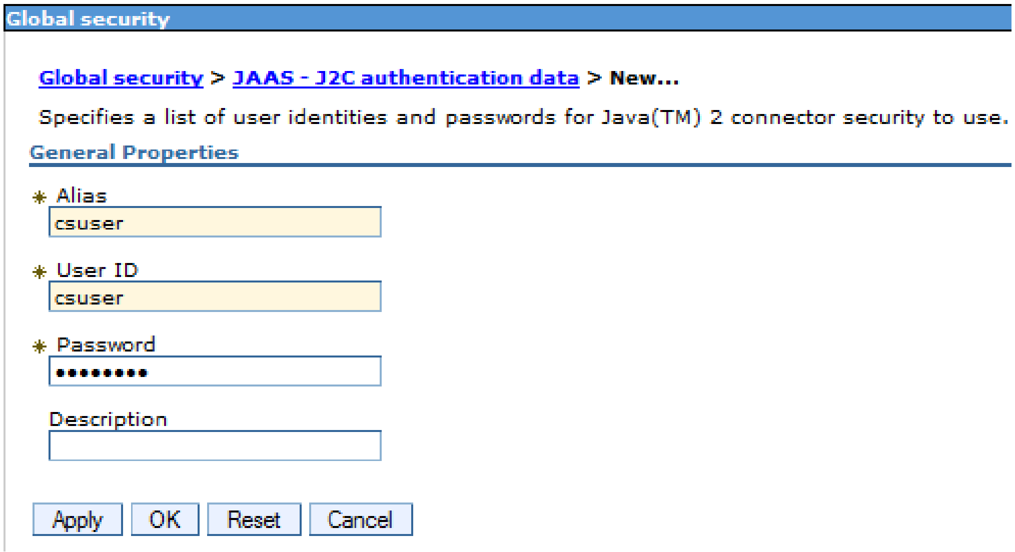
1. From the administration console, expand **Security**, and click **Global Security** (Figure 5–10).

Figure 5–10 Security Node



2. Under **Authentication**, expand **Java Authentication and Authorization**, and click **J2C authentication data**.
3. Click **New**.
4. Create J2C Authentication Data (Figure 5–11).
   a. For **Alias**, enter a unique alias for this user Id/password pair.
   b. For **User ID** and **Password**, enter the information for the user that will be used to connect to the WebCenter Sites database.

Figure 5–11 JAAS – J2C Authentication Data



Global security

Global security > JAAS - J2C authentication data > New...

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: csuser

* User ID: csuser

* Password: ••••••••

Description:

c. Click **OK** and save the changes.

5.2.6.2 Creating a JDBC Provider

A JDBC provider encapsulates all data sources that use a specific JDBC driver implementation. In a cluster environment, all servers will share the same JDBC provider. In a non-cluster environment, a provider must be created for each server.

1. Copy the database driver JAR files corresponding with the database type to the `<WAS_home>/universalDriver/lib` directory:
   - For DB2:
     - `db2jcc.jar`
     - `db2jcc_license_cu.jar`

     Note: `db2jcc4.jar` is not supported.
   - For Oracle: `ojdbc6.jar`
   - For MS SQL Server: `jtds-1.2.5.jar`
2. From the administration console, expand **Resources**, expand **JDBC**, and click JDBC providers.
3. If the provider will be for a cluster environment, under **Scope**, select **Cluster=<cluster_name>** from the dropdown menu. If the provider will be for a non-cluster environment, under **Scope**, select **Node=<appsrv_node>, Server=<server_name>**.
4. Click **New**.
5. In the Create a new JDBC provider screen, complete the following (Figure 5–12):

   a. For **Database type**, select DB2, Oracle, or if you are using MS SQL Server, User-defined.

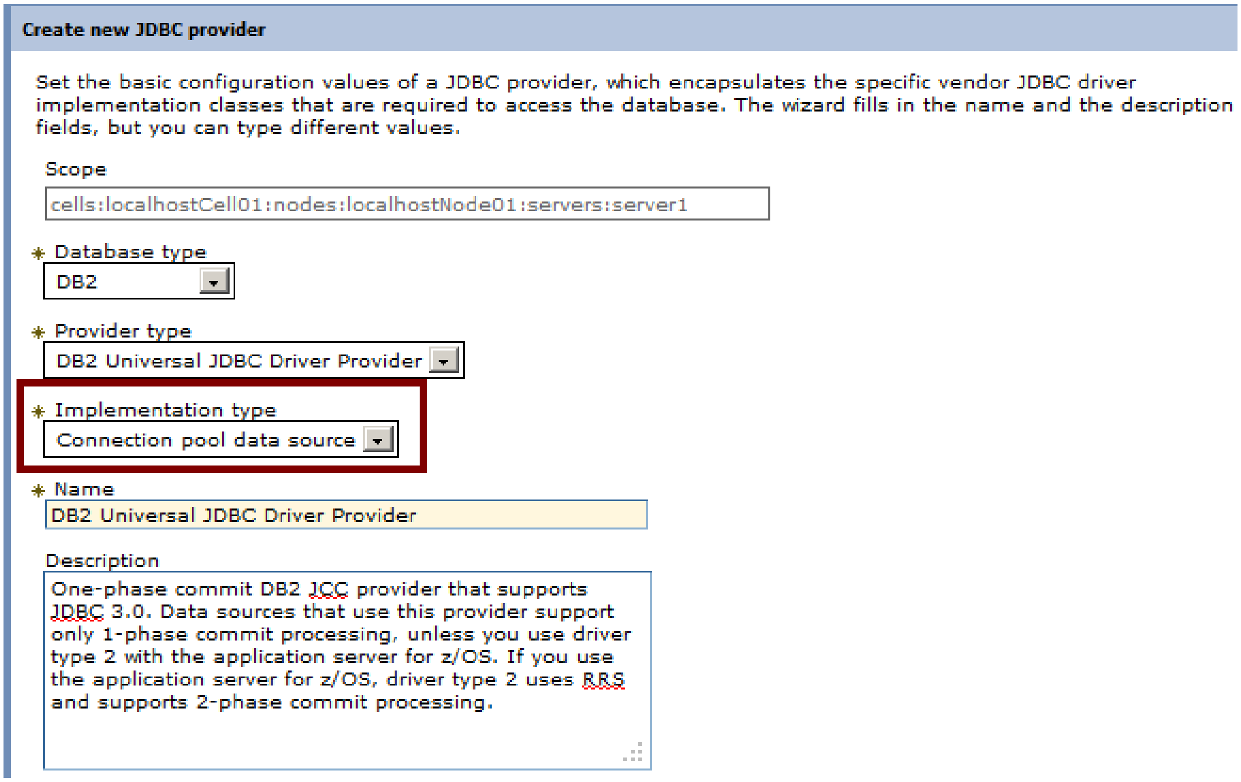
b. For **Provider type**, select DB2 Universal JDBC Driver Provider for DB2, or Oracle JDBC Driver for Oracle. If you are using MS SQL Server, this dropdown is not displayed.

c. For **Implementation type**, if you are using Oracle or DB2, select **Connection pool data source**. If you are using MS SQL Server, enter `net.sourceforge.jtds.jdbcx.JtdsDataSource`.

d. For **Name**, enter a name for the JDBC provider.

e. Click **Next**.

Figure 5–12 Connection Pool Data Source



Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope: cells:localhostCell01:nodes:localhostNode01:servers:server1

* Database type: DB2

* Provider type: DB2 Universal JDBC Driver Provider

* Implementation type: Connection pool data source

* Name: DB2 Universal JDBC Driver Provider

Description: One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.

6. On the Enter database class path information screen (Figure 5–13), do one of the following:

Oracle:

a. For the value of `${ORACLE_JDBC_DRIVER_PATH}`, enter the location of the `ojdb6.jar` file copied in step 1.

b. Click **Next**.

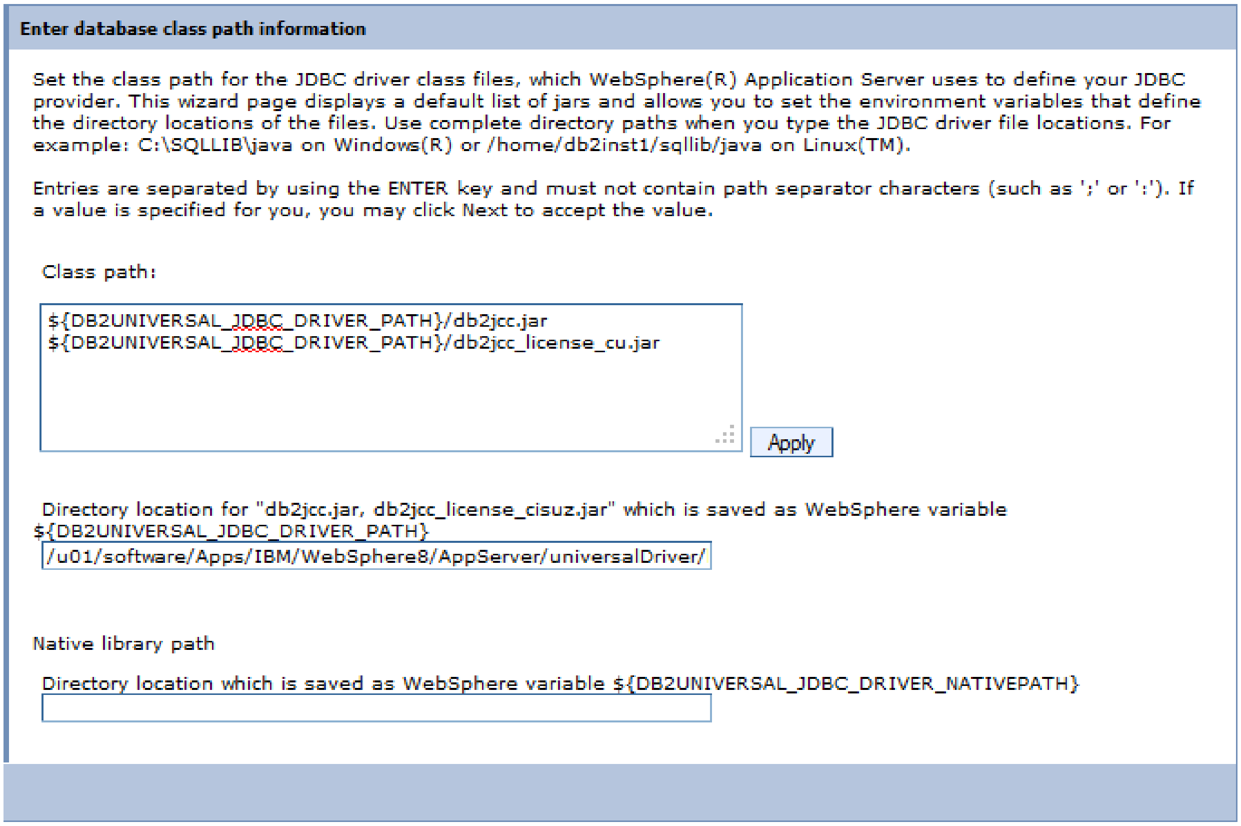
DB2:

a. Under **Class path**, replace `${UNIVERSAL_JDBC_DRIVER_PATH}` with `${DB2UNIVERSAL_JDBC_DRIVER_PATH}` and remove the line containing `db2jcc_license_cisuz.jar`.

b. Click **Apply**.

c. For the value of `${DB2UNIVERSAL_JDBC_DRIVER_PATH}`, enter the location of the `db2cc.jar` and `db2jcc_license_cu.jar` files copied in step 1.

Figure 5–13 Database Class Path Information



Enter database class path information

Set the class path for the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays a default list of jars and allows you to set the environment variables that define the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

Entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). If a value is specified for you, you may click Next to accept the value.

Class path:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
```

Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_PATH}

/u01/software/Apps/IBM/WebSphere8/AppServer/universalDriver/

Native library path

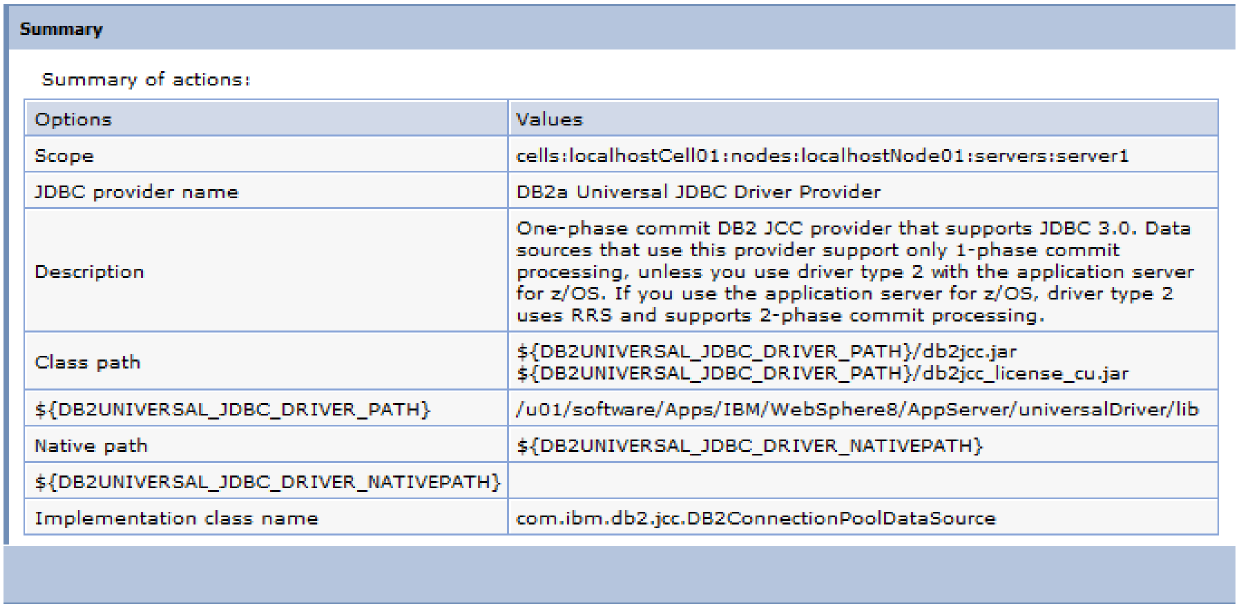
Directory location which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}

d. Click **Next**.

MS SQL Server: Under **Class path**, delete the existing line and add the path to the `jtds-1.2.5.jar` file copied in step 1: `<WAS_home>/universalDriver/lib/jtds-1.2.5.jar`

7. On the Summary screen (Figure 5–14), review the settings you have chosen and click **Finish**.

Figure 5–14 Summary



Summary

Summary of actions:

Options	Values
Scope	cells:localhostCell01:nodes:localhostNode01:servers:server1
JDBC provider name	DB2a Universal JDBC Driver Provider
Description	One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.
Class path	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}	/u01/software/Apps/IBM/WebSphere8/AppServer/universalDriver/lib
Native path	${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}	
Implementation class name	com.ibm.db2.jcc.DB2ConnectionPoolDataSource

8. Save the changes.

5.2.6.3 Creating a Data Source

1. From the administration console, expand **Resources**, expand **JDBC**, and click **Data sources**.

2. If the data source is for a cluster environment, under **Scope**, select **Cluster=<cluster_name>** from the dropdown menu. If the data source is for a non-cluster environment, under **Scope**, select **Node=<appsrv_node>, Server=<server_name>**.

3. Click **New**.

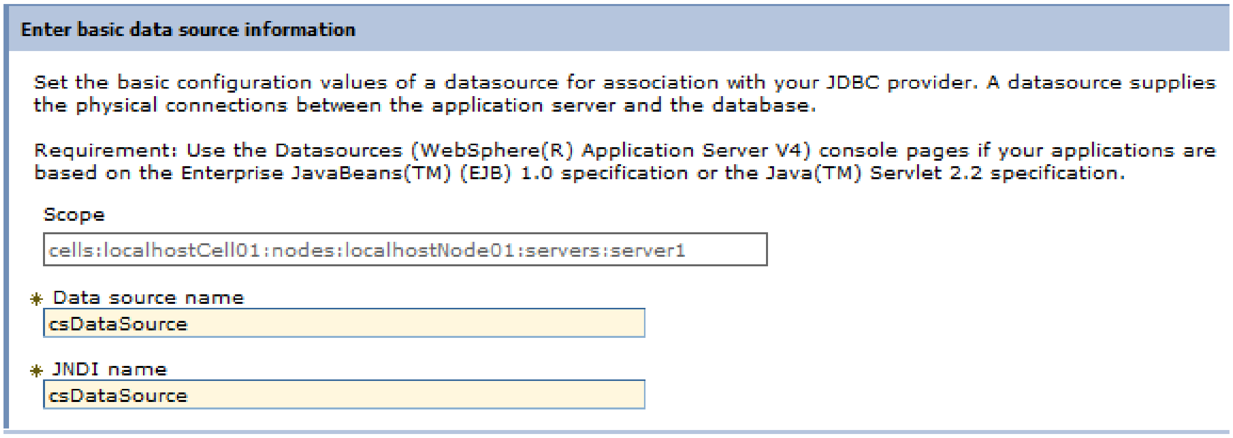
4. In the Enter basic data source information screen (Figure 5–15), do the following:

   a. In the **Data source name** field, enter a unique name for this data source.

   b. In the **JNDI name** field, enter the JNDI name for this data source.

   c. Click **Next**.

Figure 5–15 Basic Data Source Information



Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope

cells:localhostCell01:nodes:localhostNode01:servers:server1

* Data source name

csDataSource

* JNDI name

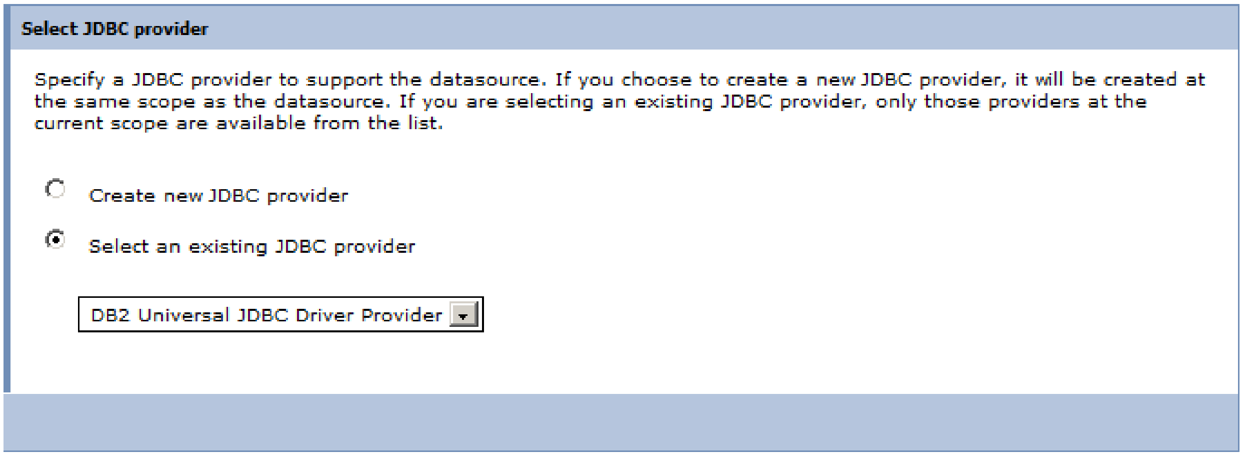
csDataSource

d. Select **Select an existing JDBC provider** (Figure 5–16).

e. In the drop-down list, select the JDBC provider you created in Section 5.2.6.2, "Creating a JDBC Provider."

f. Click **Next**.

Figure 5–16 Existing JDBC Provider



Select JDBC provider

Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

○ Create new JDBC provider

● Select an existing JDBC provider

DB2 Universal JDBC Driver Provider

5. In the Enter database-specific properties for the data source screen (Figure 5–17), do one of the following:

- If you selected a DB2 JDBC provider in step e, do the following:

  a. In the **Database name** field, enter the name of the database WebCenter Sites will be using.

  b. In the **Driver type** drop-down list, select 4.

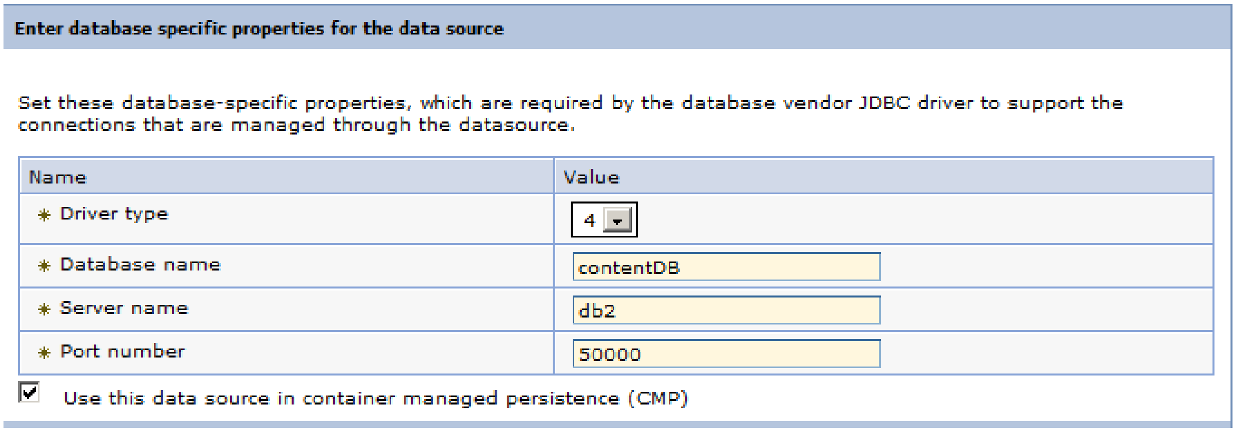
  c. In the **Server name** field, enter the host name or IP address of the machine running the WebCenter Sites database.

  d. In the **Port number** field, enter the port number on which the WebCenter Sites database is listening for connections.

  e. Select the **Use this data source in container managed persistence (CMP)** check box.

f. Click **Next**.

Figure 5–17 Database-Specific Properties for the Data Source



Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* Driver type	4
* Database name	contentDB
* Server name	db2
* Port number	50000

☑ Use this data source in container managed persistence (CMP)

- If you selected an Oracle JDBC provider in step e, do the following:

  a. In the URL field, enter the URL of the database WebCenter Sites will be using. The URL you enter must be in the following format:

  ```
  jdbc:oracle:thin:@//<db_host>:<db_port>/<db_name>
  ```

  where:

  - `<db_host>` is the host name or IP address of the machine running the WebCenter Sites database.

  - `<db_port>` is the port number on which the WebCenter Sites database is listening for connections.

  - `<db_name>` is the name of the WebCenter Sites database.

  b. In the Data store helper class name drop-down list, select **Oracle11g data store helper**.

  c. Select the **Use this data source in container managed persistence (CMP)** check box.

  d. Click **Next**.

- If you selected an SQL Server provider in step e, do the following:

  a. In the **Database name** field, enter the name of the database WebCenter Sites will be using.

  b. In the **Server name** field, enter the host name or IP address of the machine running the WebCenter Sites database.

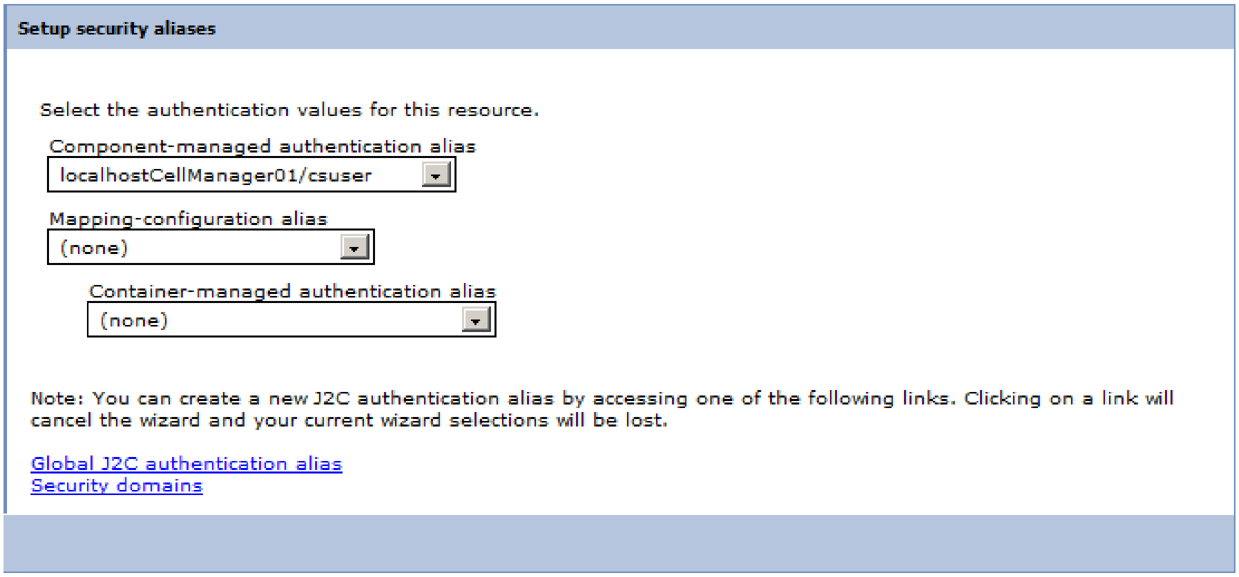
  c. In the **Port number** field, enter the port number on which the WebCenter Sites database is listening for connections.

  d. Select the **Use this data source in container managed persistence (CMP)** check box.

  e. Click **Next**.

6. In the **Component-managed authentication alias** drop-down list (Figure 5–18), select the J2C authentication you created in Section 5.2.6.1, "Creating J2C Authentication Data," then click **Next**.

Figure 5–18 Component-Managed Authentication Alias



Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias: localhostCellManager01/csuser

Mapping-configuration alias: (none)

Container-managed authentication alias: (none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

If you selected an SQL Server provider in step e, do the following:

a. For **Data store helper class name**, leave the default class.

b. Select the **Use this data source in container managed persistence (CMP)** check box.

c. Click **Next**.

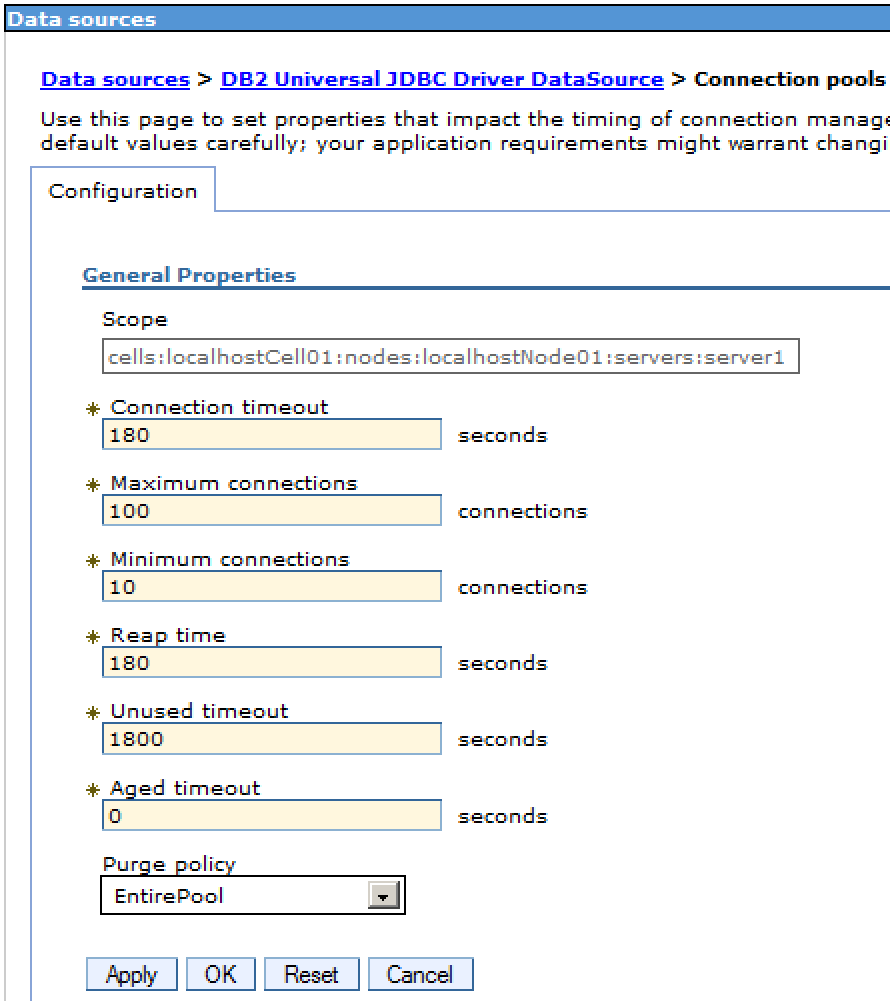
7. Save the changes.
8. In the list of data sources, select the data source you just created.
9. In the Additional Properties area of the Data source screen, click **Connection pool properties**.
10. On the Connection pools screen (Figure 5–19), do the following:

    a. In the **Maximum connections** field, enter `100` (or a value appropriate to your configuration, if known).

    b. In the **Minimum connections** field, enter `10` (or a value appropriate to your configuration, if known).

    c. Click **OK**.

Figure 5–19 Connection Pools



Data sources

Data sources > DB2 Universal JDBC Driver DataSource > Connection pools

Use this page to set properties that impact the timing of connection manage
default values carefully; your application requirements might warrant changi

Configuration

General Properties

Scope
cells:localhostCell01:nodes:localhostNode01:servers:server1

* Connection timeout
180 seconds

* Maximum connections
100 connections

* Minimum connections
10 connections

* Reap time
180 seconds

* Unused timeout
1800 seconds

* Aged timeout
0 seconds

Purge policy
EntirePool

11. Save the changes.

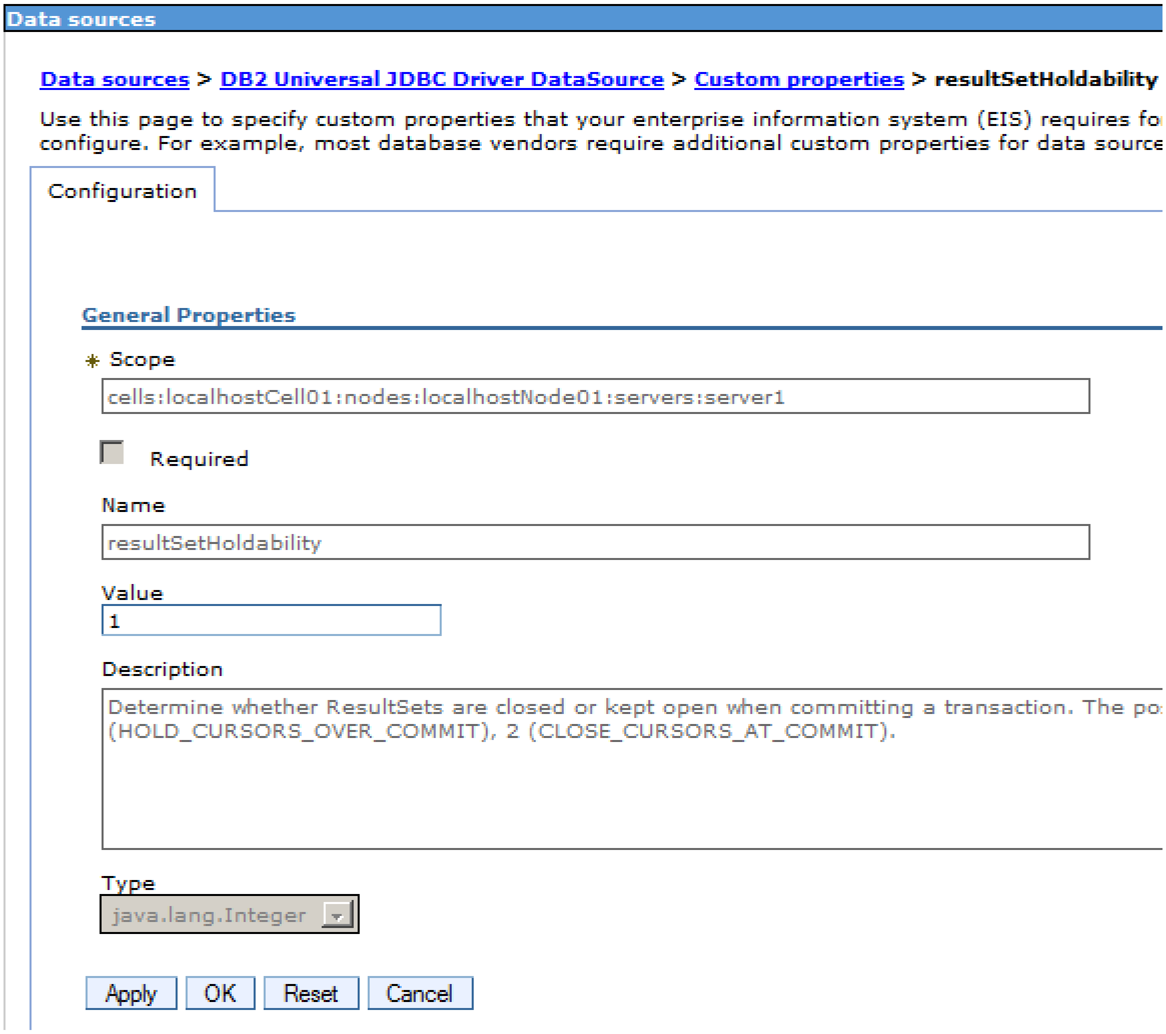
12. For installations running DB2 (Figure 5–20):

a. In the list of data sources, select the data source you just created.

b. In the Additional Properties area of the Data source screen, click **Custom properties**.

c. On the Custom properties screen click **resultSetHoldability**.

Figure 5–20 resultSetHoldability



Data sources

Data sources > DB2 Universal JDBC Driver DataSource > Custom properties > resultSetHoldability

Use this page to specify custom properties that your enterprise information system (EIS) requires fo
configure. For example, most database vendors require additional custom properties for data source

Configuration

General Properties

* Scope
cells:localhostCell01:nodes:localhostNode01:servers:server1

□ Required

Name
resultSetHoldability

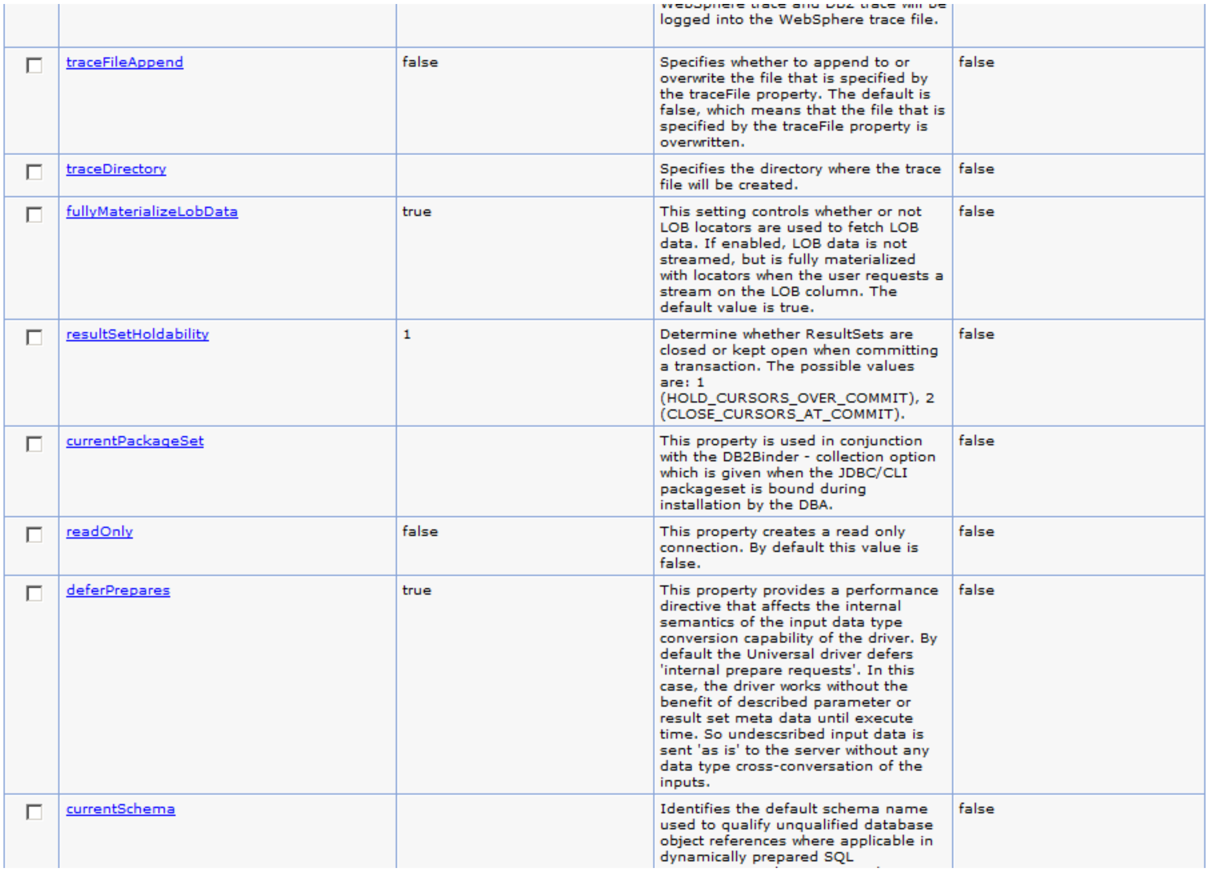
Value
1

Description
Determine whether ResultSets are closed or kept open when committing a transaction. The po
(HOLD_CURSORS_OVER_COMMIT), 2 (CLOSE_CURSORS_AT_COMMIT).

Type
java.lang.Integer

d. In the **Value** field, enter 1 (Figure 5–21). Click **OK**.

Figure 5–21 resultSetHoldability Value



			logged into the WebSphere trace file.	
☐	traceFileAppend	false	Specifies whether to append to or overwrite the file that is specified by the traceFile property. The default is false, which means that the file that is specified by the traceFile property is overwritten.	false
☐	traceDirectory		Specifies the directory where the trace file will be created.	false
☐	fullyMaterializeLobData	true	This setting controls whether or not LOB locators are used to fetch LOB data. If enabled, LOB data is not streamed, but is fully materialized with locators when the user requests a stream on the LOB column. The default value is true.	false
☐	resultSetHoldability	1	Determine whether ResultSets are closed or kept open when committing a transaction. The possible values are: 1 (HOLD_CURSORS_OVER_COMMIT), 2 (CLOSE_CURSORS_AT_COMMIT).	false
☐	currentPackageSet		This property is used in conjunction with the DB2Binder - collection option which is given when the JDBC/CLI packageset is bound during installation by the DBA.	false
☐	readOnly	false	This property creates a read only connection. By default this value is false.	false
☐	deferPrepares	true	This property provides a performance directive that affects the internal semantics of the input data type conversion capability of the driver. By default the Universal driver defers 'internal prepare requests'. In this case, the driver works without the benefit of described parameter or result set meta data until execute time. So undescsribed input data is sent 'as is' to the server without any data type cross-conversation of the inputs.	false
☐	currentSchema		Identifies the default schema name used to qualify unqualified database object references where applicable in dynamically prepared SQL	false

e. Click **New**.

f. For **Name**, enter `allowNextOnExhaustedResultSet`, and for **Value**, enter `1`. Click **OK**.

g. Click **New**.

h. For **Name**, enter `allowNullResultSetForExecuteQuery`, and for **Value**, enter `1`. Click **OK**.

i. In the Messages box, click **Review**.

j. On the Save screen, do the following:

   a. Select the **Synchronize changes with nodes** check box.

   b. Click **Save**.

13. Save the changes.

5.3 Step 2: Integrating WebSphere Application Server with a Supported Web Server

This section explains how to integrate WAS with an IBM HTTP Server or Apache 2.2.x web server using the Web Server Plug-ins for IBM WebSphere Application Server. After the web server and the web server plugin has been installed, perform the following steps:

1. Copy and run the Web Server Configuration script.

The Web Server Plug-ins installer will have created a configuration script `<plugin_home>/bin/configure<web_server_name>.sh`, which is used to add the web server to the deployment manager so that it can be managed using the administration console.

Copy this file to the `<WAS_home>/bin` directory of the primary server on which the deployment manager is running (start it if it is not already), then run the script.

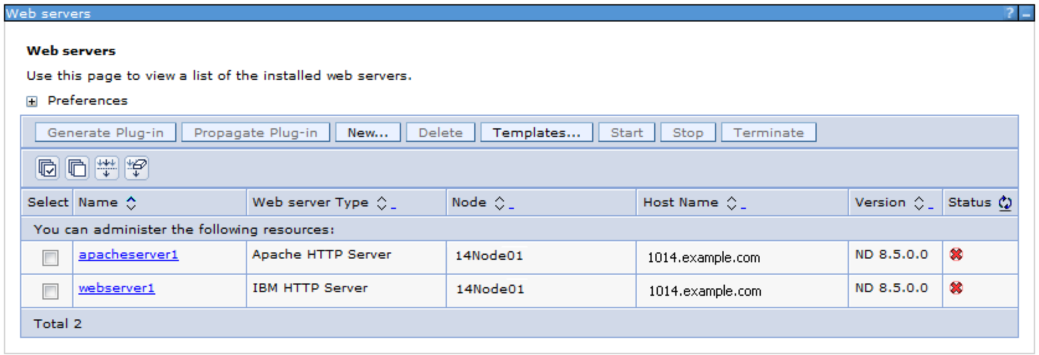
2. Generate and propagate a plug-in configuration.

   From the administration console, a plug-in configuration file can be generated based off of the WAS environment.

   a. From the administration console, on the left, expand **Server Types**, and click **Web servers**.

      You should be able to see the web server that was added using the script (Figure 5–22).

Figure 5–22 Web Servers



Web servers

Use this page to view a list of the installed web servers.

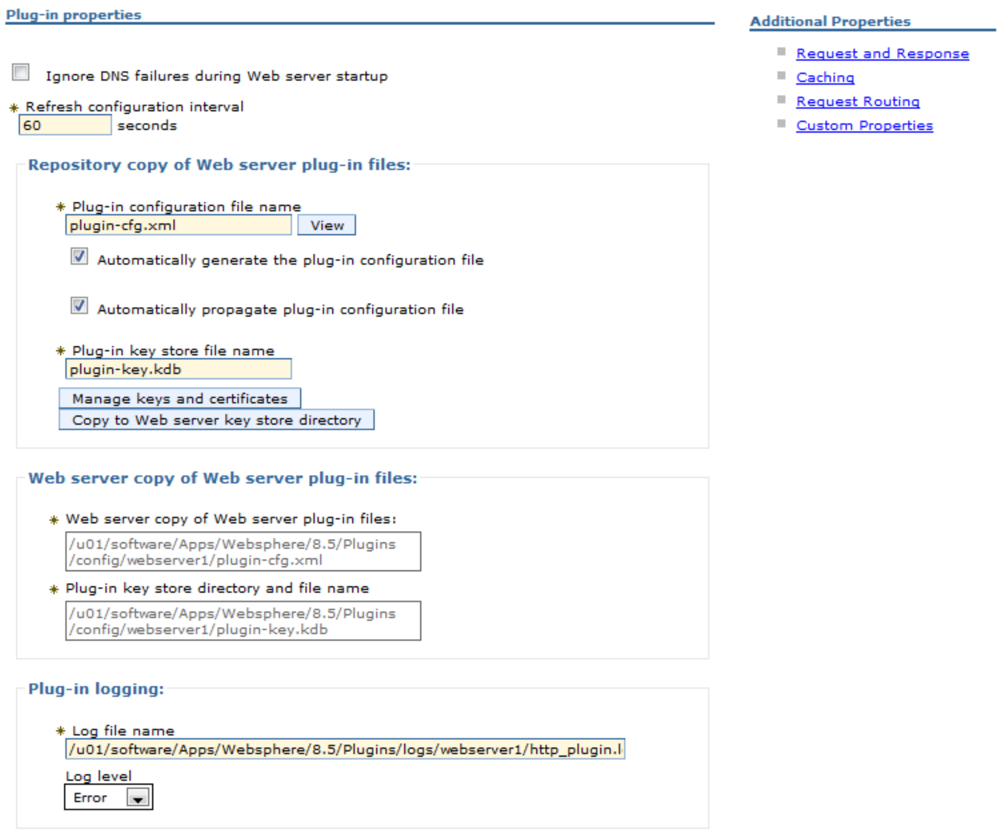
⊞ Preferences

Select	Name	Web server Type	Node	Host Name	Version	Status
You can administer the following resources:						
□	apacheserver1	Apache HTTP Server	14Node01	1014.example.com	ND 8.5.0.0	✖
□	webserver1	IBM HTTP Server	14Node01	1014.example.com	ND 8.5.0.0	✖
Total 2						

   b. Click your web server, then click **Plug-in properties**.

      The Plug-in properties screen is displayed (Figure 5–23).

Figure 5–23 Plug-In Properties



Plug-in properties

☐ Ignore DNS failures during Web server startup

* Refresh configuration interval

60 seconds

Repository copy of Web server plug-in files:

* Plug-in configuration file name

plugin-cfg.xml

☑ Automatically generate the plug-in configuration file

☑ Automatically propagate plug-in configuration file

* Plug-in key store file name

plugin-key.kdb

Web server copy of Web server plug-in files:

* Web server copy of Web server plug-in files:

/u01/software/Apps/Websphere/8.5/Plugins/config/webserver1/plugin-cfg.xml

* Plug-in key store directory and file name

/u01/software/Apps/Websphere/8.5/Plugins/config/webserver1/plugin-key.kdb

Plug-in logging:

* Log file name

/u01/software/Apps/Websphere/8.5/Plugins/logs/webserver1/http_plugin.l

Log level

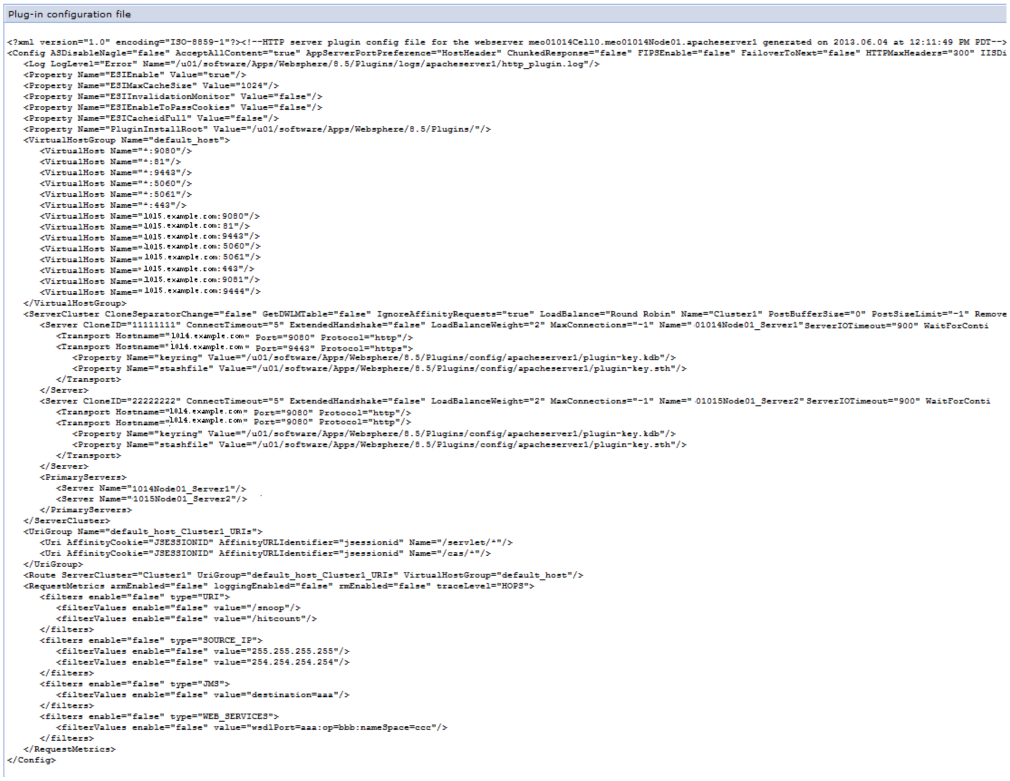
Error

c. To view the plug-in configuration file, click **View** next to `plugin-cfg.xml` (Figure 5–23). In a cluster environment, the file will look like Figure 5–24.

In the Plug-in properties screen, you can disable automatic plugin-in configuration file generation and propagation, as well as modify the log file level.

After installing and deploying the WebCenter Sites and CAS web applications, you will re-generate and re-propagate the plug-in configuration file.

Figure 5–24 Plug-In Configuration File



```
<?xml version="1.0" encoding="ISO-8859-1"?><!--HTTP server plugin config file for the webserver meo01014Cell0.meo01014Node01.apacheserver1 generated on 2013.06.04 at 12:11:49 PM PDT-->
<Config ASDisableNagle="false" AcceptAllContent="true" AppServerPortPreference="HostHeader" ChunkedResponse="false" FIPSEnable="false" FailoverToNext="false" HTTPMaxHeaders="300" IISDi
   <Log LogLevel="Error" Name="/u01/software/Apps/Websphere/8.5/Plugins/logs/apacheserver1/http_plugin.log"/>
   <Property Name="ESIEnable" Value="true"/>
   <Property Name="ESIMaxCacheSize" Value="1024"/>
   <Property Name="ESIInvalidationMonitor" Value="false"/>
   <Property Name="ESIEnableToPassCookies" Value="false"/>
   <Property Name="ESICacheidFull" Value="false"/>
   <Property Name="PluginInstallRoot" Value="/u01/software/Apps/Websphere/8.5/Plugins/"/>
   <VirtualHostGroup Name="default_host">
      <VirtualHost Name="*:9080"/>
      <VirtualHost Name="*:81"/>
      <VirtualHost Name="*:9443"/>
      <VirtualHost Name="*:5060"/>
      <VirtualHost Name="*:5061"/>
      <VirtualHost Name="*:443"/>
      <VirtualHost Name="1015.example.com:9080"/>
      <VirtualHost Name="1015.example.com:81"/>
      <VirtualHost Name="1015.example.com:9443"/>
      <VirtualHost Name="1015.example.com:5060"/>
      <VirtualHost Name="1015.example.com:5061"/>
      <VirtualHost Name="1015.example.com:443"/>
      <VirtualHost Name="1015.example.com:9081"/>
      <VirtualHost Name="1015.example.com:9444"/>
   </VirtualHostGroup>
   <ServerCluster CloneSeparatorChange="false" GetDWLMTable="false" IgnoreAffinityRequests="true" LoadBalance="Round Robin" Name="Cluster1" PostBufferSize="0" PostSizeLimit="-1" Remove
      <Server CloneID="11111111" ConnectTimeout="5" ExtendedHandshake="false" LoadBalanceWeight="2" MaxConnections="-1" Name="01014Node01_Server1"ServerIOTimeout="900" WaitForConti
         <Transport Hostname="1014.example.com" Port="9080" Protocol="http"/>
         <Transport Hostname="1014.example.com" Port="9443" Protocol="https">
            <Property Name="keyring" Value="/u01/software/Apps/Websphere/8.5/Plugins/config/apacheserver1/plugin-key.kdb"/>
            <Property Name="stashfile" Value="/u01/software/Apps/Websphere/8.5/Plugins/config/apacheserver1/plugin-key.sth"/>
         </Transport>
      </Server>
      <Server CloneID="22222222" ConnectTimeout="5" ExtendedHandshake="false" LoadBalanceWeight="2" MaxConnections="-1" Name="01015Node01_Server2"ServerIOTimeout="900" WaitForConti
         <Transport Hostname="1014.example.com" Port="9080" Protocol="http"/>
         <Transport Hostname="1014.example.com" Port="9080" Protocol="http"/>
            <Property Name="keyring" Value="/u01/software/Apps/Websphere/8.5/Plugins/config/apacheserver1/plugin-key.kdb"/>
            <Property Name="stashfile" Value="/u01/software/Apps/Websphere/8.5/Plugins/config/apacheserver1/plugin-key.sth"/>
         </Transport>
      </Server>
      <PrimaryServers>
         <Server Name="1014Node01_Server1"/>
         <Server Name="1015Node01_Server2"/>
      </PrimaryServers>
   </ServerCluster>
   <UriGroup Name="default_host_Cluster1_URIs">
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/servlet/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/cas/*"/>
   </UriGroup>
   <Route ServerCluster="Cluster1" UriGroup="default_host_Cluster1_URIs" VirtualHostGroup="default_host"/>
   <RequestMetrics armEnabled="false" loggingEnabled="false" rmEnabled="false" traceLevel="HOPS">
      <filters enable="false" type="URI">
         <filterValues enable="false" value="/snoop"/>
         <filterValues enable="false" value="/hitcount"/>
      </filters>
      <filters enable="false" type="SOURCE_IP">
         <filterValues enable="false" value="255.255.255.255"/>
         <filterValues enable="false" value="254.254.254.254"/>
      </filters>
      <filters enable="false" type="JMS">
         <filterValues enable="false" value="destination=aaa"/>
      </filters>
      <filters enable="false" type="WEB_SERVICES">
         <filterValues enable="false" value="wsdlPort=aaa:op=bbb:nameSpace=ccc"/>
      </filters>
   </RequestMetrics>
</Config>
```

d. To manually generate and propagate a plug-in, go to the Web servers screen, select the web server you are configuring, and click **Generate Plug-in**.

e. After plug-in generation is successful, select the web server you are configuring, and click **Propagate Plug-in**.

6
Procedures for Installing WebCenter Sites

This chapter provides guidelines for installing WebCenter Sites, deploying on your application server, and connecting to your database.

This chapter contains the following sections:

- Section 6.1, "Overview of the WebCenter Sites Installation Process"
- Section 6.2, "Installation Options"
- Section 6.3, "Installing the WebCenter Sites Application"
- Section 6.4, "Midpoint Configuration Changes"
- Section 6.5, "Deploying the WebCenter Sites and CAS Web Applications"
- Section 6.6, "Restarting WebCenter Sites"
- Section 6.7, "Switching WebLogic Application Server to Production Mode (Delivery Systems Only)"

> **Note:** Before starting the procedures in this chapter, ensure you have both completed pre-requisite steps described in Chapter 2, "Prerequisites for Installing WebCenter Sites," and you have configured your application server, as described in one of the following chapters:
>
> - Chapter 3, "Preparing to Install WebCenter Sites on Oracle WebLogic Application Server"
> - Chapter 4, "Preparing to Install WebCenter Sites on Apache Tomcat Application Server"
> - Chapter 5, "Preparing to Install WebCenter Sites on IBM WebSphere Application Server"

6.1 Overview of the WebCenter Sites Installation Process

After configuring all application servers and if required, integrating with a web server, you are ready to install WebCenter Sites. In a multi-server environment, one server is designated the primary server, while all other servers are designated as secondary. The installer is required to be copied and run on each server and different installation actions are required depending on the server type. Running the installer consists of two stages:

1. In the first stage, the installer gathers necessary configuration information and installs the file structure. At the end of the first stage, the installer displays the Installation Actions window prompting you to deploy WebCenter Sites. In a silent

installation, these steps are displayed on the command-line. These steps include the deployment of the WebCenter Sites application.

If the first stage fails, the installer allows you to go back and modify your configuration options (except the database type), and retry the installation.

> **Note:** If you need to change the type of database you have specified during the installation, you must delete the installed WebCenter Sites file structure and restart the WebCenter Sites installation.

2. In the second stage, the installer populates the database with the tables and data required for WebCenter Sites to function. If the second stage fails, you must drop the database tables, undeploy the WebCenter Sites application, delete the WebCenter Sites file structure, and re-install WebCenter Sites.

6.2 Installation Options

This section explains the ways you can run the WebCenter Sites installer. This step should be completed on the primary server as well as any secondary servers.

- Running the GUI installer

  When you run the GUI installer, a graphical interface guides you through the installation process, prompting you to enter information and select options as required. It also provides access to online help.

- Installing Silently

  To install silently, first generate an `omii.ini` file by running the GUI installer until the Settings Summary screen and then exiting. The silent installer uses this file to install WebCenter Sites.

For instructions on installing WebCenter Sites as a web application, see Section 6.3, "Installing the WebCenter Sites Application."

6.3 Installing the WebCenter Sites Application

> **Note:** Before starting the steps in this section, ensure you have completed the following steps:
>
> - You configured the application server for WebCenter Sites, as described in the chapter for your application server: Section 3.2, "Step 1: Configuring WebLogic Application Server," Section 4.2, "Step 1: Configuring Tomcat Application Server," or Section 5.2, "Step 1: Configuring WebSphere Application Server."
> - If necessary, you integrated the application server with a supported web server, as described in Section 3.3, "Step 2: Integrating Oracle WebLogic Server with a Supported Web Server," Section 4.3, "Step 2: Integrating Tomcat Application Server with a Supported Web Server," or Section 5.3, "Step 2: Integrating WebSphere Application Server with a Supported Web Server."
>
> The WebCenter Sites installer includes the installation of CAS. By default, CAS will be installed on the primary server.

This section includes the following topics:

- Section 6.3.1, "Running the GUI Installer"
- Section 6.3.2, "Installing Silently"

6.3.1 Running the GUI Installer

To install WebCenter Sites using the GUI installer

1. Extract the WebCenter Sites installer archive into a temporary directory.
2. Change to the temporary directory containing the installer files.
3. Execute the installer script:
   - On Windows: `csInstall.bat`
   - On UNIX: `csInstall.sh`

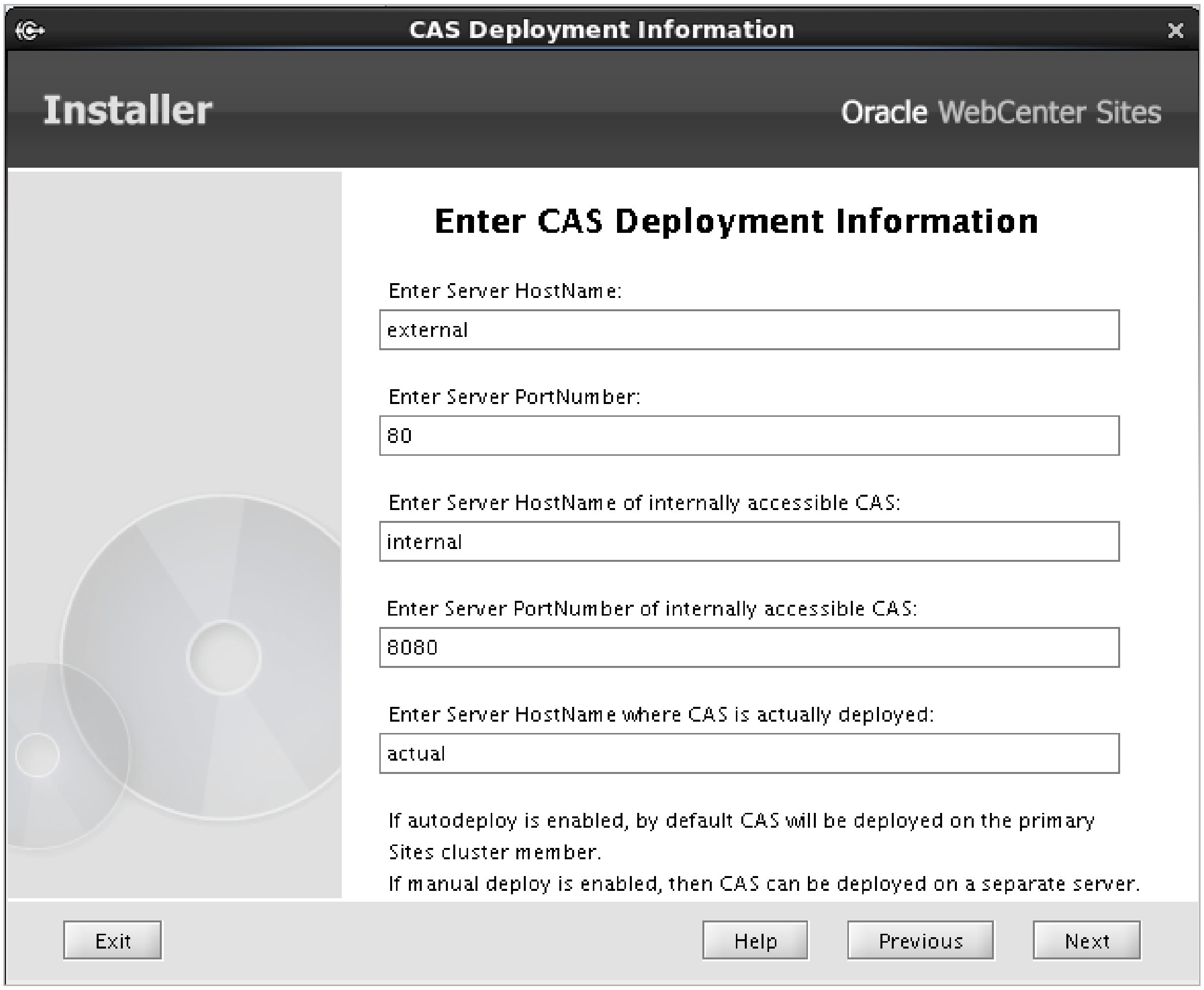
   The installer provides online help at each screen. Read the online help for detailed explanations of the options that are presented in each screen. If you encounter problems during the installation process, consult the online help for possible causes and solutions.
4. On the Clustering screen, if you are installing in a single server environment or on the primary server of a multi-server environment, select **Single Server**. If you are installing on a secondary server of a multi-server environment, select **Cluster Member**.

   > **Note:** Unless all CAS instances are in the same cluster, the CAS clustering settings (`mcast_addr, mcast_port, ClusterName`, and so on) must be unique, that is, these settings must not be similar to any other CAS instances running on the same machine.

5. On the Enter CAS Deployment information screen, do one of the following:
   - For networks with firewalls, fill in the fields as follows:
     - **Enter Server HostName** – Enter the hostname/IP address of the CAS server, as referred to by the external network. If you are clustering CAS, enter the hostname/IP address of the externally facing load balancer.
     - **Enter Server PortNumber** – Enter the port number of the CAS server, as referred to by the external network. If you are clustering CAS, enter the port number of the externally facing load balancer.
     - **Enter Server HostName of internally accessible CAS** – Enter the hostname/IP address of the CAS server, as referred to by the internal network. If you are clustering CAS, enter the hostname/IP address of the load balancer, as referred to by the internal network.
     - **Enter Server PortNumber of internally accessible CAS** – Enter the port number of the CAS server, as referred to by the internal network. If you are clustering CAS, enter the hostname/IP address of the load balancer, as referred to by the internal network.
     - **Enter Server HostName where CAS is actually deployed** – Enter the hostname of the machine where CAS will be deployed.
   - For networks without firewalls (Figure 6–1), fill in the fields as follows:

- **Enter Server HostName** – Enter the hostname/IP address of the CAS server. If you are clustering CAS, enter the hostname/IP address of the load balancer.
- **Enter Server PortNumber** – Enter the port number of the CAS server. If you are clustering CAS, enter the post number of the load balancer.
- **Enter Server HostName of internally accessible CAS** – Enter the hostname/IP address of the CAS server. If you are clustering CAS, enter the hostname/IP address of the load balancer.
- **Enter Server PortNumber of internally accessible CAS** – Enter the port number of the CAS server. If you are clustering CAS, enter the port number of the load balancer.
- **Enter Server HostName where CAS is actually installed** – Enter the hostname of the machine where CAS will be deployed.

Figure 6–1 CAS Deployment Information



Enter CAS Deployment Information

Enter Server HostName: external

Enter Server PortNumber: 80

Enter Server HostName of internally accessible CAS: internal

Enter Server PortNumber of internally accessible CAS: 8080

Enter Server HostName where CAS is actually deployed: actual

If autodeploy is enabled, by default CAS will be deployed on the primary Sites cluster member.
If manual deploy is enabled, then CAS can be deployed on a separate server.

6. On the Settings Summary screen, if you are using the GUI installer to generate an `omii.ini` file, click **Exit**, otherwise click **Next**.
7. Halfway through the installation, the installer displays the Installation Actions window, which lists the steps you must perform to complete the installation. Do the following:

a. Complete the required configuration changes as described in Section 6.4, "Midpoint Configuration Changes."

b. Deploy the WebCenter Sites and CAS web applications. For instructions, see one of the following sections, depending on your application server:

   – Section 6.5.1, "Deploying WebCenter Sites and CAS on WebLogic Server"

   – Section 6.5.2, "Deploying WebCenter Sites and CAS on WebSphere"

   – Section 6.5.3, "Deploying WebCenter Sites and CAS on Tomcat"

c. Start the application server.

> **Note:** In a multi-server environment, only the application server on which the installer was run should be running, while all other server should be shutdown.

d. Click **Test**, if all tests succeed, click **OK** to continue the installation.

8. When the installation process completes successfully, do the following:

   a. For all installations, continue to Section 6.6, "Restarting WebCenter Sites."

   > **Note:** If you deployed WebCenter Sites on WebLogic Server, also complete the steps in Section 6.7, "Switching WebLogic Application Server to Production Mode (Delivery Systems Only)."

   b. For all installations, follow up by completing the post-installation steps in Chapter 7, "WebCenter Sites Post-Installation Steps."

6.3.2 Installing Silently

1. Generate an `omii.ini` file using the GUI installer by following steps 1 through 6 of Section 6.3.1, "Running the GUI Installer."

   > **Note:** This only needs to be done once per environment. For each additional server beyond the first, you may use the same `omii.ini` file after any necessary port and path changes. For installing silently on a secondary server, make sure the following properties are set:
   >
   > - `- CSInstallType=cluster`
   > - `- IsPrimaryClusterMember=false`

2. Copy the `omii.ini` file to a folder outside of `<sites_install_dir>` and clear out any files created by the GUI installer.

3. If you are installing a delivery system, you must set unique passwords for the `fwadmin` and `ContentServer/SatelliteServer` users:

   a. Open the `cscore.xml` file in the ContentServer folder of the temporary directory.

   b. Set passwords in the following section:

```
<IF COND="Variables.bShowInstallTypeDialog=false">
  <THEN>
```

```
     <DIALOGACTION>
       <SETVARIABLE NAME="passwordVar" VALUE=" "/>
       <SETVARIABLE NAME="passwordAdminVar" VALUE=" "/>
     </DIALOGACTION>
    </THEN>
  </IF>
```

   a. Set the password for the fwadmin user in the VALUE field following `NAME="passwordVar"`.

   b. Set the password for the `ContentServer/SatteliteServer` user in the `VALUE` field following `NAME="passwordAdminVar"`.

  c. Save and close the file.

4. Edit the `install.ini` file in the root folder of the temporary directory:

   a. Set the `nodisplay` property to `true`.

   b. Uncomment the `loadfile` property and set it to the path and name of the `omii.ini` file from step 2.

   Note: Verify that you have correctly specified the file system path. For example, for Windows:

   ```
   CSInstallDirectory=C\:/csinstall

   - or -

   c\:\\install
   ```

   c. Save and close the file.

5. Change to the temporary directory containing the installer files.

6. Execute the installer script:

   - On Windows: `csInstall.bat -silent`
   - On UNIX: `csInstall.sh -silent`

7. Halfway through the installation, the installer will print a prompt with steps you must perform to complete the installation. Do the following:

   a. Complete required configuration changes as described in Section 6.4, "Midpoint Configuration Changes."

   b. Deploy the WebCenter Sites and CAS web applications. For instructions, see one of the following sections, depending on your application server:

      – Section 6.5.1, "Deploying WebCenter Sites and CAS on WebLogic Server"

      – Section 6.5.2, "Deploying WebCenter Sites and CAS on WebSphere"

      – Section 6.5.3, "Deploying WebCenter Sites and CAS on Tomcat"

   c. Start the application server.

   Note: In a multi-server environment, only the application server on which the installer was run, should be running, while all other server should be shutdown.

d. Open the following URLs in a browser, using the hostname, context root, and port specified for the WebCenter Sites web application in the installer:

   – `http://<hostname>:<port>/<context_root>/HelloCS`

   – `http://<hostname>:<port>/<context_root>/CatalogManager?ftcmd=pingdb`

   If the tests succeed, press **Enter** in the window where the installer was run to continue the installation.

6.4 Midpoint Configuration Changes

This section explains the steps required at the midpoint of the WebCenter Sites installer, prior to deployment.

> **Note:** When setting a time to live value to control the propagation of multicast packets, use one of the following values based on the distance between communicating servers:
>
> - `0` (multicast packets restricted to the local server)
> - `1` (multicast packets restricted to the same subnet)
> - `32` (multicast packets restricted to the same region)
> - `64` (multicast packets restricted to the same continent)
> - `255` (multicast packets are unrestricted)

This section includes the following:

- Section 6.4.1, "Copy CAS Files"
- Section 6.4.2, "Modify CAS Files"
- Section 6.4.3, "Modify Web Application Files"

6.4.1 Copy CAS Files

If you selected **Single Server** in step 4 of Section 6.3.1, "Running the GUI Installer", and you will be deploying the CAS web application on physical servers other than those on which the WebCenter Sites web application is installed, copy the files below from the `<sites_install_dir>/bin` directory to a location on those servers. This location must be in the path of the application servers where CAS will be deployed.

If you selected **Cluster Member** in step 4 of Section 6.3.1, "Running the GUI Installer", copy the following files from the `<sites_install_dir>/bin` directory of the primary server, to the `<sites_install_dir>/bin` directory of the secondary server.

- `host.properties`
- `cas.properties`
- `jbossTicketCacheReplicationConfig.xml`
- `customBeans.xml`

6.4.2 Modify CAS Files

If you selected Single Server in step 4 of Section 6.3.1, "Running the GUI Installer", modify the following files (located in the sitesinstall/bin folder) on the primary server, and if necessary, any other physical servers on which the files were copied.

If you selected **Cluster Member** in step 4 of Section 6.3.1, "Running the GUI Installer", modify the following files (located in the sitesinstall/bin folder) on the secondary server:

- host.properties

  The host.name property is used to identify which server a ticket was generated from. By default it uses the format cas-<local_hostname>-<cluster_id>. While you may change the format of this value, it must be unique for all servers.

- jbossTicketCacheReplicationConfig.xml

  In the **ClusterConfig** section of the file, modify the following attributes of the UDP tag:

  - mcast_addr

    In a multi-CAS server environment, the mcast_addr and mcast_port combination must be the same for all members, but it must also be unique from other CAS environments.

  - mcast_port

    In a multi-CAS server environment the mcast_addr and mcast_port combination must be the same for all members, but it must also be unique from other CAS environments.

  - bind_addr

    Set the value to the hostname or IP address (required if multi-homed server) of the physical server on which the CAS server will be running.

  - ip_ttl

    Set this time to live value based on the note above.

- customBeans.xml

  The customBeans.xml file contains a list of trusted URLS that CAS can redirect a visitor to upon successful login. By default, the URL to the WebCenter Sites web application is specified in this file.

  If you installed WebCenter Sites with other WebCenter Sites web applications (such as Community-Gadgets), you must list the URLs of those web applications in this file.

  For example:

```
<bean id="customServiceValidator"
class="com.fatwire.cas.web.ServiceValidator">
<property name="validUrls">
<list>
.
<value>http://hostname:port/cs/wem/fatwire/wem/Welcome</value>
<value>http://hostIP:port/SitesWebapp/*</value>
<value>http://hostname:port/Community-GadgetsWebapp/*</value>
.
</list>
</property>
```

You can list URLs in the following formats:

- List complete URLs for exact comparison. For example:

  ```
  http://hostname:port/cs/wem/fatwire/wem/Welcome
  ```

  allows access to only the specified URL (the WebCenter Sites Welcome page in this example). This format does not allow access to sub-resources.

- List group URLs by entering `/*` at the end. For example:

  ```
  http://hostIP:port/SitesWebapp/*
  ```

  allows access to all resources under the WebCenter Sites web application (*`SitesWebapp`*) including access to `http://hostIP:port/`*`SitesWebapp`*, which is the exact URL to the WebCenter Sites web application.

If you intend to access WebCenter Sites and its web applications using `hostname:port` and `hostIP:port` interchangeably, both URLs must be specified in the `customBeans.xml` file.

> **Note:** If you are working in a clustered WebCenter Sites environment, the host name (or host IP address) and port number must point to the load balancer.

6.4.3 Modify Web Application Files

Modify the following web application files below for a multi-server environment.

> **Note:** For a WebSphere environment, you will need to explode the WebCenter Sites `.ear` file located in `<sites_install_dir>/ominstallinfo/app` to a temporary directory prior to making changes. After the changes have been made, re-compress the `.ear` file and replace the original file.
>
> For Tomcat and WebLogic, the WebCenter Sites and CAS `.war` files are already exploded by the installer for deployment.

- Cache files (`cs-cache.xml`, `cas-cache.xml`, `ss-cache.xml`, `linked-cache.xml`)

  Modify these cache files in the `WEB-INF/classes` directory of WebCenter Sites web application. Changes are required for the following attributes of the `cacheManagerPeerProviderFactory` tag:

  - `multicastGroupAddress`

    The `multicastGroupAddress` and `multicastGroupPort` combination must be the same for all members, but must also be unique from other WebCenter Sites environments.

  - `multicastGroupPort`

    The `multicastGroupAddress` and `multicastGroupPort` combination must be the same for all members, but it must also be unique from other WebCenter Sites environments.

  - `timeToLive`

    Set this time to live value based on the following note:

Note: The `multicastGroupPort` property must be set identically across cluster members. For example, if CS1 specifies `multicastGroupPort=4444` in its `cs-cache.xml` file, then CS2 and CS3 must specify the same setting in their files.

The `timeToLive` property specifies the number of hops allowed to a packet, which determines how far the packet propagates. Valid values range between `0` and `255` with the following restrictions:

- `0` - same host
- `1` - same subnet
- `32` - same site
- `64` - same region
- `128` - same continent
- `255` - unrestricted

– Bootstrapping (for `cas-cache.xml` only)

When a node is restarted when using clustered CAS, there is an option to pre-populate the cache with existing session and ticket information from the neighboring nodes on server startup. This may be helpful in a very small percentage of cases when the session is replicated onto the restated node. However, this may increase the application server startup time. This feature is turned off by default.

Enable this feature by uncommenting the lines in `cas-cache.xml` after the commented line

```
<!-- Uncomment following lines to enable bootstrapping -->
```

In all the three sections, set `replicateAsynchronously=true` to pre-populate the CAS cache asynchronously. Otherwise set `replicateAsynchronously=false` as there is a possibility of a race condition when setting `replicateAsynchronously=true`.

- (WebLogic Only) `weblogic.xml`

To set the URI encoding character set on WebLogic, in the `weblogic.xml` file in the `WEB-INF` directory of the WebCenter Sites web application, add the following lines inside the `weblogic-web-app` element:

```
<charset-params>
   <input-charset>
     <resource-path>*</resource-path>
     <java-charset-name>UTF-8</java-charset-name>
   </input-charset>
</charset-params>
```

If you are configuring a WebLogic cluster environment with session persistence enabled, modify the `weblogic.xml` file in the `WEB-INF` directory of the WebCenter Sites and CAS web applications by adding the following line inside the session-descriptor element:
`<persistent-store-type>replicated</persistent-store-type>`

- (Tomcat Only) `web.xml`

If you are configuring a Tomcat cluster environment, modify the `web.xml` file in the `WEB-INF` directory of the WebCenter Sites web application by adding the following tag as the first line after the `web-app` tag: `<distributable/>`

6.5 Deploying the WebCenter Sites and CAS Web Applications

This section contains the following topics:

- Section 6.5.1, "Deploying WebCenter Sites and CAS on WebLogic Server"
- Section 6.5.2, "Deploying WebCenter Sites and CAS on WebSphere"
- Section 6.5.3, "Deploying WebCenter Sites and CAS on Tomcat"

6.5.1 Deploying WebCenter Sites and CAS on WebLogic Server

This section describes how to deploy the WebCenter Sites and CAS web applications on WebLogic application server using either the WebLogic Scripting Tool or the Administration Console.

This section includes the following topics:

- Section 6.5.1.1, "Deploying using the WebLogic Scripting Tool"
- Section 6.5.1.2, "Deploying from the WebLogic Administration Console"

6.5.1.1 Deploying using the WebLogic Scripting Tool

The WebLogic Scripting Tool is a command-line scripting environment used to managed WebLogic domains. It requires the administration server be running. You can use the tool to deploy the WebCenter Sites and CAS web applications by following the steps below.

> **Note:** In a cluster environment, deployment of the WebCenter Sites and CAS applications only needs to be done once at the midpoint of the primary server installation.

1. Create a Jython script.

   The WLST will run the commands contained in this script to deploy the applications. Create a new `.py` file containing the commands below:

   > **Note:** For the `connect` command, enter the information for the WebLogic admin server.
   >
   > For the `deploy` command, use the application name specified in the installer, the path to the exploded web application, and the name of the target server or cluster. In a cluster environment, use the name of the cluster as the target.

   ```
   connect('<admin_user>', '<admin_password>', 't3://<admin_host>:<admin_port>')
   deploy('<sites_application_name>', '<sites_deploy_home>', '<target>',
   'nostage')
   deploy('cas', '<cas_deploy_home>', '<target>', 'nostage')
   disconnect()
   exit()
   ```

2. Run the WebLogic Scripting Tool.

   To deploy the applications, run the following command:

   ```
   <weblogic_home>/common/bin/wlst.sh <path_to_jython_script>
   ```

6.5.1.2 Deploying from the WebLogic Administration Console

This section shows you how to deploy the Sites and CAS web applications on WebLogic application server from the WebLogic administration console. Complete the steps below for each application.

> **Note:** In a cluster environment, deployment of the WebCenter Sites and CAS applications only needs to be done once at the midpoint of the primary server installation.

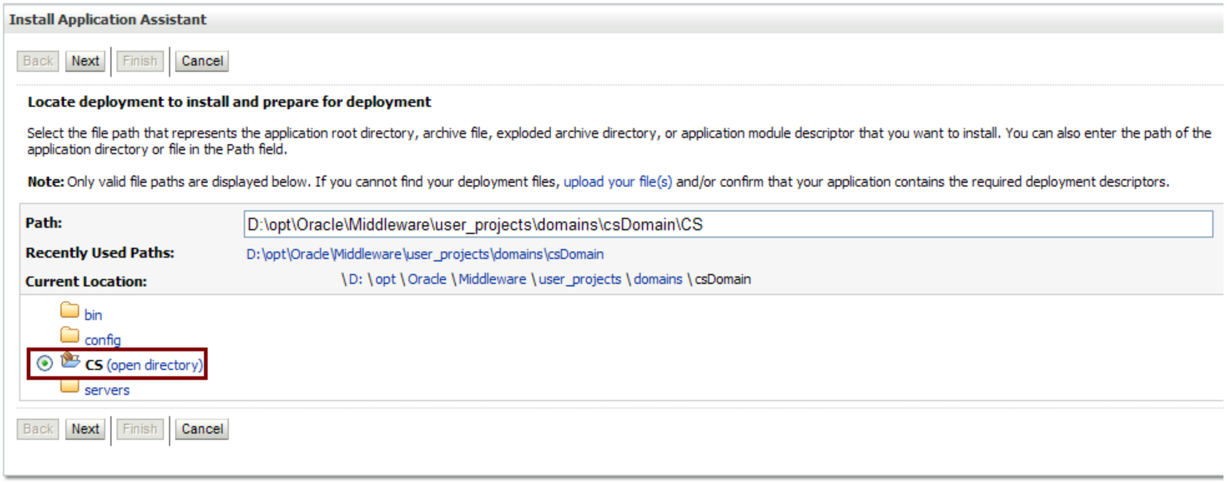
1. Log in to the WebLogic Server Administration Console.
2. Deploy the web application (Figure 6–2):

   a. In the Domain Structure panel, click **Deployments**.

   b. Click the **Install** button. If necessary, change the current location to the path that is specified in the Manual Deployment screen of the WebCenter Sites installer.

   c. Select the directory with the specified name of the WebCenter Sites web application, marked **open directory**, then click **Next**.

Figure 6–2 Install Application Assistant - CS (open directory)



Install Application Assistant

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. You can also enter the path of the application directory or file in the Path field.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contains the required deployment descriptors.

Path: D:\opt\Oracle\Middleware\user_projects\domains\csDomain\CS

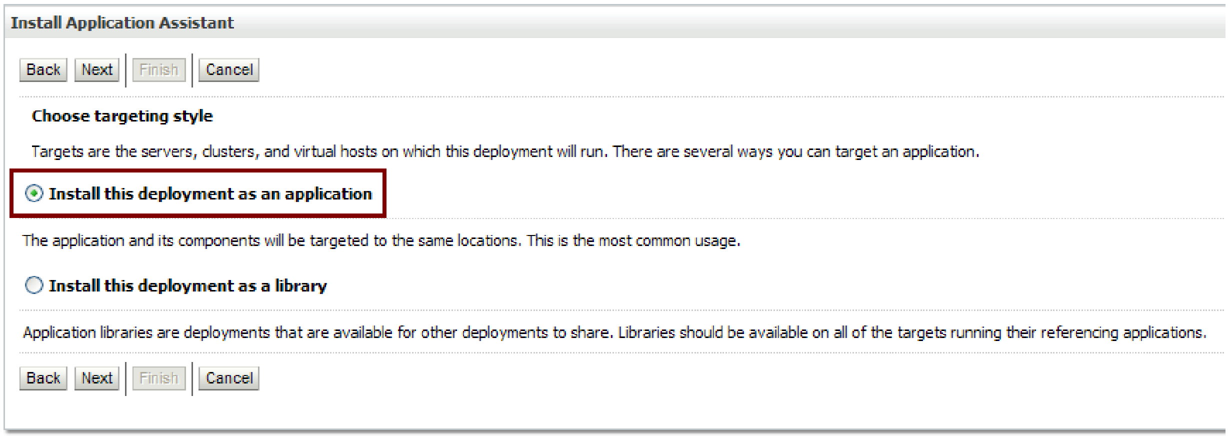
Recently Used Paths: D:\opt\Oracle\Middleware\user_projects\domains\csDomain

Current Location: \D: \opt \Oracle \Middleware \user_projects \domains \csDomain

bin
config
● CS (open directory)
servers

   d. Select the **Install this deployment as an application** option (Figure 6–3), then click **Next**.

Figure 6–3 Installation of This Deployment as an Application



Install Application Assistant

Choose targeting style

Targets are the servers, clusters, and virtual hosts on which this deployment will run. There are several ways you can target an application.

● **Install this deployment as an application**

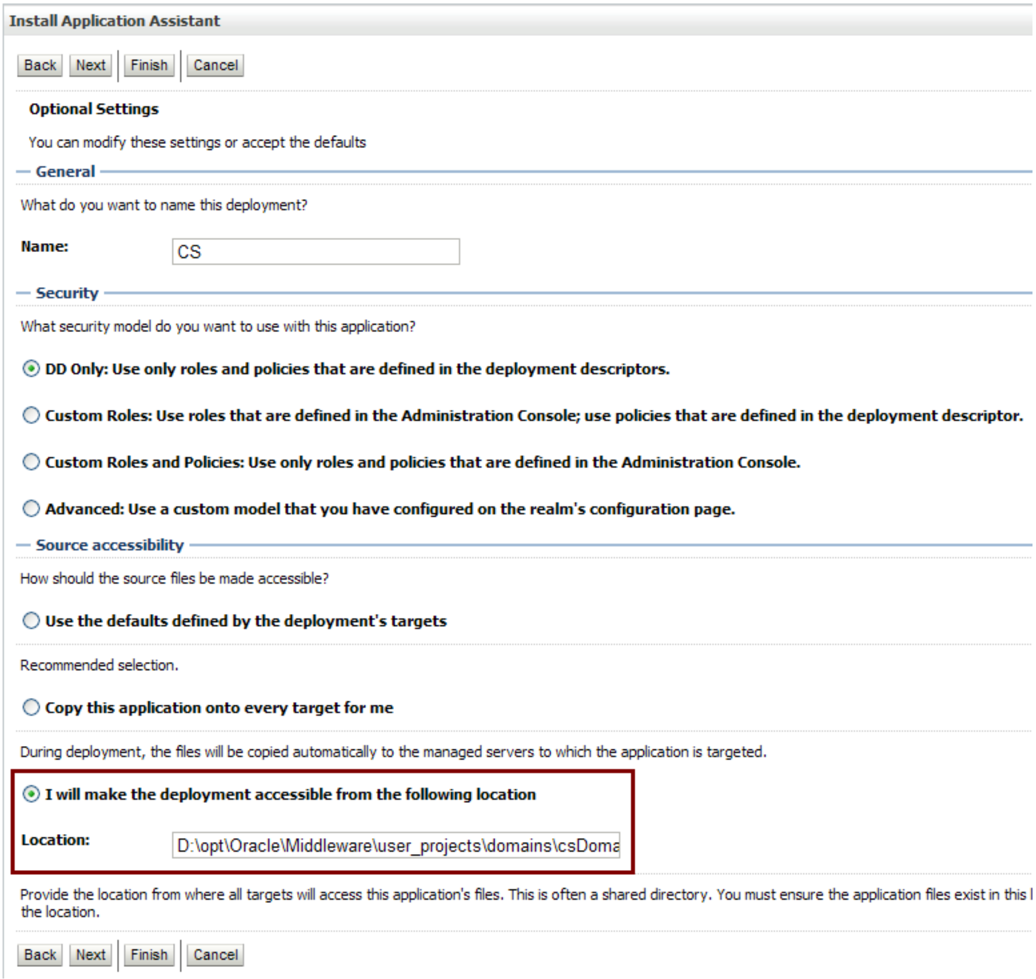
The application and its components will be targeted to the same locations. This is the most common usage.

○ **Install this deployment as a library**

Application libraries are deployments that are available for other deployments to share. Libraries should be available on all of the targets running their referencing applications.

e. In the Source accessibility section, select **I will make the deployment accessible from the following location** (Figure 6–4).

Figure 6–4 Location to Access Deployment



Install Application Assistant

Optional Settings

You can modify these settings or accept the defaults

General

What do you want to name this deployment?

Name: CS

Security

What security model do you want to use with this application?

● **DD Only: Use only roles and policies that are defined in the deployment descriptors.**

○ **Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.**

○ **Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.**

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

Source accessibility

How should the source files be made accessible?

○ **Use the defaults defined by the deployment's targets**

Recommended selection.

○ **Copy this application onto every target for me**

During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

● **I will make the deployment accessible from the following location**

Location: D:\opt\Oracle\Middleware\user_projects\domains\csDoma

Provide the location from where all targets will access this application's files. This is often a shared directory. You must ensure the application files exist in this l the location.

f. Click **Next**, then click **Finish**.

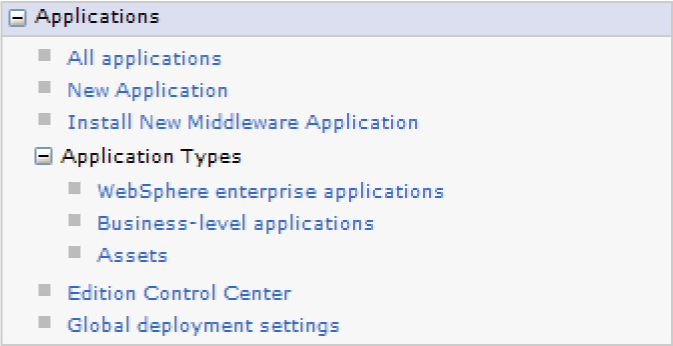
6.5.2 Deploying WebCenter Sites and CAS on WebSphere

This section describes how to deploy the WebCenter Sites and CAS web applications on WebSphere application server from the WebSphere Integrated Solutions administration console. Complete the following steps for each application.

> **Note:** In a cluster environment, deployment of the WebCenter Sites and CAS applications only needs to be done once at the midpoint of the primary server installation.

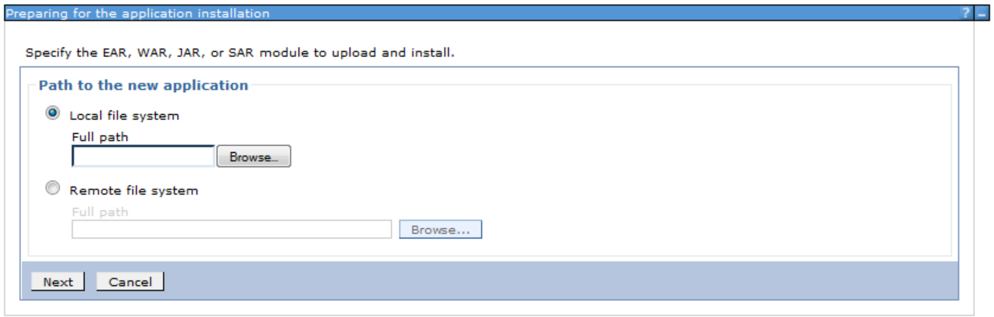
1. Log in to the WebSphere Integrated Solutions administration console.
2. In the tree on the left, expand **Applications**, then **Application Types**, and click **WebSphere enterprise applications** (Figure 6–5).

Figure 6–5 WebSphere Enterprise Applications


Applications
All applications
New Application
Install New Middleware Application
Application Types
WebSphere enterprise applications
Business-level applications
Assets
Edition Control Center
Global deployment settings

3. Click **Install**.
4. On the Preparing for the application installation screen select **Remote file system** and click **Browse** (Figure 6–6).

Figure 6–6 Remote File System Field


Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

Full path

○ Remote file system

Full path

5. On the Browse Remote Filesystems screen, do the following:
   a. Click the application server node where the `.ear` file is located.
   b. For WebCenter Sites:

      Browse to the `<sites_install_dir>/ominstallinfo/app` directory.

      For CAS:

Browse to the `<sites_install_dir>/ominstallinfo` directory.

c. For WebCenter Sites:

   Select the `ContentServer.ear` file.

   For CAS:

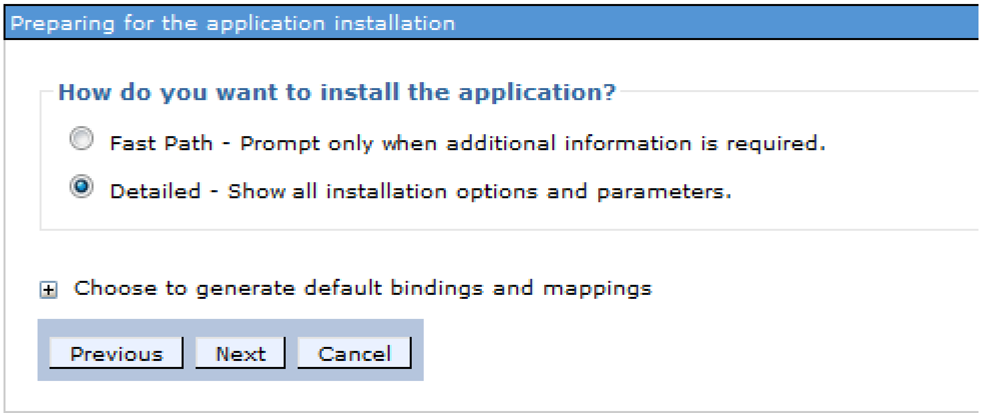
   Select the `cas.ear` file.

d. Click **OK**.

e. Click **Next**.

6. On the Preparing for the application installation screen, select **Detailed** and click **Next** (Figure 6–7).

Figure 6–7 Detailed Option



Preparing for the application installation

How do you want to install the application?

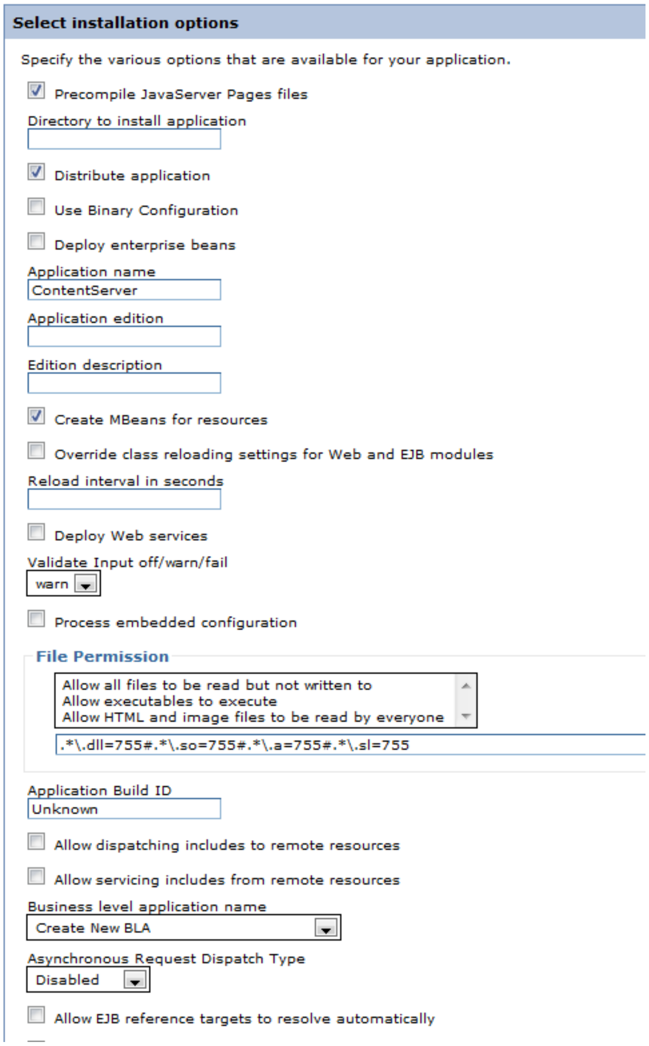
○ Fast Path - Prompt only when additional information is required.

● Detailed - Show all installation options and parameters.

7. On the Application Security Warnings screen, click **Continue**.

8. On the Select installation options screen, select **Precompile JavaServer Pages** files and click **Next** (Figure 6–8).

Figure 6–8 Select Installation Options



Select installation options

Specify the various options that are available for your application.

☑ Precompile JavaServer Pages files

Directory to install application:

☑ Distribute application

□ Use Binary Configuration

□ Deploy enterprise beans

Application name: ContentServer

Application edition:

Edition description:

☑ Create MBeans for resources

□ Override class reloading settings for Web and EJB modules

Reload interval in seconds:

□ Deploy Web services

Validate Input off/warn/fail: warn

□ Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Application Build ID: Unknown

□ Allow dispatching includes to remote resources

□ Allow servicing includes from remote resources

Business level application name: Create New BLA

Asynchronous Request Dispatch Type: Disabled

□ Allow EJB reference targets to resolve automatically

9. On the Map modules to servers screen (Figure 6–9), do the following:

   a. Select the checkbox for the `.war` file module.

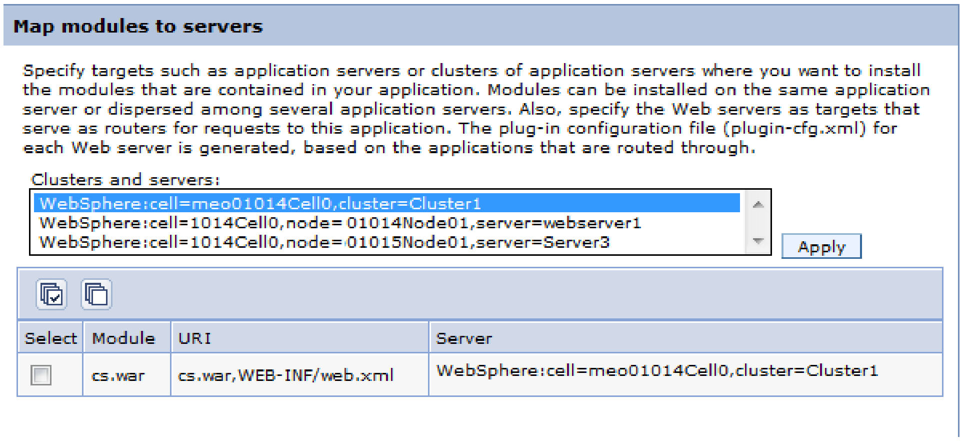
   b. Select the deployment target under **Clusters and servers**.

      In a cluster environment, select the cluster.

   c. Click **Apply**.

   d. Click **Next**.

Figure 6–9 Clusters and Servers



Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

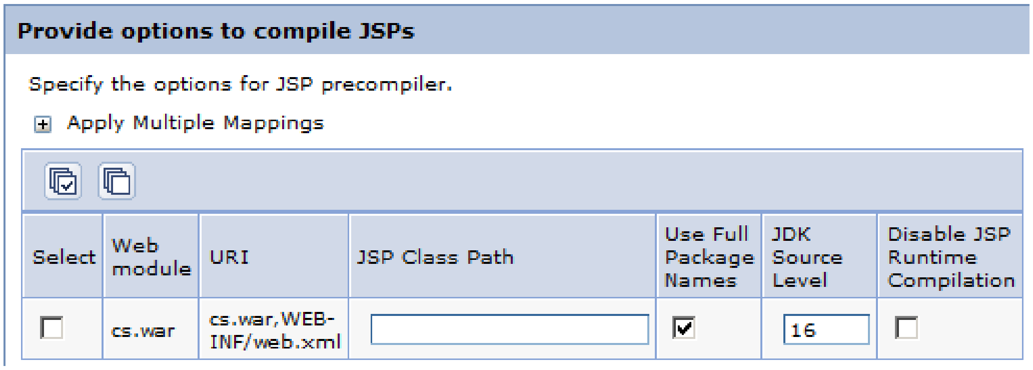
Clusters and servers:

- WebSphere:cell=meo01014Cell0,cluster=Cluster1
- WebSphere:cell=1014Cell0,node=01014Node01,server=webserver1
- WebSphere:cell=1014Cell0,node=01015Node01,server=Server3

Select	Module	URI	Server
□	cs.war	cs.war,WEB-INF/web.xml	WebSphere:cell=meo01014Cell0,cluster=Cluster1

10. On the Provide options to compile JSPs screen (Figure 6–10), change the value of the JDK Source Level field to `16`, then click **Next**.

Figure 6–10 Provide Options to Compile JSPs



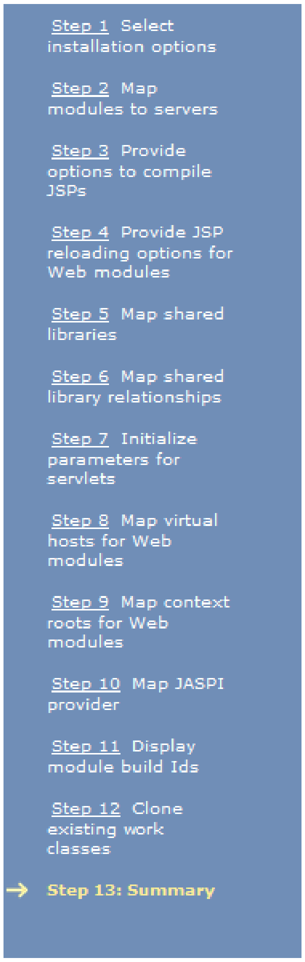
Provide options to compile JSPs

Specify the options for JSP precompiler.

Select	Web module	URI	JSP Class Path	Use Full Package Names	JDK Source Level	Disable JSP Runtime Compilation
□	cs.war	cs.war,WEB-INF/web.xml		☑	16	□

11. On the Provide JSP reloading options for Web modules screen, on the left side, click the link for **Step 13: Summary** (Figure 6–11).

Figure 6–11 Step 13: Summary



Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Provide options to compile JSPs

Step 4 Provide JSP reloading options for Web modules

Step 5 Map shared libraries

Step 6 Map shared library relationships

Step 7 Initialize parameters for servlets

Step 8 Map virtual hosts for Web modules

Step 9 Map context roots for Web modules

Step 10 Map JASPI provider

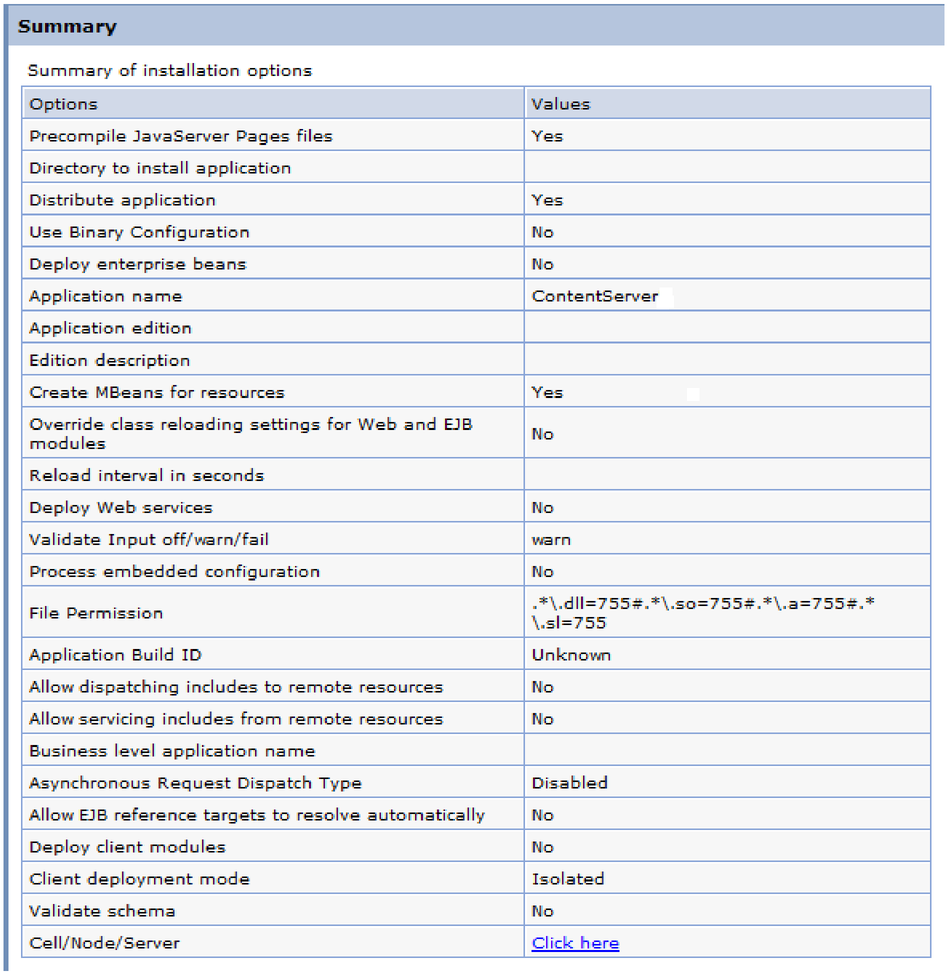
Step 11 Display module build Ids

Step 12 Clone existing work classes

→ **Step 13: Summary**

12. On the Summary screen (Figure 6–12), review the options you have chosen and click **Finish**.

Figure 6–12 Summary of Installation Options



Summary

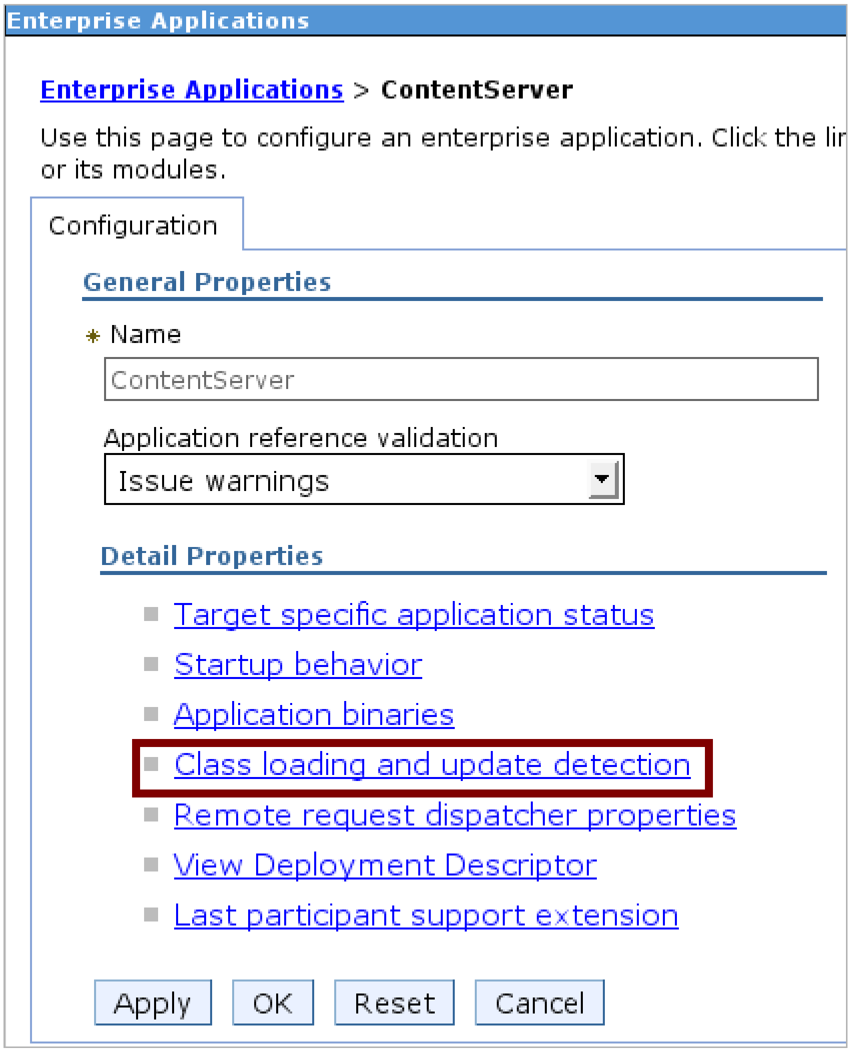
Summary of installation options

Options	Values
Precompile JavaServer Pages files	Yes
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	ContentServer
Application edition	
Edition description	
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Deploy client modules	No
Client deployment mode	Isolated
Validate schema	No
Cell/Node/Server	Click here

13. On the Enterprise Applications screen, click the newly deployed application.

14. On the screen that follows, click **Class loading and update detection** (Figure 6–13).

Figure 6–13 Class Uploading and Update Detection



Enterprise Applications

Enterprise Applications > ContentServer

Use this page to configure an enterprise application. Click the lir
or its modules.

Configuration
General Properties

* Name

ContentServer

Application reference validation

Issue warnings

Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
Apply
OK
Reset
Cancel

15. On the screen that opens (Figure 6–14), do the following:

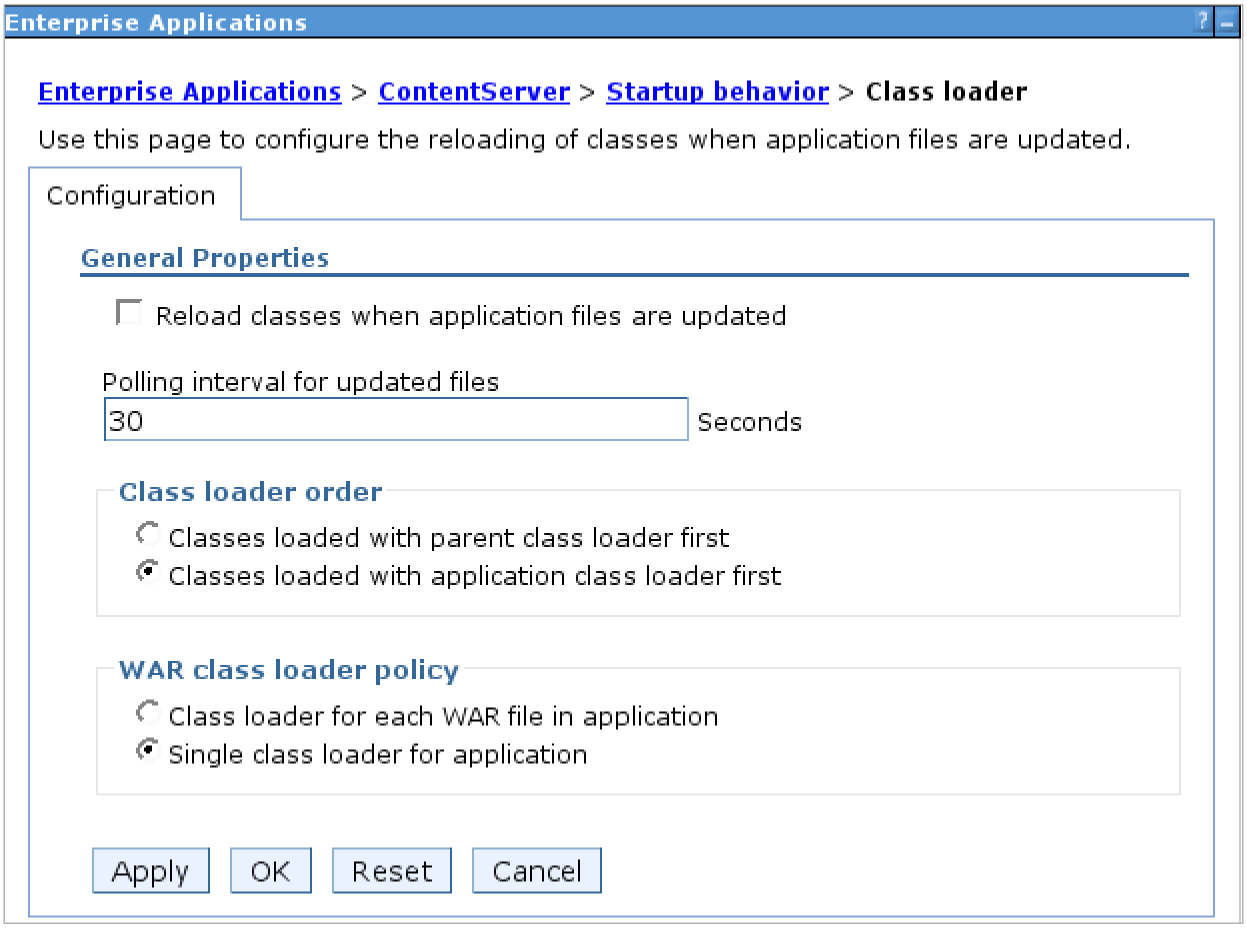
    a. In the **Polling interval for updated files** field, enter `30`.

    b. In the Class load order section, select **Classes loaded with application class loader first**.

    c. In the WAR class loader policy section, select **Single class loader for application**.

    d. Click **OK**.

Figure 6–14 Class Loader


Enterprise Applications

Enterprise Applications > ContentServer > Startup behavior > Class loader

Use this page to configure the reloading of classes when application files are updated.

General Properties

□ Reload classes when application files are updated

Polling interval for updated files
30 Seconds

Class loader order

○ Classes loaded with parent class loader first
● Classes loaded with application class loader first

WAR class loader policy

○ Class loader for each WAR file in application
● Single class loader for application

16. Save the changes.

6.5.3 Deploying WebCenter Sites and CAS on Tomcat

The WebCenter Sites installer automatically explodes the WebCenter Sites web application to the `webapps` directory of the Tomcat server during the primary and secondary server installations.

The WebCenter Sites installer automatically explodes the CAS web application to the `webapps` directory of the primary Tomcat server during the primary server installation. The CAS web application should be copied to any other servers and their respective `webapps` directories by this point.

6.6 Restarting WebCenter Sites

After the installation for an application server has completed, restart the server before verifying it.

6.7 Switching WebLogic Application Server to Production Mode (Delivery Systems Only)

If you have installed a delivery system, switch WebLogic to production mode:

1. Log in to the WebLogic Administration Console.

2. Click the domain name in the tree
3. Click the **General** tab.
4. Select the check box next to **Production Mode**.
5. Click **Save**.
6. Click **Accept Changes**.
7. Restart all servers.
8. You are now ready to verify your WebCenter Sites installation and complete various post-installation steps, if necessary. Continue with Chapter 7, "WebCenter Sites Post-Installation Steps."

7

WebCenter Sites Post-Installation Steps

When the WebCenter Sites installation process completes successfully, perform the following steps as necessary for your environment:

- Section 7.1, "Loading the XML Parser"
- Section 7.2, "Specifying Trusted URLs for CAS"
- Section 7.3, "Verifying the WebCenter Sites Installation"
- Section 7.4, "Integrating with LDAP (Optional)"
- Section 7.5, "Integrating Oracle Access Manager (OAM) with WebCenter Sites (Optional)"
- Section 7.6, "Installing Satellite Servers"
- Section 7.7, "Setting Up WebCenter Sites for Its Business Purpose"

7.1 Loading the XML Parser

WebCenter Sites contains a modified version of the Microsoft XML Parser (`MSXML.jar` in the `WEB-INF/lib` directory). If a different version of the Parser is referenced in the `CLASSPATH` environment variable, you must change the path to refer to the version used by WebCenter Sites; otherwise, WebCenter Sites will fail when parsing XML.

7.2 Specifying Trusted URLs for CAS

On successful login, CAS redirects the user to the requested service. For security, a requested service can be validated prior to CAS redirecting the user to the service. To validate requested services, specify the list of trusted (permitted) URLs in the `cs_install/bin/customBeans.xml` file, and restart the CAS server. The URLs can be in either one of the following formats:

- Exact URL: For example,

  ```
  http://<hostname>:<port>/cs/wem/fatwire/wem/Welcome
  ```

- Matching URLs: For example,

  ```
  http://<hostname>:<port>/SitesWebapp/*
  ```

  where `/*` indicates that all URLs beginning with `http://<hostname>:<port>/SitesWebapp` are permitted.

If the user needs to access trusted resources using IP addresses, specify their URLs, as well.

7.3 Verifying the WebCenter Sites Installation

Verify the installation by logging in to Oracle WebCenter Sites as a general administrator and accessing the WebCenter Sites Admin, Contributor, and WEM Admin interfaces. In the process, you will verify that single sign-on functions (by accessing a different application without logging out and back in to WebCenter Sites).

To access the WebCenter Sites interfaces

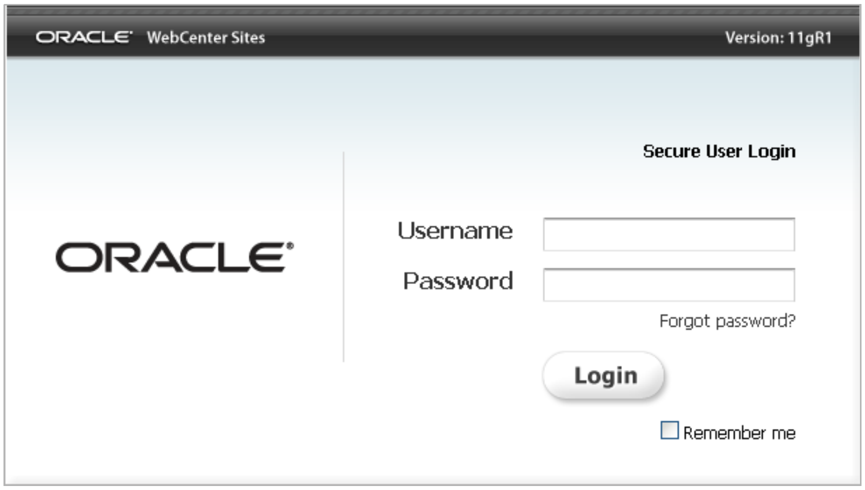
1. Point your browser to the following URL:

   ```
   http://<server>:<port>/<context>/login
   ```

   where `<server>` is the host name or IP address of the server running WebCenter Sites, `<port>` is the port number of the WebCenter Sites application, and `<context>` is the name of the WebCenter Sites application that was deployed on the server.

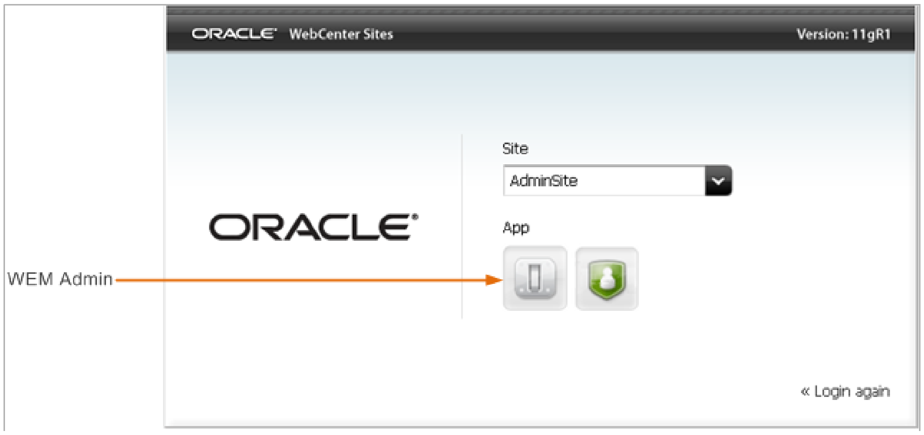
   The WebCenter Sites login form is displayed (Figure 7–1).

Figure 7–1 WebCenter Sites Login Form



2. Enter the following credentials:
   - User name: `fwadmin`
   - Password: `xceladmin`
3. Click **Login**.
4. Access the WEM Admin interface. In the **Site** drop-down list, select **AdminSite** and the icon for the **WEM Admin** interface (Figure 7–2).

Figure 7–2 AdminSite Option and WEM Admin Icon

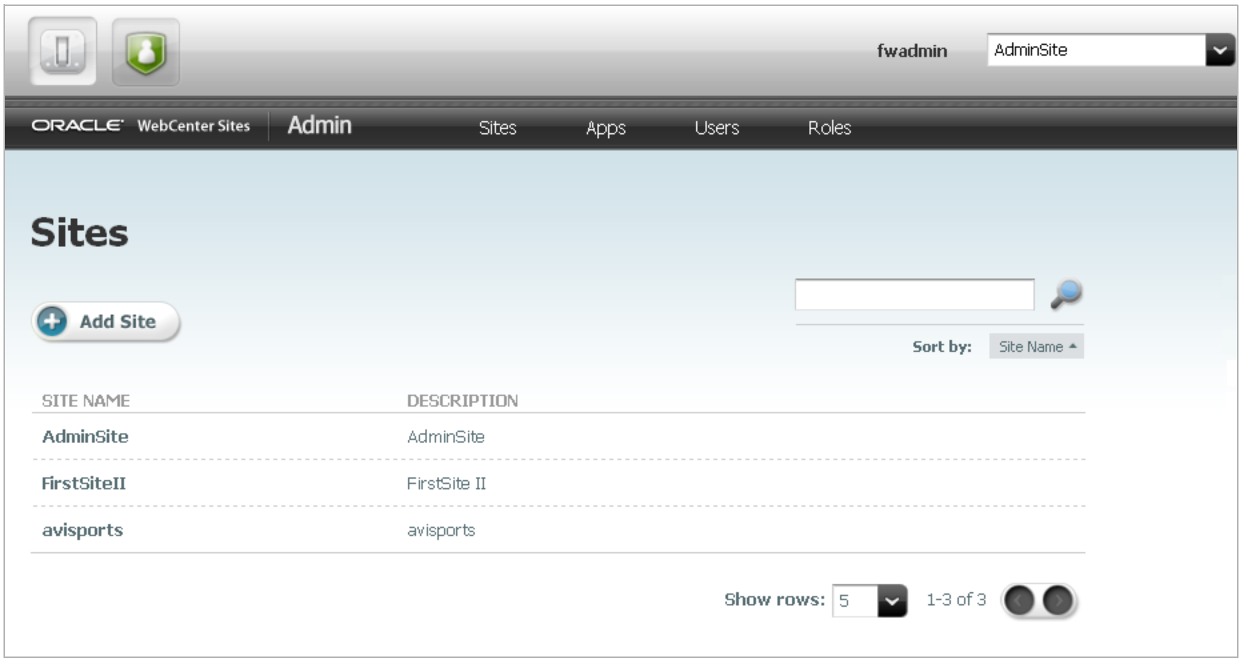


Site: AdminSite

App

The WEM Admin interface (on AdminSite) is displayed (Figure 7–3)

Figure 7–3 WEM Admin Interface



AdminSite

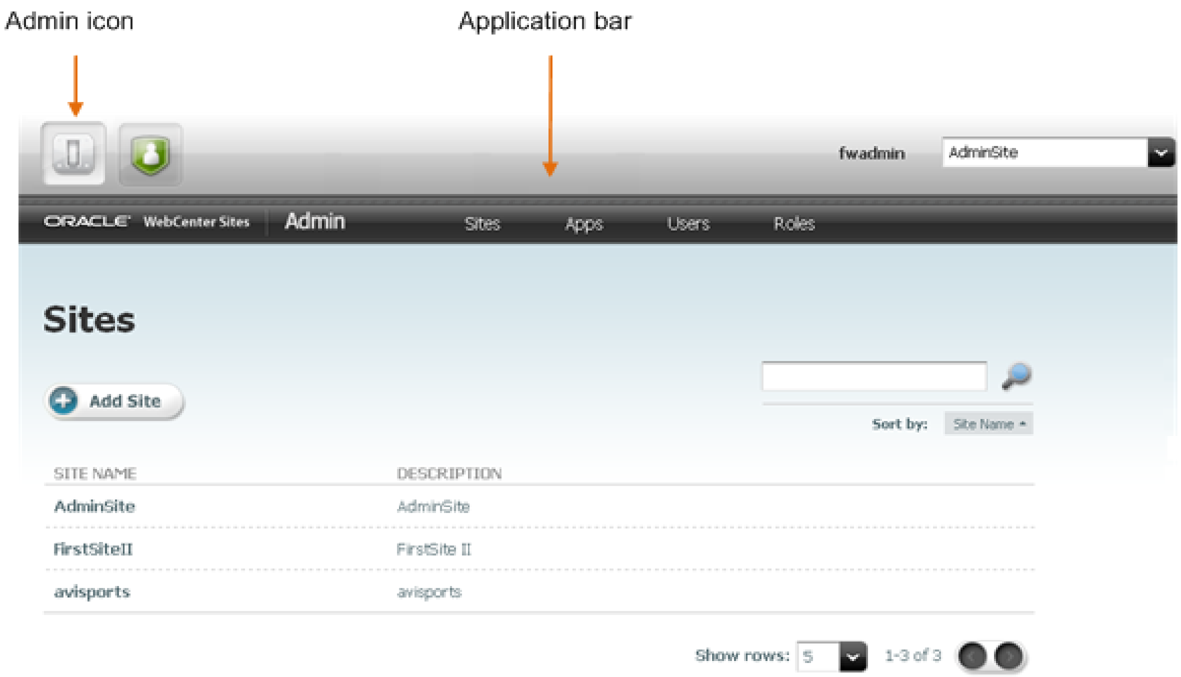
Sort by: Site Name

SITE NAME	DESCRIPTION
AdminSite	AdminSite
FirstSiteII	FirstSite II
avisports	avisports

Show rows: 5 1-3 of 3

5. Access the WebCenter Sites Admin interface by navigating to the applications bar (at the top of the interface) and selecting the icon for the WebCenter Sites **Admin** interface (Figure 7–4).

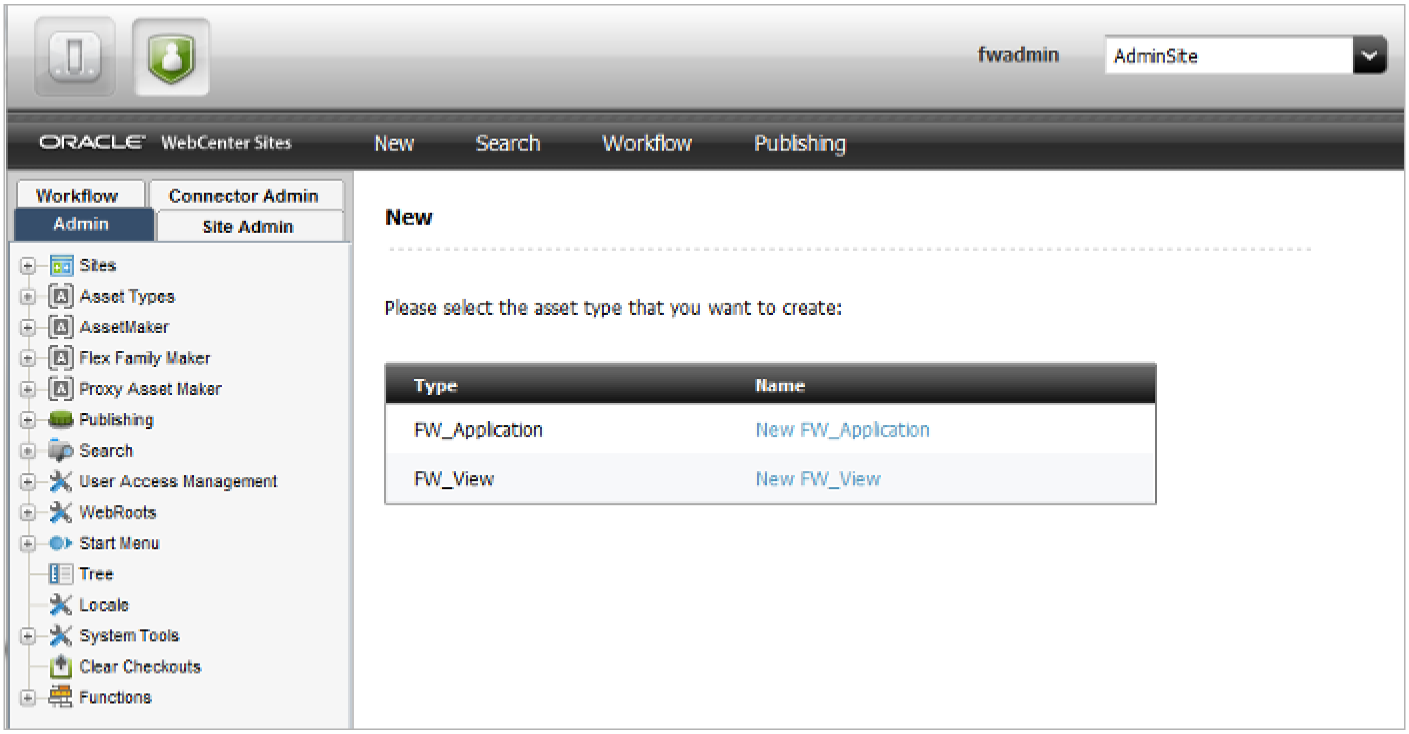
Figure 7–4 WEM Admin Icon in the Application Bar



SITE NAME	DESCRIPTION
AdminSite	AdminSite
FirstSiteII	FirstSite II
avisports	avisports

The WebCenter Sites Admin interface (on AdminSite) is displayed (Figure 7–5). Only system administration functionality is available.

Figure 7–5 Administration Functionality on AdminSite



New

Please select the asset type that you want to create:

Type	Name
FW_Application	New FW_Application
FW_View	New FW_View

6. Access the WebCenter Sites Contributor interface:

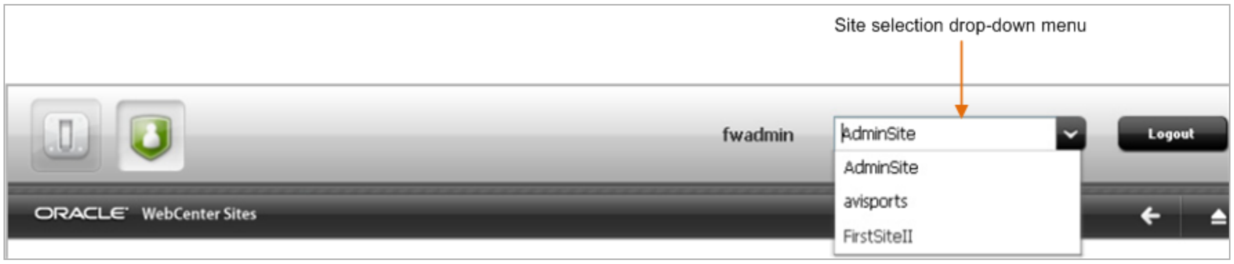
Note: The WebCenter Sites Contributor interface is not associated with AdminSite. Start with one of the following steps:

- If the avisports or FirstSite II sample site is installed, the Contributor interface is assigned to each site. Continue to step 6a and select either one of these sample sites to access the Contributor interface.
- If no sample sites are installed, you will have to open the WEM Admin interface (see step 4), create a site, create users, and then assign the users to the site. The Contributor interface is automatically assigned to the same site. For instructions on assigning users to a site from the WEM Admin interface, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* and then continue with this step.

---

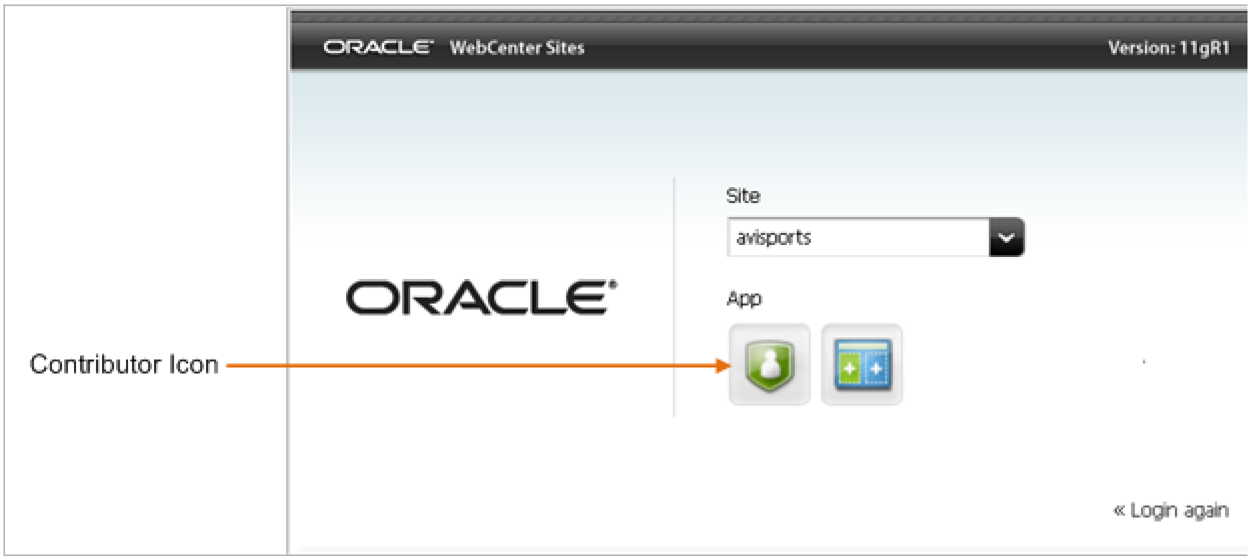
a. Navigate to the applications bar. In the site selection drop-down list (Figure 7–6), select a site other than AdminSite.

Figure 7–6 Application Bar Navigation



b. Since this is the first time you are accessing the selected site, the screen shown in Figure 7–7 is displayed. Select the icon for the Contributor interface.

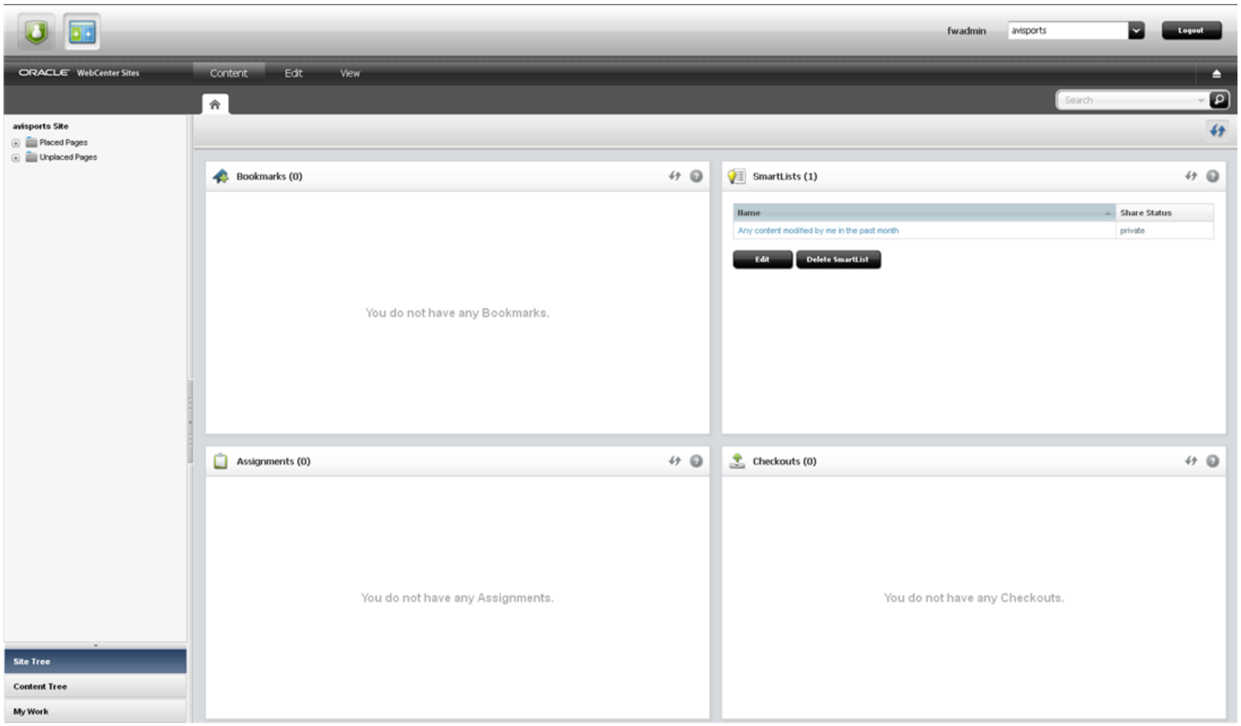
Figure 7–7 Contributor Interface Icon



(On subsequent access, WebCenter Sites opens the application you last accessed for the site you selected.)

The Contributor interface on your selected site is displayed (Figure 7–8).

Figure 7–8 Contributor Interface



Name	Share Status
Any content modified by me in the past month	private

You do not have any Bookmarks.

You do not have any Assignments.

You do not have any Checkouts.

WebCenter Sites is now ready to be configured. Follow the steps in the rest of this chapter.

7.4 Integrating with LDAP (Optional)

LDAP integration is optional for web installations.

If you need to perform LDAP integration, do the following:

1. Set up a supported LDAP server of your choice. For instructions, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.
2. Run the LDAP integration program included on the WebCenter Sites CD.

   For more information, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

   > **Note:** If you have configured LDAP for the primary cluster member, make sure to run the `configuredLDAP.sh` for all secondary cluster members.

3. Test the integration:

   ```
   http://<servername>:<port>/servlet/CatalogManager?ftcmd
   =login&username=<username>&password=<password>
   ```

where:

`<servername>` is the name of the machine that is hosting WebCenter Sites

`<port>` is the port number of that server

`<username>` is user-defined (`ContentServer`, for example)

`<password>` is user-defined (`password`, for example)

7.5 Integrating Oracle Access Manager (OAM) with WebCenter Sites (Optional)

If you wish to replace CAS with Oracle Access Manager (OAM), refer to instructions in *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

7.6 Installing Satellite Servers

For information, see Chapter 9, "Procedures for Installing Remote Satellite Servers."

7.7 Setting Up WebCenter Sites for Its Business Purpose

Once you have completed your WebCenter Sites installation, you are ready to configure it for business use. For instructions, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*. The guides explain how to create and enable a content management environment including the data model, content management sites, site users, publishing functions, and client interfaces.

Part II

Installing Oracle WebCenter Sites: Satellite Server

Part II contains the following chapters:

8

Overview of Satellite Server

Satellite Server works with your WebCenter Sites system to provide the following benefits:

- An additional layer of caching, supplementing the layer of caching that is provided by the WebCenter Sites cache.
- Scalability. You can quickly and economically scale your WebCenter Sites system by adding remote installations of Satellite Server.
- Improved performance. Satellite Server improves your website's performance by reducing the load on WebCenter Sites and moving content closer to the website visitors who will view it.
- The ability to cache REST calls. For this reason you may wish to add a remote Satellite Server in front of a content management installation as well as a delivery installation.

This chapter introduces you to the configurations that you implement in order to receive these benefits.

You can configure Satellite Server in the following ways:

- Co-Resident, which provides a second layer of caching and allows to simulate actual delivery of content on development and management systems.
- Remote, which improves performance and scalability on delivery systems.

This document contains the following sections:

- Section 8.1, "Co-Resident"
- Section 8.2, "Remote"

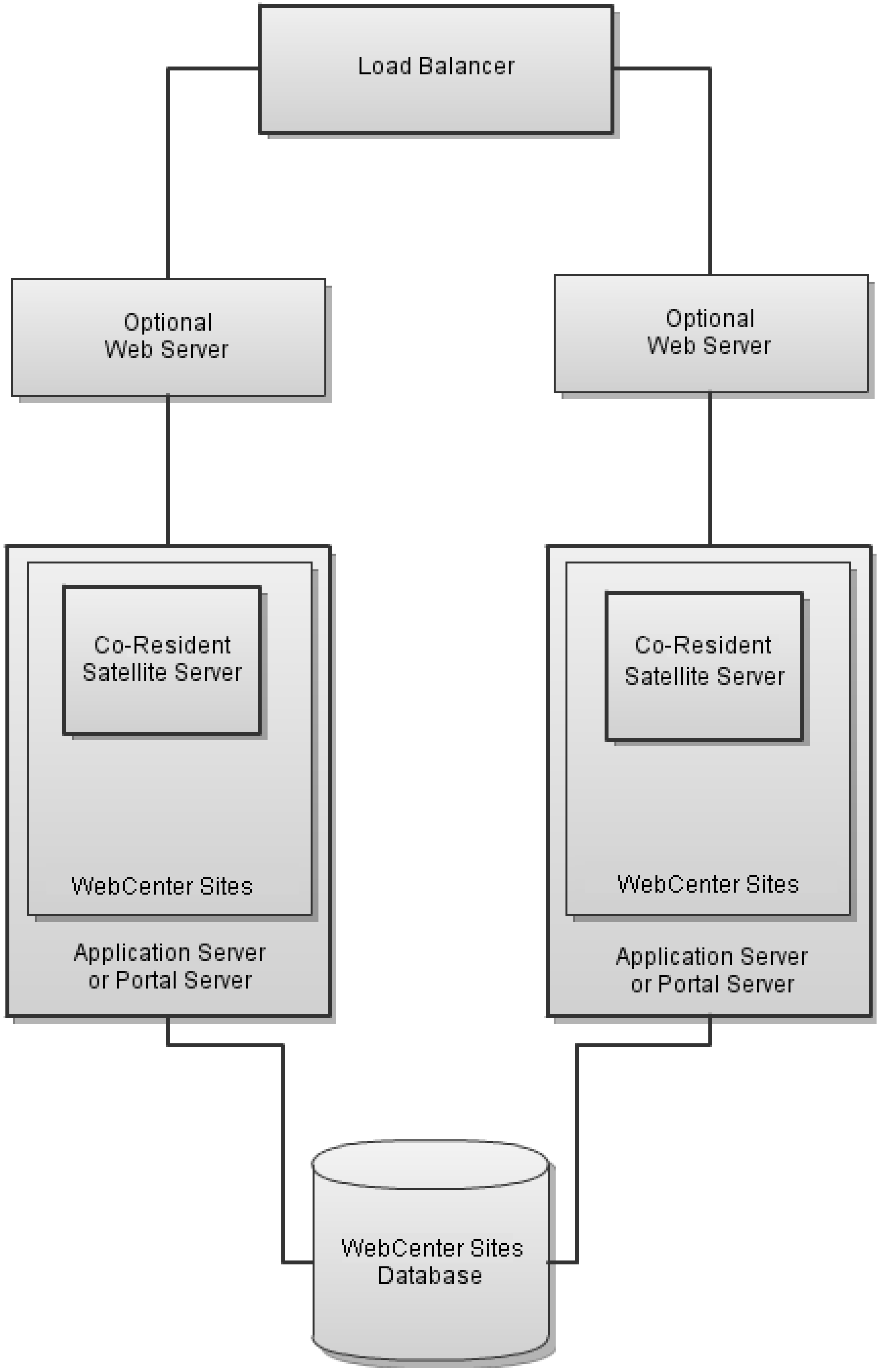
8.1 Co-Resident

WebCenter Sites ships with a copy of Satellite Server that is automatically installed and enabled on the same machine as your WebCenter Sites software. This is your co-resident Satellite Server. The purpose of the co-resident Satellite Server is to provide development and management systems with the ability to simulate page delivery as it occurs on the active site (delivery system).

Co-resident Satellite Server provides a layer of caching in addition to that provided by the WebCenter Sites cache. Satellite Server and the WebCenter Sites caches work in tandem to provide double-buffered caching, where copies of cached pages are stored in both the Satellite Server and the WebCenter Sites caches. The disadvantage of this configuration is increased memory utilization, as there are now two copies of each object stored in memory. For more information about double-buffered caching, see the caching chapter of the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*. Figure 8–1 illustrates a co-resident installation of Satellite Server.

We recommend setting your co-resident Satellite Server host to optimize the performance of your WebCenter Sites system, as described in Section 10.1, "Adjusting Caching Conditions."

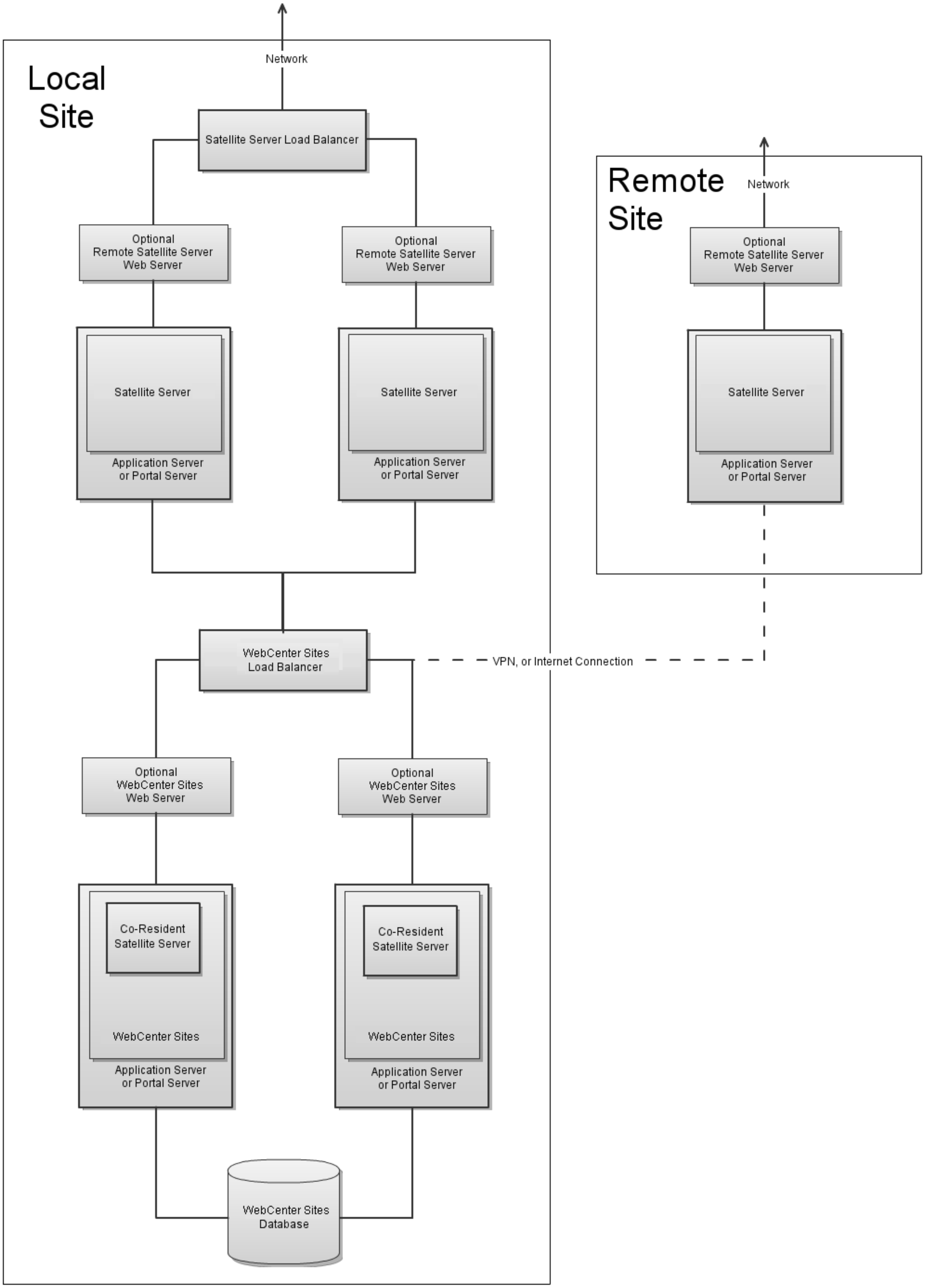
Figure 8–1 Co-Resident Satellite Server Architecture


Load Balancer
Optional Web Server
Optional Web Server
Co-Resident Satellite Server
WebCenter Sites
Application Server or Portal Server
Co-Resident Satellite Server
WebCenter Sites
Application Server or Portal Server
WebCenter Sites Database

8.2 Remote

On delivery systems, you should set up one or more remote Satellite Server instances. At a minimum, a remote Satellite Server should be installed at the same geographic location as WebCenter Sites. For additional performance at remote geographic locations, additional remote Satellite Servers can be configured, as shown in Figure 8–2.

Figure 8–2 Remote Satellite Server Architecture


Local Site
Network
Satellite Server Load Balancer
Optional Remote Satellite Server Web Server
Optional Remote Satellite Server Web Server
Satellite Server
Application Server or Portal Server
Satellite Server
Application Server or Portal Server
WebCenter Sites Load Balancer
VPN, or Internet Connection
Optional WebCenter Sites Web Server
Optional WebCenter Sites Web Server
Co-Resident Satellite Server
WebCenter Sites
Application Server or Portal Server
Co-Resident Satellite Server
WebCenter Sites
Application Server or Portal Server
WebCenter Sites Database
Remote Site
Network
Optional Remote Satellite Server Web Server
Satellite Server
Application Server or Portal Server

Remote installations of Satellite Server provide several benefits in addition to allowing double-buffered caching:

- They improve the performance of the website by moving the content closer to its audience. In the preceding diagram, for instance, the main data center is located in New York City, while the secondary data centers are located in Europe and Asia.
- They allow for improved scaling, as cached content does not require requests to be forwarded to WebCenter Sites and therefore free this resource to handle other tasks.

For information on installing and configuring remote instances of Satellite Server, see Section 9, "Procedures for Installing Remote Satellite Servers."

9 Procedures for Installing Remote Satellite Servers

When you install remote instances of Satellite Server, you can install them on any supported application server (see the application matrix). Note that installing and configuring remote instances of Satellite Server is an iterative process. You must initially install, configure, and test one remote Satellite Server, then install, configure, and test your other remote Satellite Server installations. After you have completed the installation and initial configuration of your Satellite Server software, you can adjust the configurations to make maximum use of the hardware and software stack.

To install and configure remote instances of Satellite Server, complete the following steps:

- Section 9.1, "Step 1: Installing Required Hardware and Software"
- Section 9.2, "Step 2: Expanding the Installation File"
- Section 9.3, "Step 3: Installing Satellite Server"
- Section 9.4, "Step 4: Registering Satellite Server with WebCenter Sites"
- Section 9.5, "Step 5: Specifying Trusted URLs for CAS"
- Section 9.6, "Step 6: Starting Satellite Server"
- Section 9.7, "Step 7: Testing the Configuration"
- Section 9.8, "Step 8: Installing Satellite Server on Additional Remote Servers"
- Section 9.9, "Next Step"

9.1 Step 1: Installing Required Hardware and Software

Before you install Satellite Server, ensure you have the required hardware and software. For the latest information, refer to the Oracle WebCenter Sites Certification Matrix.

This section contains the following topics:

- Section 9.1.1, "Network Considerations"
- Section 9.1.2, "Load Balancer Requirements"
- Section 9.1.3, "Configuration Requirements"
- Section 9.1.4, "Application Server Requirements"
- Section 9.1.5, "Satellite Server Contents"

9.1.1 Network Considerations

The connection between the Satellite Server hosts and the WebCenter Sites host is a limiting factor for performance of serving uncached data. Remote connections will suffer a performance penalty related to the bandwidth and latency when accessing data not present in the Satellite Server cache

9.1.2 Load Balancer Requirements

You must have a load balancer (hardware or software based) if using more than one remote Satellite Server. Oracle does not require a particular brand of load balancer, but we do strongly recommend using a load balancer that supports session affinity and the session affinity features are enabled.

9.1.3 Configuration Requirements

Your Satellite Server hosts must meet the minimum requirements per Table 9–1.

Table 9–1 Satellite Server Minimum Requirements

Requirements	Description
Operating System	See the Oracle WebCenter Sites Certification Matrix.
Application Server	See the Oracle WebCenter Sites Certification Matrix.
CPU	Dual Core System (recommended: Quad Core system or greater)
Physical Memory	4GB (12GB or more recommended)
Disk Space	2GB (4GB or more recommended)
JVM	32 bit (64 bit recommended)

9.1.4 Application Server Requirements

inCache, the default caching framework used in WebCenter Sites requires the following JVM parameters to be set in order to work efficiently:

```
-Dnet.sf.ehcache.enableShutdownHook=true
-DnumOfDiskStores=<will vary based on drive speed>
```

9.1.5 Satellite Server Contents

Satellite Server needs a full-featured servlet container and a servlet engine. See the *Oracle WebCenter Sites Certification Matrix* for supported environments.

9.2 Step 2: Expanding the Installation File

The installation file is named `SatelliteServer.zip`. Extract this file to a host machine.

- UNIX: Unzip `SatelliteServer.zip`
- Windows: Double click **SatelliteServer.zip** in Windows Explorer

Extracting the zip file creates a subdirectory named `SatelliteServer`. Do not change the name of any of its subdirectories, and retain the archived directory structure; otherwise the installer will fail.

9.3 Step 3: Installing Satellite Server

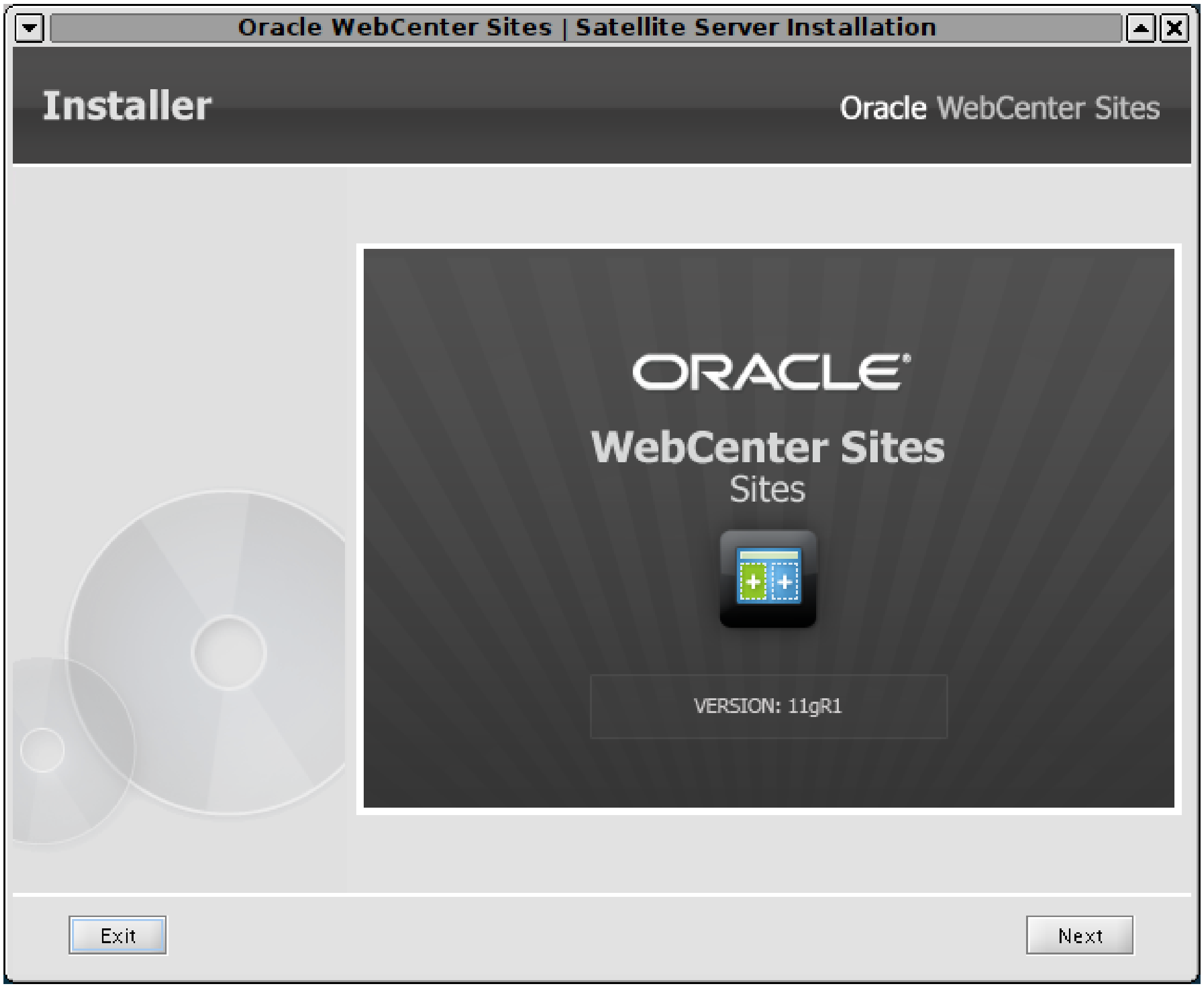
You can install Satellite Server in the following ways:

- Section 9.3.1, "Installing Satellite Server Using GUI Installer"
- Section 9.3.2, "Installing Satellite Server Silently"

9.3.1 Installing Satellite Server Using GUI Installer

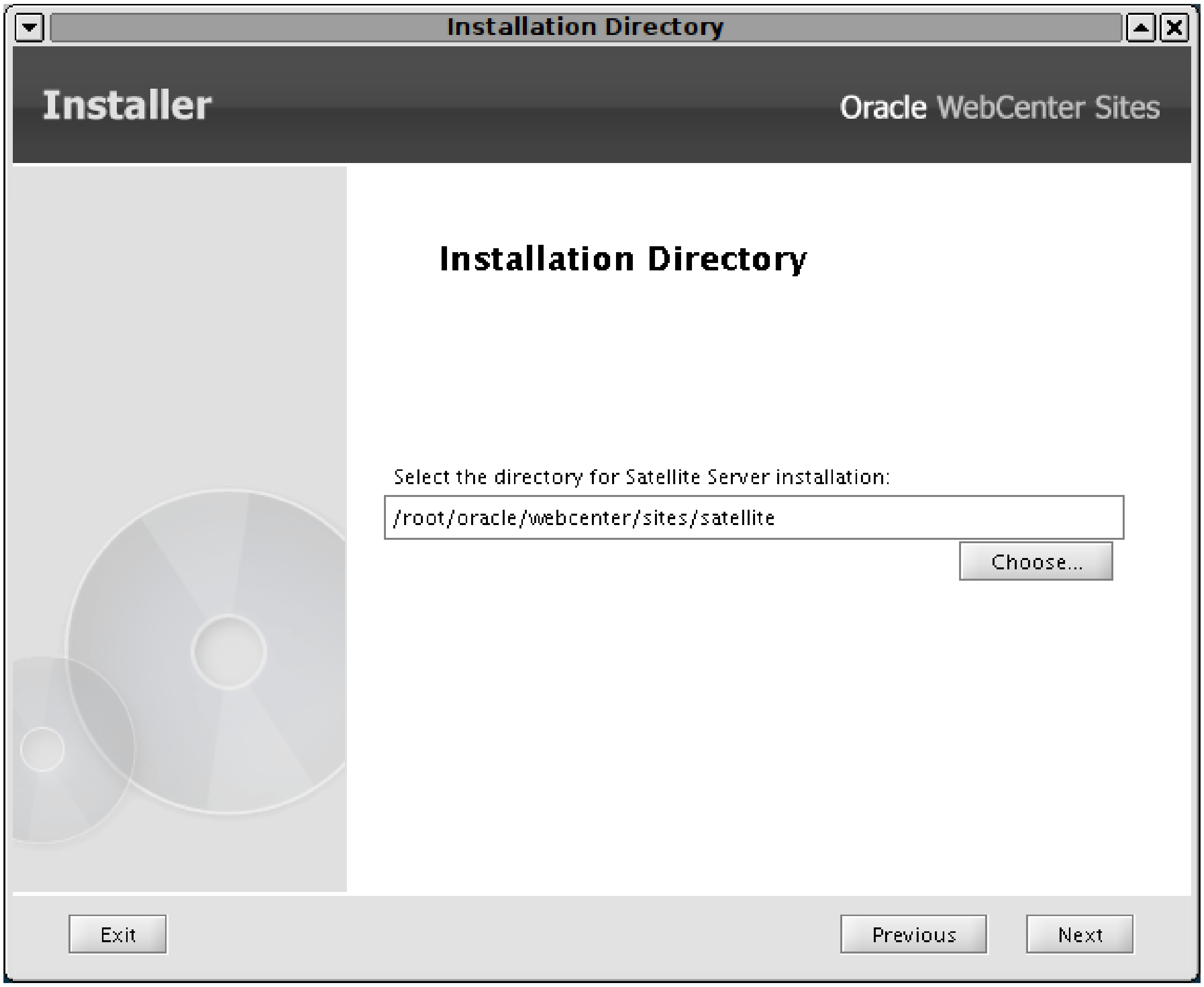
1. Run the installer script:
   - On Windows: `ssInstall.bat`
   - On UNIX based systems: `ssInstall.sh`
2. On the Welcome screen (Figure 9–1), click **Next**.

Figure 9–1 Welcome Screen



3. In the Installation Directory screen (Figure 9–2), enter the target installation path for Satellite Server. Ensure that you have the required permissions. If the directory you specify does not exist, the installer will prompt you for permission to create it.

Figure 9–2 Installation Directory



Installation Directory

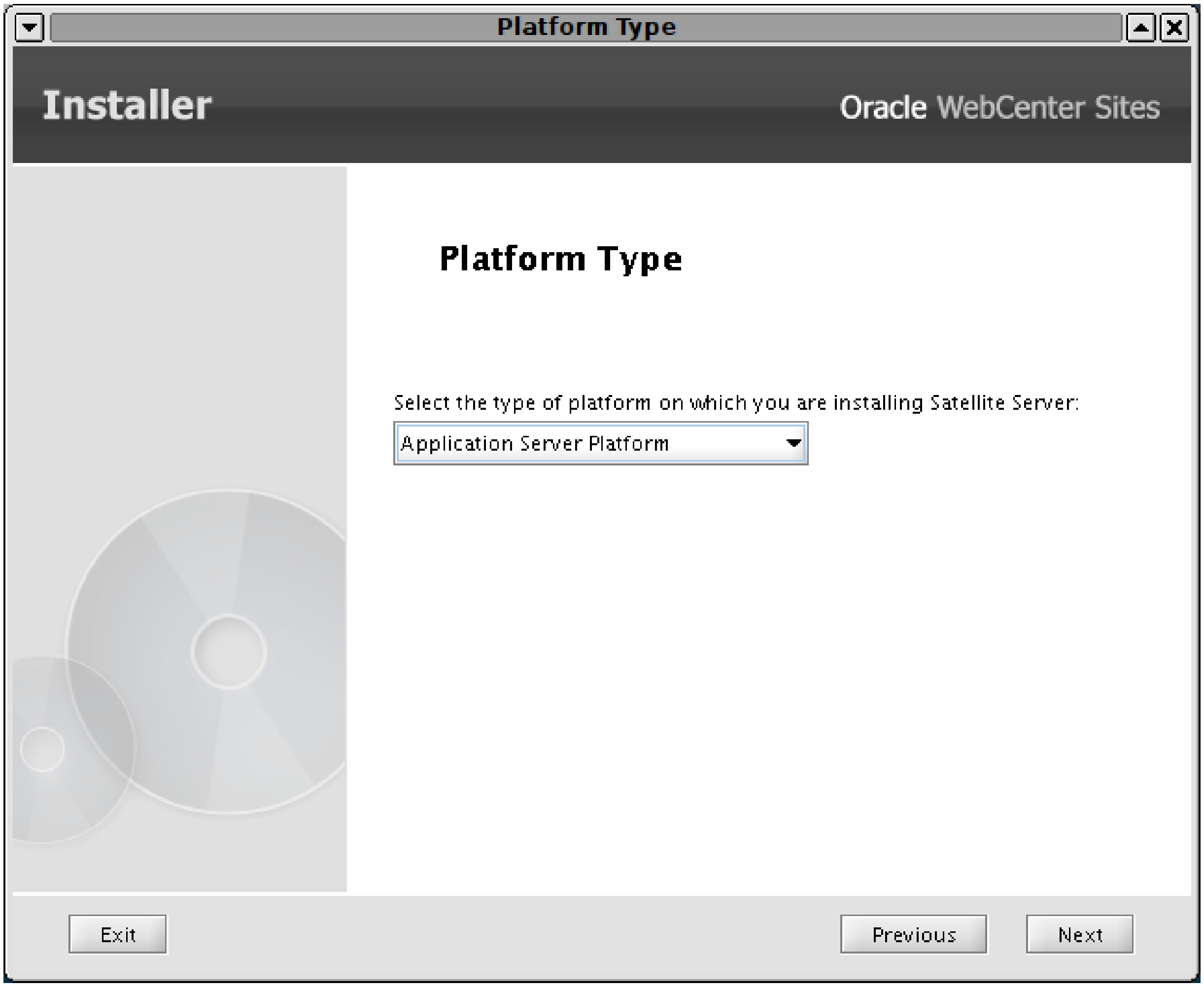
Select the directory for Satellite Server installation:

/root/oracle/webcenter/sites/satellite

Click **Next**.

4. Select the desired platform type (Figure 9–3).

Figure 9–3 Platform Type



Platform Type

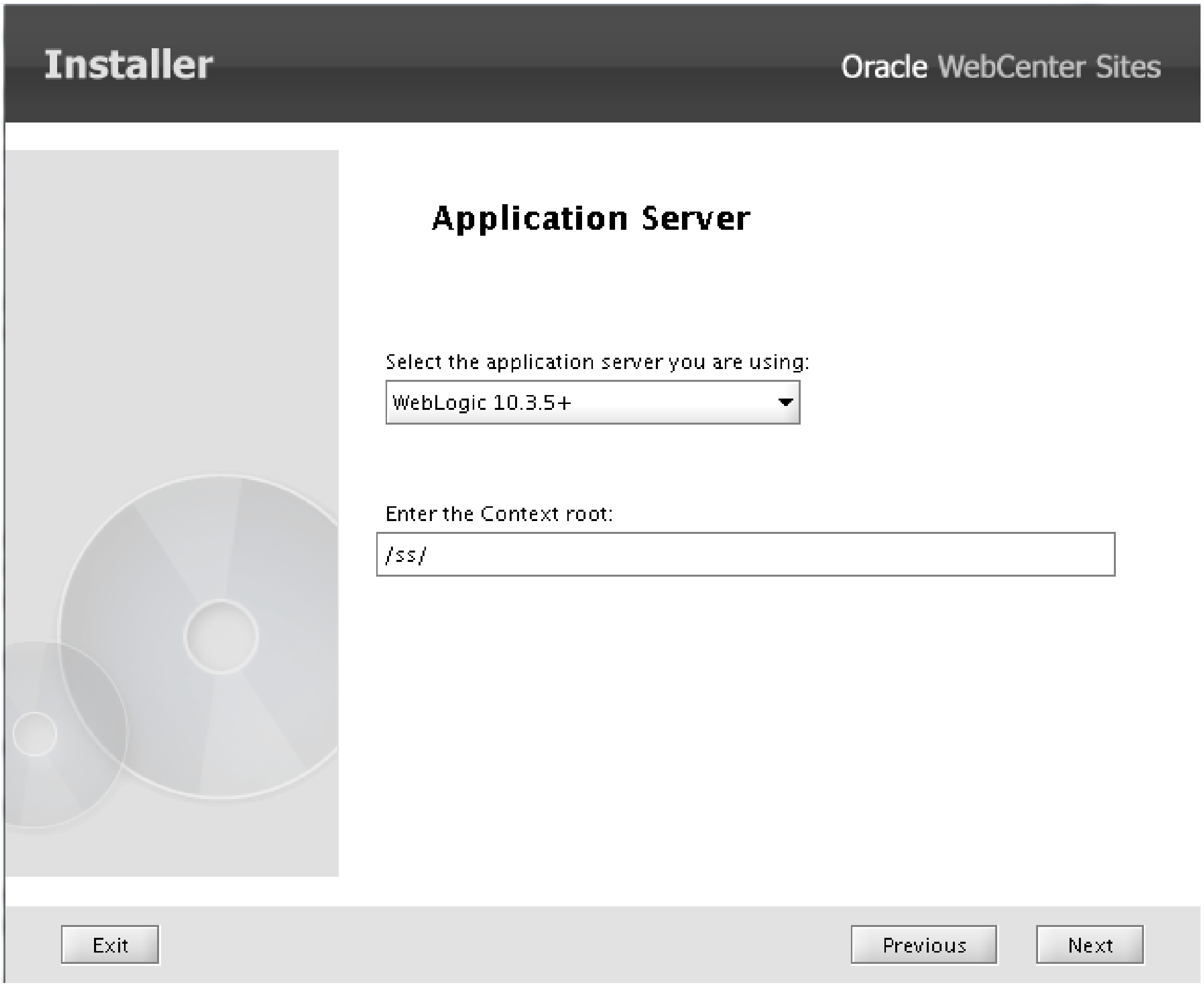
Select the type of platform on which you are installing Satellite Server:

Application Server Platform

Click **Next**.

5. Select the desired application server and enter the Satellite Server context root (Figure 9–4).

Figure 9–4 Application Server



Application Server

Select the application server you are using:

WebLogic 10.3.5+

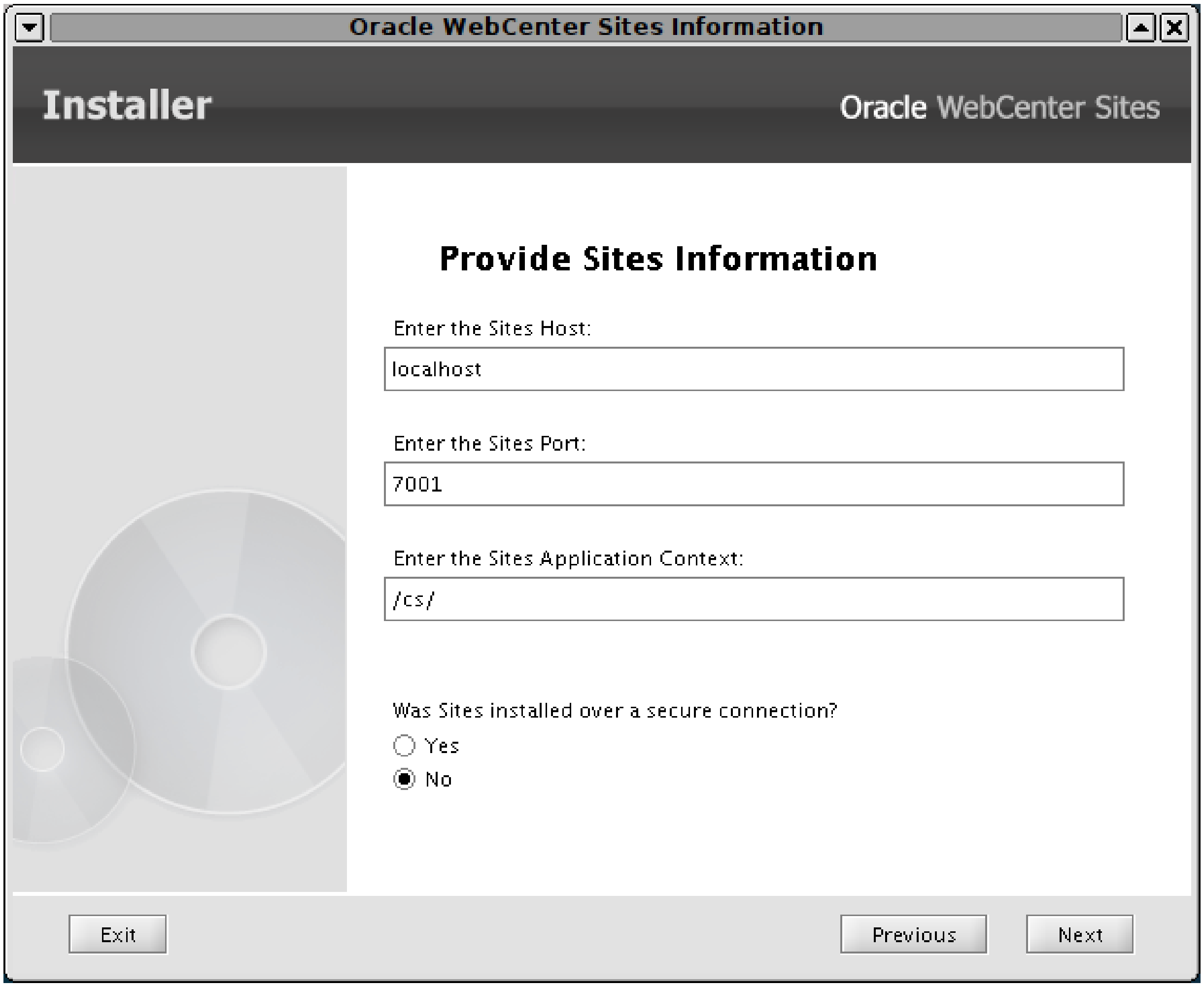
Enter the Context root:

/ss/

Click **Next**.

6. Provide WebCenter Sites information (Figure 9–5).
   - Host name or IP address of the machine running WebCenter Sites
   - Port number that WebCenter Sites is listening for connections
   - Application context root that Satellite Server will connect to
   - Whether WebCenter Sites was installed over a secure connection

Figure 9–5 Sites Information



Oracle WebCenter Sites Information

Installer

Oracle WebCenter Sites

Provide Sites Information

Enter the Sites Host:

localhost

Enter the Sites Port:

7001

Enter the Sites Application Context:

/cs/

Was Sites installed over a secure connection?

○ Yes

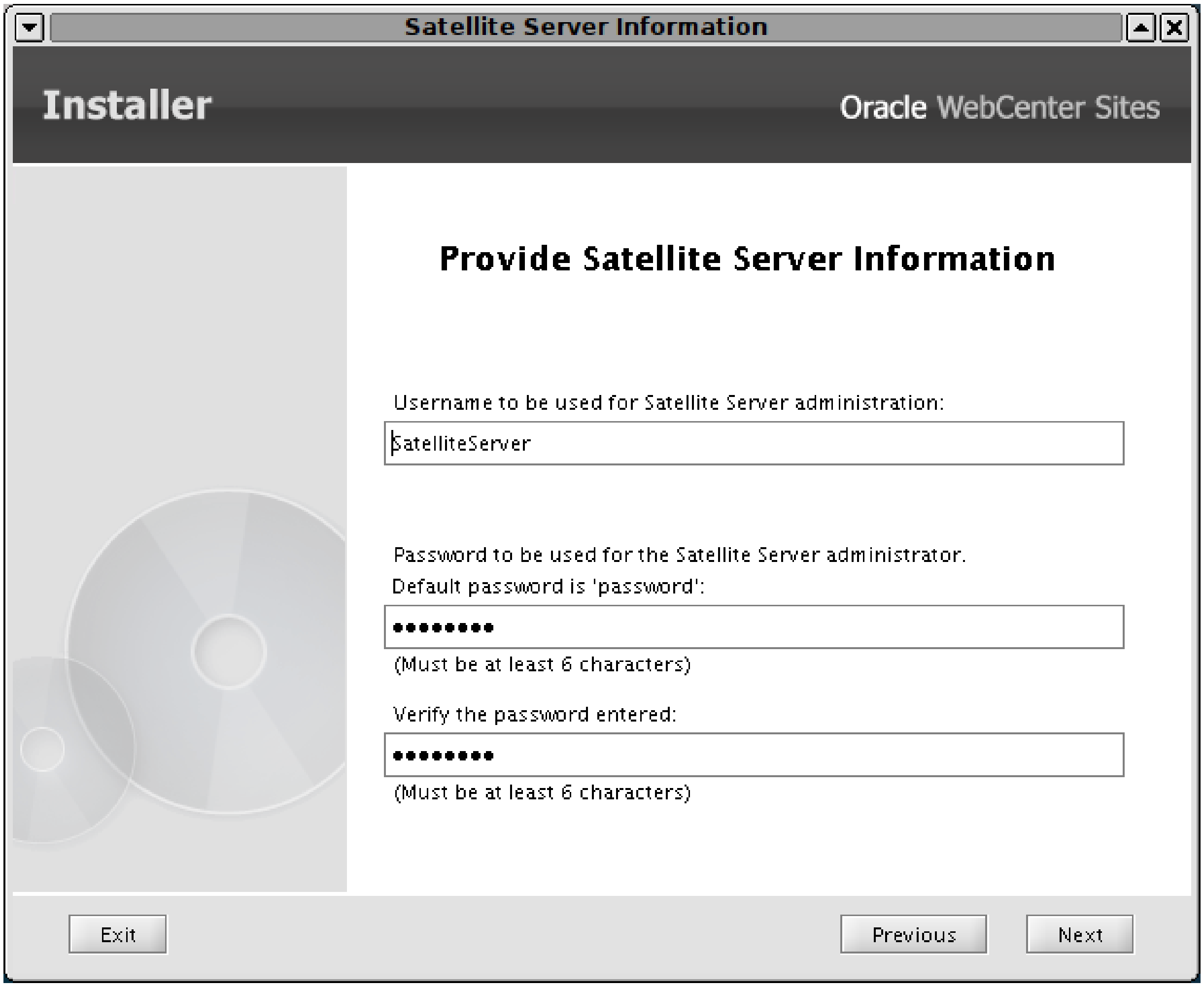
● No

Click **Next**.

7. Provide Satellite Server administrative account information (Figure 9–6).

   - Enter the user's administrative account username. By default, the field is populated with `SatelliteServer`.
   - Enter the user's password.

Figure 9–6 Satellite Server Information



Provide Satellite Server Information

Username to be used for Satellite Server administration:

SatelliteServer

Password to be used for the Satellite Server administrator. Default password is 'password':

●●●●●●●●

(Must be at least 6 characters)

Verify the password entered:

●●●●●●●●

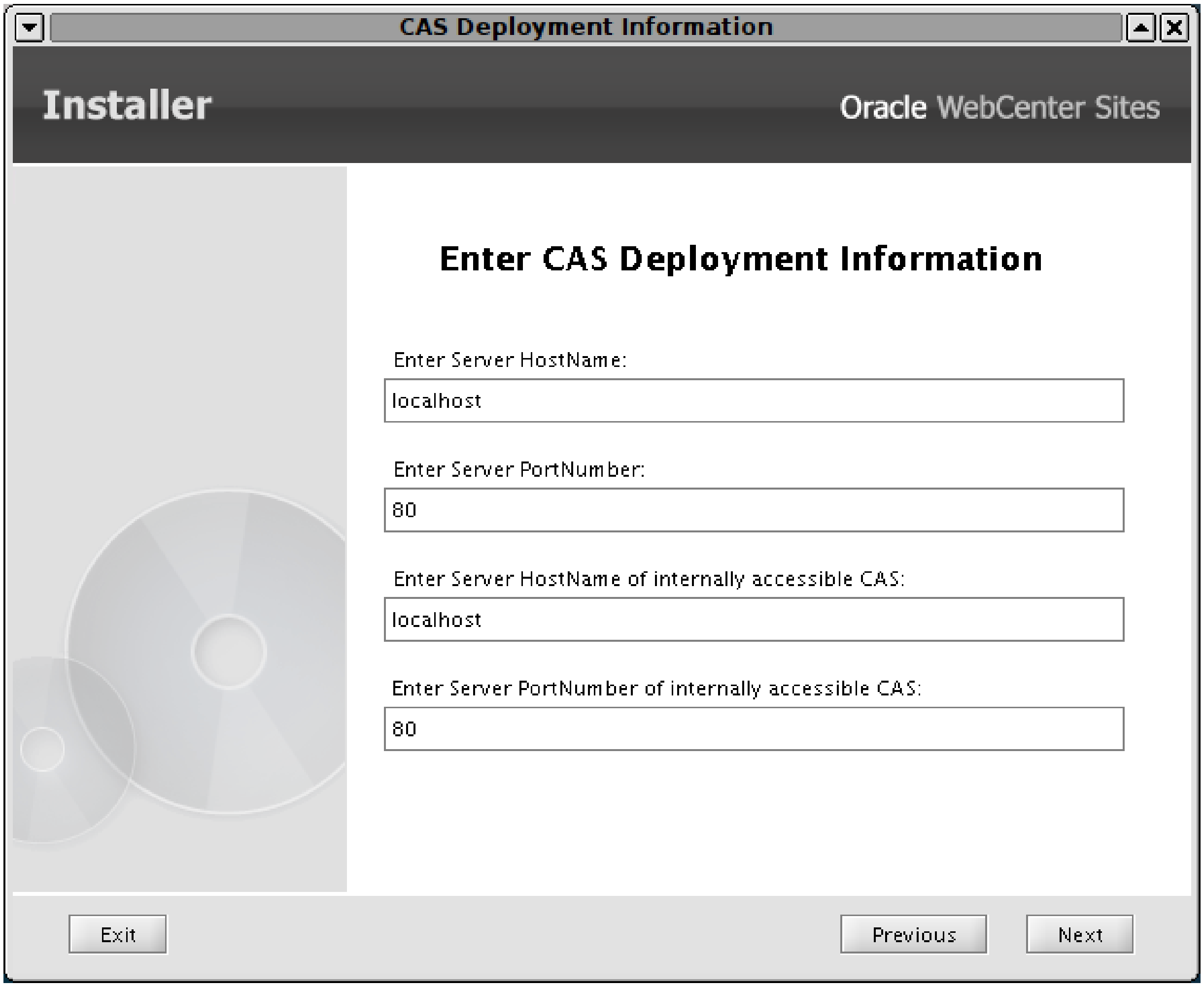
(Must be at least 6 characters)

Click **Next**.

8. Enter CAS Deployment information (Figure 9–7).

- In the first two fields (Server HostName and PortNumber), enter the host name and port number of the server where CAS is deployed.
- The fields for server HostName and PortNumber of internally accessible CAS are mapped to `CASHostNameLocal` and `CASPortNumberLocal` properties respectively. These properties should point to the server on which CAS is actually deployed and not to the load balancer. However, in a cluster, the host should point to the load balancer. Do the following: If the CAS server is not accessible from behind the firewall, enter the host name and port number of the server that is accessible from behind the firewall. Otherwise, enter the same information as in the first two fields.

Figure 9–7 CAS Deployment Information



CAS Deployment Information

Installer

Oracle WebCenter Sites

Enter CAS Deployment Information

Enter Server HostName: localhost

Enter Server PortNumber: 80

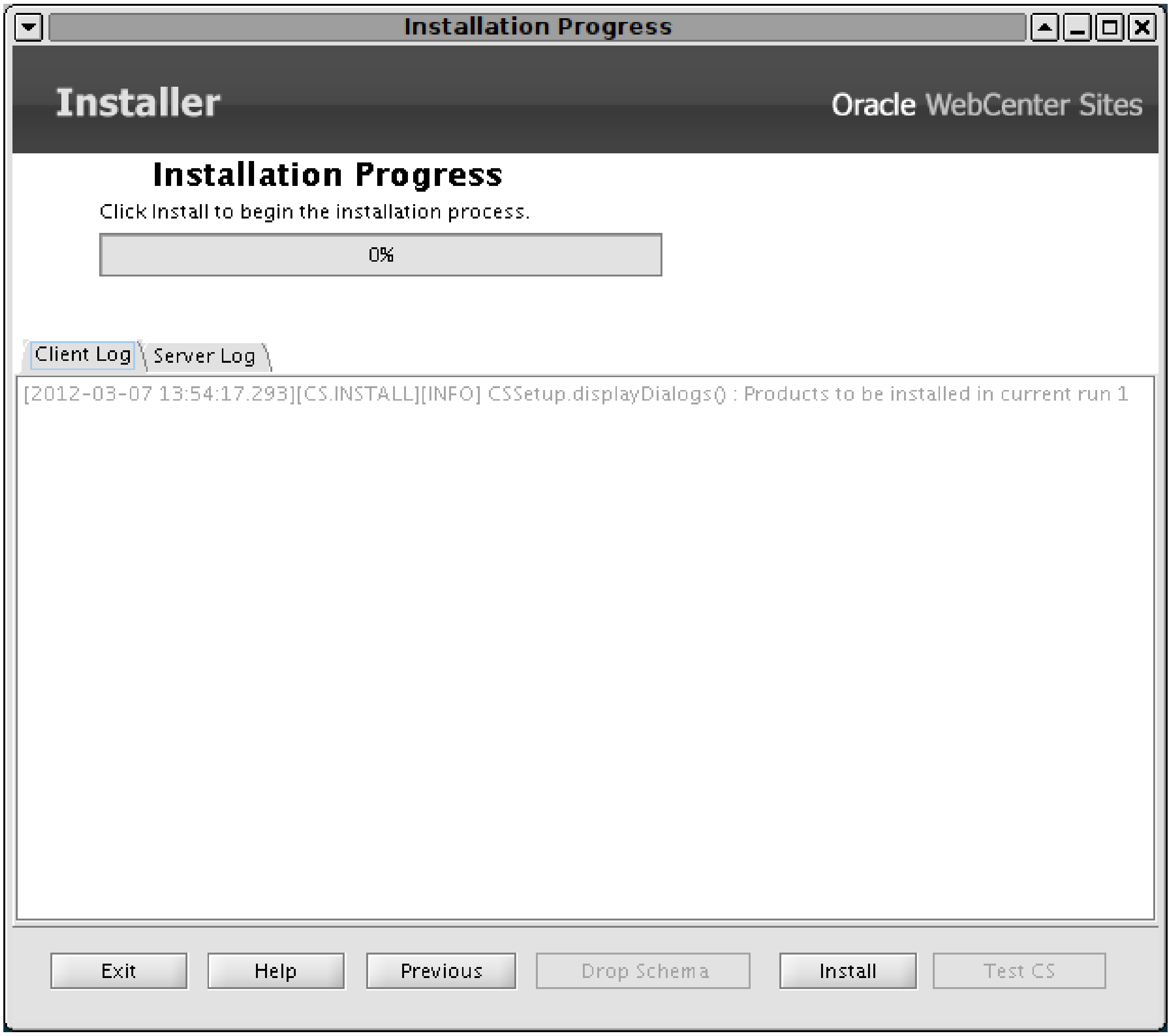
Enter Server HostName of internally accessible CAS: localhost

Enter Server PortNumber of internally accessible CAS: 80

Click **Next**.

9. Click **Install** to start the installation process (Figure 9–8).

Figure 9–8 Installation Progress

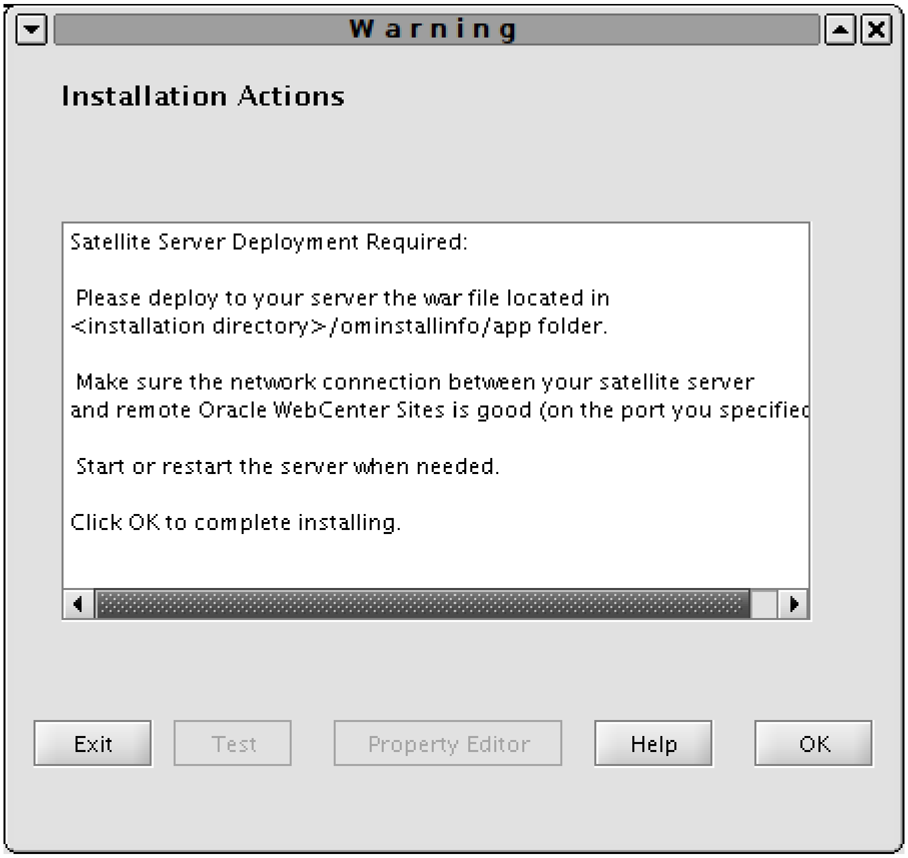


```
[2012-03-07 13:54:17.293][CS.INSTALL][INFO] CSSetup.displayDialogs() : Products to be installed in current run 1
```

10. Deploy Satellite Server (Figure 9–9).

    Satellite Server's `war` file (by default named `cs.war`) can be found in `ominstallinfo/app/` under the Satellite Server installation directory. If necessary, rename the `cs.war` file and deploy the exploded WAR file to the server, then start (or restart) the server, and click **OK** to complete the installation.

Figure 9–9 Installation Actions



Warning

Installation Actions

Satellite Server Deployment Required:

Please deploy to your server the war file located in <installation directory>/ominstallinfo/app folder.

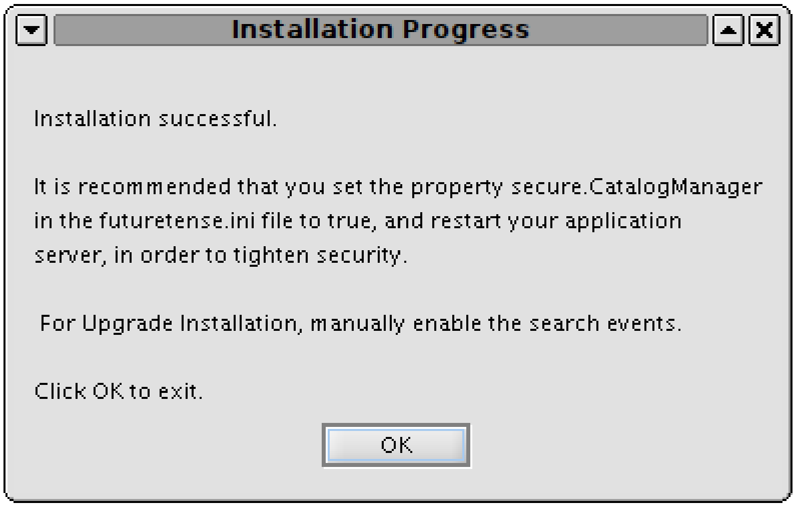
Make sure the network connection between your satellite server and remote Oracle WebCenter Sites is good (on the port you specifiec

Start or restart the server when needed.

Click OK to complete installing.

11. If deployment was successful, you will be presented with a confirmation dialog box. Click **OK** to review the installation log (Figure 9–10).

Figure 9–10 Installation Progress



Installation Progress

Installation successful.

It is recommended that you set the property secure.CatalogManager in the futuretense.ini file to true, and restart your application server, in order to tighten security.

For Upgrade Installation, manually enable the search events.

Click OK to exit.

12. Click **Exit** to quit the installer.

9.3.2 Installing Satellite Server Silently

To install Satellite Server silently:

1. Generate the `omii.ini` file:

   a. Run the GUI installer for Satellite Server, as described in steps 1 to 8 in Section 9.3.1, "Installing Satellite Server Using GUI Installer."

   b. Stop the GUI installer when the Installation Progress screen (Figure 9–8) is displayed.

2. From `<RSS_Home>\omii`, copy the `omii.ini` file that contains the parameters entered into the GUI installer.

3. Edit the `omii.ini` file as required and paste it in the Satellite Server installation directory.

4. Edit the `install.ini` file located in the Satellite Server installation directory, as follows:

   ```
   nodisplay=true
   loadfile=<path to the omii.ini file you just edited>
   ```

5. Run `ssInstall.sh -silent`.

9.4 Step 4: Registering Satellite Server with WebCenter Sites

In this step, you will register remote Satellite Server with WebCenter Sites so that WebCenter Sites can properly manage the remote Satellite Server cache. All Satellite Server installations must be registered with their respective WebCenter Sites installations.

> **Note:** Each remote Satellite Server can be associated with only a single WebCenter Sites installation. This means that the same remote Satellite Server cannot be used for two independent WebCenter Sites installations (such as management and production)

1. From a Windows system, open Sites Explorer and log in to WebCenter Sites as a user that has `SiteGod` privileges (for example, the general administrator or the `ContentServer` user).

2. Click the **System Satellite** tab.

3. The table that opens must be populated with specific values, and each row represents a unique Satellite Server system. Consult the table below for suggested values.

Column	Value
id	A numerical (positive integer) value that identifies this Satellite Server. It must not be the same as any other value in the same column in any other row of the table.
description	A text description identifying this Satellite Server to users. It is used for reference purposes only.
protocol	The protocol on which Satellite Server is accepting requests. This is usually: `http`

Column	Value
host	The host name or IP address of the Satellite Server. This must be a host of the actual Satellite Server engine, not the load balancer.
port	The port on which Satellite Server is listening for requests
satelliteservletpath	The part of the URL from the port number up to, and including, the name of the Satellite servlet. This is usually: `/servlet/Satellite`
flushservletpath	The part of the URL from the port number up to, and including, the name of the `FlushServer` servlet. This is usually: `/servlet/FlushServer`
inventoryservletpath	The part of the URL from the port number up to, and including, the name of the inventory servlet. This is usually: `/servlet/Inventory`
pastramiservletpath	The part of the URL from the port number up to, and including, the name of the inventory servlet. This is typically: `/servlet/Pastrami`
username	The user name assigned to this Satellite Server.
password	The password assigned to this Satellite Server. The password will be automatically encrypted by Sites Explorer after you enter it.

4. From the File menu, select **Save** to save the changes.
5. Exit Sites Explorer.

Note: For configuring co-RSS follow the steps as mention in Step 4: Registering Satellite Server with WebCenter Sites.

9.5 Step 5: Specifying Trusted URLs for CAS

On successful login, CAS redirects the user to the requested service. For security, a requested service can be validated prior to CAS redirecting the user to the service. For instructions on specifying the URLs of requested services, see Section 7.2, "Specifying Trusted URLs for CAS."

9.6 Step 6: Starting Satellite Server

Restart the application server, if necessary. See Part I, "Installing Oracle WebCenter Sites" for information on how to restart (or deploy) your application server.

9.7 Step 7: Testing the Configuration

Before you install Satellite Server on other machines, test the newly installed Satellite Server to ensure it is communicating properly with WebCenter Sites. To test Satellite Server, you will need to publish content from a WebCenter Sites system running in content management mode.

To test your configuration

1. Configure your load balancer to send all WebCenter Sites requests to a single Satellite Server instance.

2. Using a browser, go to a Satellite Server URL. For example:

```
http://<server>:<port>/<URI>/Satellite?pagename=<MyPage>
```

   where `<MyPage>` is any page on your WebCenter Sites system.

   > **Note:** It is easiest to discover the Satellite Server URL you will need to enter by previewing a page on the content management system, copying its URL, and then changing it to point to the remote Satellite Server

3. If you configured everything properly, your browser displays the selected page.

   If your browser did not display the selected page, review the following:

   - Did you set up the load balancer properly? Remember, for this test, every request for WebCenter Sites has to go to the Satellite Server machine. (The other machines are not yet configured and therefore not enabled to handle these requests.)
   - Did you set the Satellite Server properties properly? In particular, ensure that you set the host name and port number to the proper values.
   - Did you request an invalid page from WebCenter Sites?

9.8 Step 8: Installing Satellite Server on Additional Remote Servers

After you have installed and tested your first remote Satellite Server instance, install and configure the remaining Satellite Server instances by repeating the steps in this chapter.

9.9 Next Step

Continue to Chapter 10, "Satellite Server Post-Installation Steps" to adjust caching conditions and optimize the performance of your WebCenter Sites system.

10

Satellite Server Post-Installation Steps

After you have installed and configured your remote Satellite Servers, you can adjust caching conditions and optimize the performance of your WebCenter Sites system.

This chapter contains the following topics:

- Section 10.1, "Adjusting Caching Conditions"
- Section 10.2, "Log Configuration"

10.1 Adjusting Caching Conditions

If You Are Using inCache

If you are using the inCache framework, you can tune it by setting properties that are listed in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

For either type of caching framework, inCache or traditional, you can set the expiration property, in the satellite.properties file. The expiration property defines the default expiration time for blobs when a cache expiration value is not specifically set for blobs with the `satellite.blob` or `RENDER.SATELLITEBLOB` tag that generated the items.

- Setting expiration to never notifies Satellite Server that blobs should never expire for time reasons. Such objects are not guaranteed to remain in the cache indefinitely. For example, if the cache is full, Satellite Server still removes objects from cache based on an LRU (least recently used) algorithm.
- Setting expiration to immediate instructs Satellite Server not to cache pages, pagelets, or blobs at all.

To define a specific set of expiration dates and times, assign a string that uses the following format for the expiration property:

```
hh:mm:ss W/DD/MM
```

The value of this property follows the syntax of a `TimePattern` object. The syntax definition is reproduced in Table 10–1 for convenience.

Table 10–1 TimePattern Syntax

Parameters	Legal Values	Description
hh	0-23	The hour. For example, 0 means midnight, 12 means noon, 15 means three in the afternoon, and so on.
mm	0-59	The number of minutes past the hour.

Table 10–1 (Cont.) TimePattern Syntax

Parameters	Legal Values	Description
ss	0-59	The number of seconds past the minute.
w	0-6	The day of the week. For example, 0 means Sunday, 1 means Monday, and so on.
DD	1-31	The day of the month.
MM	1-12	The month of the year. For example, 1 means January, 2 means February, and so on.

For example, the following expiration value means "3:30 in the afternoon every Monday and on the 15th of April":

```
15:30:00 1/15/4
```

If you specify a value for both W and DD, both values apply. Thus, pages expire on Monday (the W field) and on the 15th (the DD field). To indicate a day-of-week expiration only, place an asterisk in the DD field. For example, to indicate expiration at 3:30 in the afternoon every Monday in April, set the expiration value to:

```
15:30:00 1/*/4
```

To indicate a day-of-month expiration only, place an asterisk in the W field. For example, to indicate expiration at 3:30 in the afternoon on April 15, set the expiration value to:

```
15:30:00 */15/4
```

Setting the hh, mm, ss, or MM fields to an asterisk means all legal values. For example, to indicate expiration at 3:30 in the afternoon on Mondays and the 15th of every month, set the expiration value to:

```
15:30:00 1/15/*
```

You can also place multiple values for any of the six fields by separating the values with commas. To represent a range of values, use a minus sign. For example, the following expiration value represents 6:00 (morning), 1:00 (afternoon), and 5:00 (afternoon), Monday through Friday in June.

```
6,13,17:00:00 1-5/*/6
```

To indicate that pages must expire every 15 minutes, set the expiration value to the following:

```
*:15,30,45:0 */*/*
```

The default value is:

```
5:0:0 */*/*
```

This means that everything in the Satellite Server cache expires every day at 5:00 a.m.

10.2 Log Configuration

Satellite Server uses Apache's Jakarta Commons Logging. By default, no specific JCL configuration information is specified. As a result, JCL will record INFO, WARN and ERROR messages to the console. Users can specify detailed configuration information by placing an empty file called `commons-logging.properties` in the following directory:

```
<$SatelliteServerRoot>/WEB-INF/classes
```

and then editing the file using the Property Editor. The Property Editor provides detailed log configuration information about each property.

To open the Property Editor, run the settings.bat batch file (Windows) or the settings.sh script (UNIX). Open the commons-logging.properties file; it will open with several tabs. Under the Loggers tab, among other entries, you will see:

```
com.fatwire.logging.cs.satellite
com.fatwire.logging.cs.satellite.cache
com.fatwire.logging.cs.satellite.host
com.fatwire.logging.cs.satellite.request
```

These are the loggers that Satellite Server uses. Consult the property descriptions in the Property Editor for information about each logger, as well as the possible values. Under the **Factory** tab, you can choose the type of logger you want Satellite Server to use. By default, the Property Editor sets this to:

```
COM.fatwire.cs.core.logging.TraditionalLog
```

This allows you to write log messages to a log file that is configured under the **TraditionalLog** tab. (Note that the `logging.file` property is required.)

To send messages to the console, set the `org.apache.commons.logging.Log` property to either blank or `COM.FutureTense.Logging.StandardLog`. When you are done, save the changes, exit the Property Editor, and restart Satellite Server by restarting the application server. Consult the JCL website at `http://jakarta.apache.org/commons/logging/` for more information about JCL.

Part III

Internationalizing Oracle WebCenter Sites

Part III contains the following chapters:

11 Configuring Your Environment for Multi-Language Operations

This chapter describes how to configure your environment to enable multi-language operations in WebCenter Sites.

This chapter contains the following sections:

- Section 11.1, "WebCenter Sites-Specific Settings"
- Section 11.2, "Internet Explorer Settings"
- Section 11.3, "Additional Specifications for Internationalization"

Note: For setting UTF values, see "Configuring URI Encoding", and "Setting the Classpath and Java Options".

11.1 WebCenter Sites-Specific Settings

Note:

- The settings that you choose for a given WebCenter Sites instance must be reproduced on all cluster members (if any) and across all environments (development, content management, delivery, and so on).
- If none of the properties and variables described below are set, the `cs.contenttype` property defaults to `text/html`. The character set of the output now defaults to default system encoding.

11.1.1 cs.contenttype Property

The `cs.contenttype` property, in `futuretense.ini`, is a system-wide (global) property that defines the outgoing character encoding. By default, the property is set to `text/html; charset=UTF-8`. If you need a specific encoding, change the value. For example, if you want the outgoing encoding to be `Shift_JIS`, set this property to `text/html; charset=Shift_JIS`. Sites Explorer depends on this setting to display data correctly.

11.1.2 cs.contenttype Variable (Set in SiteCatalog resargs1)

The `cs.contenttype` variable enables you to control the outgoing encoding on a page-by-page basis. This variable overrides the value defined in `cs.contenttype` property. The variable should be set in the same way as the `cs.contenttype` property, as shown in Section 11.1.1, "cs.contenttype Property."

Note that pages under WebServices are set to cs.contenttype=application/xml;charset=UTF-8.

11.1.3 _charset_ hidden Variable in Forms

If you are using an HTML form tag to input international data, make sure to set the _charset_ input type variable at the very beginning, after the form declaration. For example:

```
<form action='ContentServer' method='get'>
<input type='hidden' name='_charset_'/>
<input type='hidden' name='pagename' value='<%=ics.GetVar("pagename")%>'/>
<input type='text' name='name' value='<%=ics.GetVar("name")%>'/>
<input type='submit'/>
</form>
```

Without a value, the _charset_ hidden variable works only on the Internet Explorer browser.

11.1.4 Preferred Encoding

When WebCenter Sites needs to consume HTTP requests with certain encodings (Cp943C for example) that are closely related to a more widely used encoding (Shift_JIS), it is not sufficient to rely on the _charset_ hidden variable alone. Internet Explorer, when it encounters a _charset_ value set to Cp943C, changes it to Shift_JIS. This forces WebCenter Sites to read all data in Shift_JIS. To overcome this, a special names property syntax is used:

```
cs.contenttype.<charset>=<preferred_encoding_for_this_charset>
```

For example, in relation to the scenario described above, this property would be specified as follows to indicate to WebCenter Sites to use Cp943C:

```
cs.contenttype.Shift_JIS=Cp943C
```

Note that this property structure is necessary only in special circumstances such as the one described above, where the behavior of Internet Explorer conflicts with and changes the value of _charset_ value.

11.1.5 Encoding Specified in XML or JSP Elements

The encoding in the <?xml line of an XML element specifies the encoding of the .xml file on disk. The same is true of JSP. The encoding specified in the page directive specifies two things. The first is the encoding of the .jsp file on disk. The second is the outgoing encoding of the evaluated JSP element. This gets converted to the encoding of the enclosing JSP, or in the XML case, the outgoing encoding of the page (content-type). So cs.contenttype can be used to indicate that the outgoing page will have a specific encoding, such as Shift-JIS, but a JSP can output UTF-8, and the UTF-8 will get converted to Shift-JIS into the output stream of the page response. The following example shows how to specify the encoding:

- XML: <?xml version="1.0" encoding="UTF-8"?>
- JSP: <%@ page contentType="text/html; charset=UTF-8" %>

11.1.6 Using SetVar Tag

You can also control the outgoing page encoding by using the SetVar tag in JSP and XMLs. The SetVar tag must be set before anything is streamed out.

In JSPs, you can write:

```
<cs:ftcs>
<ics:setvar name="cs.contenttype" value="text/html; charset=UTF-8" />
...
</cs:ftcs>
In XML, you have the following options:
<ftcs>
<setvar name="cs.contenttype" value="text/html; charset=utf-8"/>
...
</ftcs>
```

The second option is to use the `ics.streamheader` XML tag, but again this must be done before anything is streamed, and only in XML.

```
<ftcs>
<ics.streamheader name="Content-Type" value="text/html; charset=utf-8"/>
...
</ftcs>
```

11.1.7 Using HTTP META tag

If the encoding is specified by any of the methods above, then the META tag has no effect. Otherwise, the browser tries to display the data in the encoding specified by the META tag.

11.2 Internet Explorer Settings

Internet Explorer 8 by default has all the languages installed. If you are unable to see text in a particular language, you most probably need to enable it in your browser.

To view content in different languages

1. Go to **Tools** > **Internet Options** > **General** tab.
2. Click **Languages** on the bottom right of the page.
3. Click **Add** to add more languages to the list of languages already displayed in the box.
4. Click **OK**.
5. Close and reopen Internet Explorer. You should now see the content in your specified language if the web page so provides.

11.3 Additional Specifications for Internationalization

The following sections list several features in WebCenter Sites and spark with specific internationalization requirements. The features are:

- Section 11.3.1, "Files Stored on the File System"
- Section 11.3.2, "Attribute Editor"
- Section 11.3.3, "XML Post"
- Section 11.3.4, "Catalog Mover"
- Section 11.3.5, "Sites Explorer"
- Section 11.3.6, "Sites Desktop and Sites DocLink"

- Section 11.3.7, "WebCenter Sites Interfaces"
- Section 11.3.8, "Functional Restrictions"

11.3.1 Files Stored on the File System

Many indirect files are stored on the file system because of the `url` column references. The files are identified in this section.

11.3.1.1 XML and JSP Files

See Section 11.1.5, "Encoding Specified in XML or JSP Elements."

11.3.1.2 HTML Files

The HTML files are read using the file.encoding Java parameter value. The data in the file also depends on the way it was stored initially.

11.3.1.3 SystemSQL Queries

SystemSQL uses a `url` column to point to a file on the file system that holds a SQL query. When this file is loaded, it is assumed that the encoding of the file is the Java default encoding (`System.getProperty("file.encoding")`). There are several possible ways to create the SystemSQL queries (use Sites Explorer, a text editor, or WebCenter Sites). It is probably best to always use ASCII7 when creating queries, since any data is typically merged using variable replacement at run time.

11.3.1.4 Page Cache Files Referenced from the SystemPageCache Table

For page cache files, we manage the page cache file so that the encoding of the file is UTF-8. Since we write and read this file only through WebCenter Sites or Spark, it can be managed this way.

11.3.2 Attribute Editor

You have two ways to specify the text for an attribute editor:

- Type in the text in the text area provided. The form post will determine the encoding.
- Use the **Browse** button to select a text file. The text file encoding should match the encoding specified in xcelerate.charset encoding. (The `xcelerate.charset` property is located in the `futuretense_xcel.ini` file.)

11.3.2.1 Article Bodies, Flex Assets, User-Defined Assets

The article bodies are stored on the file system using the file.encoding Java property value.

11.3.3 XML Post

When non ASCII files are posted via XML Post, the java file encoding must match the encoding of the file. For example, if you are posting a Japanese file (stored as UTF-8) to a UTF-8 system, then one of the following should be set before the XML Post command is run:

- The system locale must be set to UTF-8.
- The option of `-Dfile.encoding=UTF-8` must be specified in the XML Post command.

Similarly, if the file is stored as `Shift_JIS`, then the corresponding system locale should be set or the java `file.encoding` option must be specified.

WebCenter Sites supports the encoding in the `<?xml` line, as the first line in the posted XML file. This overrides everything else as far as the encoding in which the `.xml` file is read.

11.3.4 Catalog Mover

Open the `CatalogMover.bat` (or `.sh` on UNIX) file and modify the `java` command to include the file.encoding parameter with a value that reflects the encoding needed to display the characters stored in the catalogs. This step can be avoided if the default encoding of the file system matches that of the data stored in the catalogs.

11.3.5 Sites Explorer

Sites Explorer requires that the cs.contenttype property in `futuretense.ini` be set to the correct value. If you are viewing simple ASCII characters, nothing needs to be done. However, for viewing complex characters, such as Japanese, you need to set `cs.contenttype` to one of the following:

- `text/html;charset=SJIS`
- `text/html;charset=UTF-8`

Also, users may need to load font support for Japanese and various other character sets in order to have them display correctly. To do this in Windows 2003 for example, go to **Settings** > **Control Panel** > **Regional Options**. In the first tab, **General**, select the languages you want to support from the list titled "Language settings for the system." Click **Apply** then **OK**. At this point you will be required to put in the Windows installation CD.

11.3.6 Sites Desktop and Sites DocLink

The Sites Desktop and Sites DocLink clients support the character sets supported by Windows. To enable a specific character set in Sites Desktop or Sites DocLink, first enable it in Windows. For instructions, see the Microsoft Windows documentation.

11.3.7 WebCenter Sites Interfaces

The user's machine must be able to support the characters that are to be displayed in the WebCenter Sites interfaces. For languages other than English, the user needs to make sure of the following:

- The appropriate fonts to display the characters are installed.
- On a Windows machine, the locale and language settings must support the characters that will be displayed. For example, if Japanese characters are to be displayed in the interfaces, Windows must first be configured to display Japanese characters. (For instructions on configuring Windows to support the target language, see the Windows documentation.)
- For a UNIX machine, the locale (`LANG` and `LC_ALL` environment variables) must be appropriately set.
- The browser's encoding must be correctly set.

11.3.7.1 Single-Language Restrictions

Although you can configure a content management system to be multi-lingual, certain parts of the user interface can be displayed in one language only.

For example, the names of tables and columns in the WebCenter Sites database as well as individual items such as categories and source codes can have only one name. This means that although much of the text on an individual WebCenter Sites form can be displayed in multiple languages, items such as field names and asset type names can be displayed in one language only.

Following is a list of items in WebCenter Sites that can have one name only, which means that they can be displayed in one language only:

- Asset type names
- Field names
- Asset names
- Categories
- Source codes
- Tree tab names
- Site names
- Names of workflow building blocks (actions, e-mail objects, conditions, states, steps, processes)
- Role names
- Start menu items—both Search and New

On a system that supports two or more languages, you must determine which language is going to be used by the majority of content providers and then use that language to name your sites, tabs, asset types, and so on.

11.3.8 Functional Restrictions

WebCenter Sites has the following functional restrictions for international use:

- The Property Editor supports ASCII only.
- Decimal numbers must be entered in US format ("." decimal separator).

Part IV

Installing Oracle WebCenter Sites: Community-Gadgets

This part describes the process of installing and configuring the Oracle WebCenter Sites: Community-Gadgets web application in single and clustered mode.

Part IV contains the following chapters:

- Chapter 12, "Overview of Community-Gadgets"
- Chapter 13, "Prerequisites for Installing Community-Gadgets"
- Chapter 14, "Configuring Application Servers for Community-Gadgets"
- Chapter 15, "Generating Community-Gadgets Deployment Files"
- Chapter 16, "Completing Community-Gadgets Pre-Deployment Steps"
- Chapter 17, "Deploying Community-Gadgets"
- Chapter 18, "Verifying and Implementing Community-Gadgets"
- Chapter 19, "Adding Sample Gadgets"

12 Overview of Community-Gadgets

This chapter provides an overview of WebCenter Sites: Community-Gadgets and describes its configuration for both a simple development and content management environment, and a complex production environment.

This chapter contains the following sections:

- Section 12.1, "About Community-Gadgets"
- Section 12.2, "Production and Management Environments"
- Section 12.3, "Community-Gadgets Configurations"

12.1 About Community-Gadgets

Community-Gadgets is a set of Java EE web applications that integrates with Oracle WebCenter Sites and works in a distributed environment as a social computing application. Community-Gadgets provides its users with two Web Experience Management (WEM) applications: Community and Gadgets.

The Community WEM application enables its users to configure widgets for gathering visitors' comments, reviews, and ratings on website content. The Community WEM application also enables its users to create and manage polls for surveying site visitors about desired topics. The Gadgets WEM application enables its users to deploy individual gadgets, or even a dashboard with one or more gadgets, to website visitors.

Before proceeding to describe the configurations in which Community-Gadgets can be installed, this chapter provides an overview of Community-Gadgets, its relationship to WebCenter Sites, its components, and their purposes.

This section contains the following topics:

- Section 12.1.1, "Community-Gadgets Using Components of WebCenter Sites"
- Section 12.1.2, "Community-Gadgets Management and Production Components"

12.1.1 Community-Gadgets Using Components of WebCenter Sites

Community-Gadgets integrates with WebCenter Sites through the Web Experience Management Framework to make use of the following resources:

- **Asset repository**: Community-Gadgets requires an asset repository. WebCenter Sites provides an asset repository, that is, a database. Community-Gadgets does not directly access the database. All interaction with the database is managed by WebCenter Sites.

- **WebCenter Sites: Web Experience Management Framework (WEM Framework)**, which provides Community-Gadgets with the REST service and the Central Authentication Service (CAS).
  - REST service enables Community-Gadgets to communicate with WebCenter Sites to make use of its asset model.
  - CAS is used to provide authentication services for the REST API.

> **Note:** Like WebCenter Sites, Community-Gadgets has its own CAS instance, but only on the production side. To distinguish between CAS instances, CAS shipped with WebCenter Sites is called "WEM CAS" (or Sites CAS), and CAS shipped with Community-Gadgets is called "Visitors CAS."
>
> Despite the fact that CAS web applications (WEM CAS and Visitors CAS) are nearly identical at the code level, their configurations differ and they are not interchangeable; nor can they be installed on the same application server or cluster.

More information about CAS is available at the following URL:

`http://www.jasig.org/cas`

> **Note:** WebCenter Sites can be integrated with OAM (instead of CAS).
>
> - On the management system, Community-Gadgets uses the same OAM that is integrated with WebCenter Sites in order to communicate with WebCenter Sites.
> - On the production system, Community-Gadgets uses Visitors CAS for visitors authentication. It cannot be replaced with OAM. However, WebCenter Sites on the production system can be integrated with OAM (like WebCenter Sites on the management system), and Community-Gadgets can use OAM to communicate with WebCenter Sites on the production system.

12.1.2 Community-Gadgets Management and Production Components

Community-Gadgets consists of two parts: a management set of applications, and a production (delivery) set of applications.

> **Note:** Management Community-Gadgets and production Community-Gadgets must be deployed on separate application server instances. Deployment of both on a single application server is not supported.

- Management Community-Gadgets contains two web applications:
  - Community-Gadgets web application
  - Shindig web application
- Production Community-Gadgets contains three web applications:
  - Community-Gadgets web application

- Shindig web application
- CAS web application

Management Community-Gadgets is used for the administration of user-generated content (UGC), registering gadgets, and similar functions. For security reasons, it is typically accessible only internally. Production Community-Gadgets is accessed by visitors through widgets and gadget dashboards deployed onto web pages. It manages visitor authorization through Visitors CAS.

Both management and production Community-Gadgets use Shindig OpenSocial container for rendering gadgets. Production Community-Gadgets must be externally accessible. More information about the Shindig web application is available at the following URL:

`http://shindig.apache.org/`

For its end users, Community-Gadgets provides two WEM applications:

- **Community WEM application** — provides the Community interface, which is used to configure widgets for collecting visitors' comments, reviews, ratings, and votes at polls. The Community WEM application is also used to moderate user-generated content.
- **Gadgets WEM application** — provides the Gadgets administrator interface (Global Catalog) and the Gadgets user interface. Which interface is displayed to the user depends on the user's roles. Both interfaces are used for gadgets management, which includes registering, sharing, and configuring gadgets and the dashboard.

For more information about the WEM Framework and WEM applications, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

12.2 Production and Management Environments

To manage user-generated content, Community-Gadgets requires two environments: a management environment and a production (delivery) environment, where the management and production Community-Gadgets instances are in constant communication with WebCenter Sites.

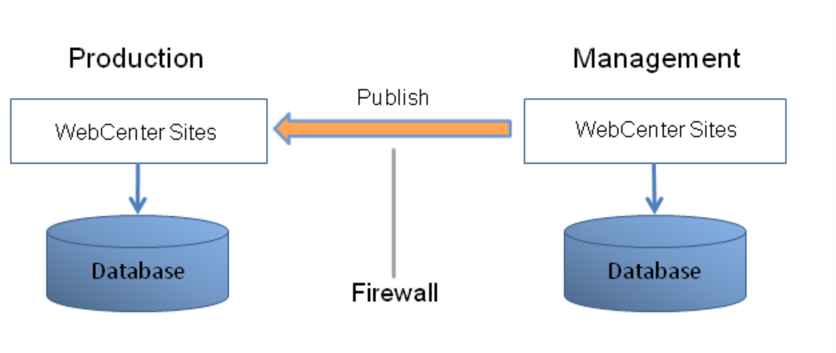
This section contains the following topics:

- Section 12.2.1, "WebCenter Sites Communications"
- Section 12.2.2, "Community-Gadgets Communications with WebCenter Sites"

12.2.1 WebCenter Sites Communications

In WebCenter Sites, content for the website is generated and managed strictly on the management system, then published to the production system, as shown in Figure 12–1. Templates are generated on the content management system, and then published to the production system to configure the appearance of published content.

Figure 12–1 WebCenter Sites Communications



Website content is generated, managed, and published strictly by users of the WebCenter Sites content management application.

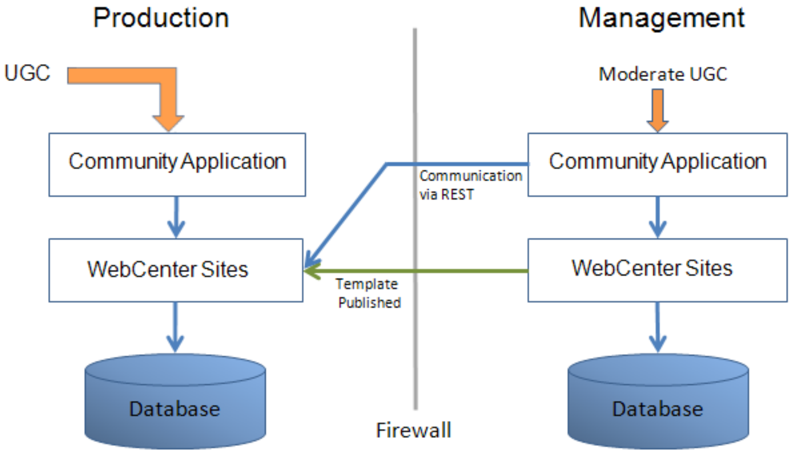
12.2.2 Community-Gadgets Communications with WebCenter Sites

In Community-Gadgets systems, content generation and management processes differ from those on WebCenter Sites because they involve several types of users, including website visitors, all interacting with each other's content through WebCenter Sites.

- Widgets and gadgets are generated and deployed by users of the management Community-Gadgets instance to the WebCenter Sites production system.
- User-generated content, such as comments and reviews, is entered by website visitors into deployed widgets on the Community-Gadgets production instance.
- User-generated content is stored in the WebCenter Sites production database. The content is managed by administrator and moderator users of the management Community-Gadgets.

Throughout these interactions, widgets, gadgets, and content are stored in the WebCenter Sites management and production databases. As shown in Figure 12–2, all content generation and management processes involving Community-Gadgets pass through WebCenter Sites.

Figure 12–2 Community-Gadgets Communications with WebCenter Sites



Note: Figure 12–2 shows the use of a firewall. The firewall must allow free information exchange on all ports used by WebCenter Sites and Community-Gadgets.

12.3 Community-Gadgets Configurations

The Community-Gadgets environment can be configured in many ways. This section describes the commonly used configurations: basic and production.

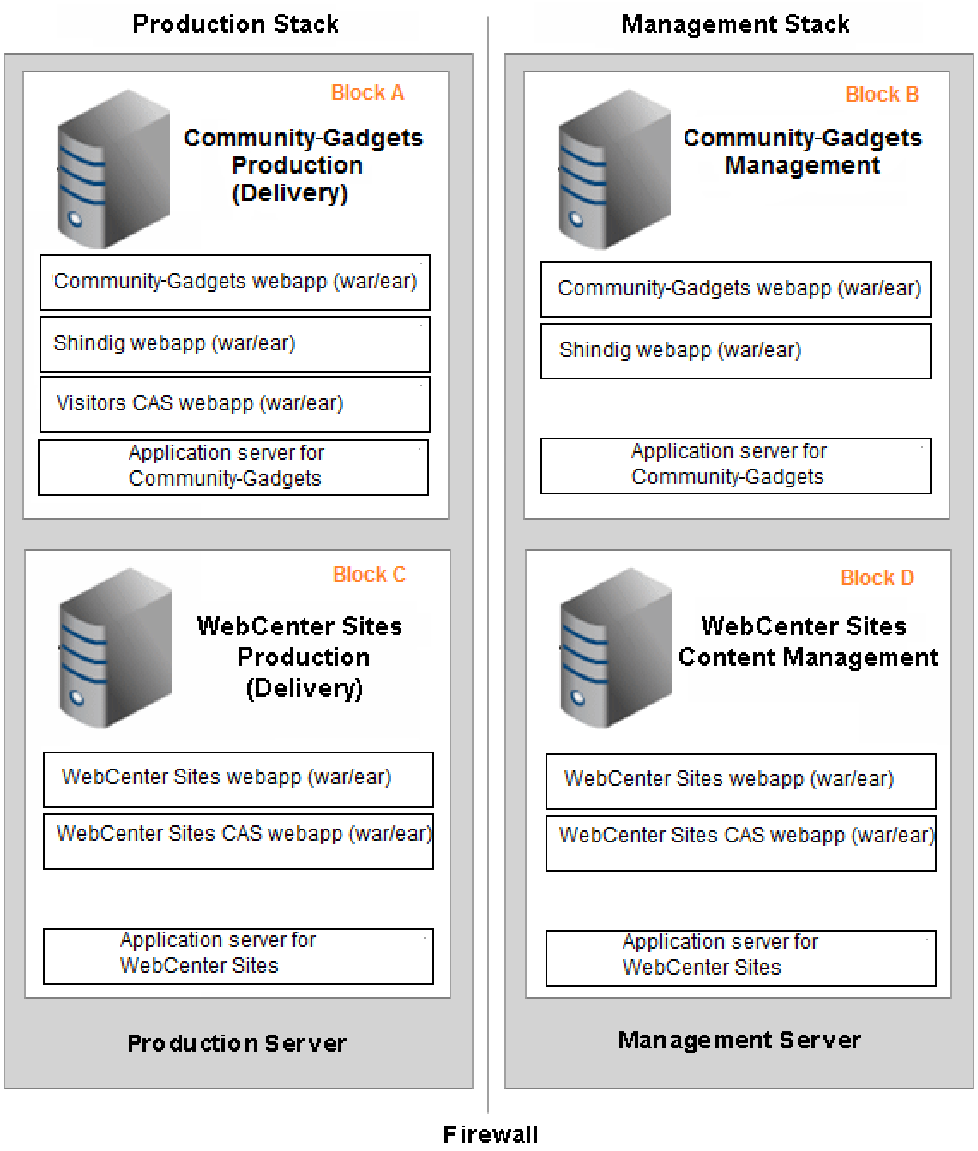
This section contains the following topics:

- Section 12.3.1, "Basic Configuration"
- Section 12.3.2, "Production Configurations"

12.3.1 Basic Configuration

A basic Community-Gadgets environment consists of four self-contained blocks: Community-Gadgets (production), Community-Gadgets (management), WebCenter Sites (production), and WebCenter Sites (content management). These blocks are depicted in Figure 12–3, where they are labeled as A, B, C, and D.

Figure 12–3 Basic Community-Gadgets Environment: Development, Staging, or Limited Production Environment


Production Stack
Management Stack
Block A
Community-Gadgets Production (Delivery)
Community-Gadgets webapp (war/ear)
Shindig webapp (war/ear)
Visitors CAS webapp (war/ear)
Application server for Community-Gadgets
Block B
Community-Gadgets Management
Community-Gadgets webapp (war/ear)
Shindig webapp (war/ear)
Application server for Community-Gadgets
Block C
WebCenter Sites Production (Delivery)
WebCenter Sites webapp (war/ear)
WebCenter Sites CAS webapp (war/ear)
Application server for WebCenter Sites
Block D
WebCenter Sites Content Management
WebCenter Sites webapp (war/ear)
WebCenter Sites CAS webapp (war/ear)
Application server for WebCenter Sites
Production Server
Management Server
Firewall

The blocks shown in Figure 12–3 can be deployed in a number of ways, in any configuration, as long as independent application servers are used for each application inside each block.

> **Note:** Possible configurations range from running all four blocks on different servers to running all the blocks on a single server. Deployment of management and production applications on a single server are supported only for development purposes (non-production environment).

The configuration shown in Figure 12–3 is best suited - and commonly used - for development and staging. It may also be used for QA and production where limited load and no expandability or redundancy are required. In Figure 12–3, two independent servers are used. Each server has an independent stack consisting of an instance of WebCenter Sites: Community-Gadgets and an instance of WebCenter Sites. This means, each server has two independent (non-clustered) application servers and a local database. Between the two servers, communications flow through an optional firewall (depending on security requirements).

12.3.2 Production Configurations

Deploying Community-Gadgets on a production environment uses the same four basic blocks introduced in Section 12.3.1, "Basic Configuration." However, each of these blocks is now divided into sub-blocks to provide both redundancy and scalability. In addition to breaking up the blocks, we recommend using HTTPS for all communications to improve security.

The figures in the rest of this chapter illustrate, at a high level, the different clustered configurations that will be used in a production environment.

- Figure 12–4 shows a sample management stack in a production environment.
- Figure 12–5 shows a sample production (delivery) stack in a production environment.
- Figure 12–6 shows a Community-Gadgets production block divided into two sub-blocks.
- Figure 12–7 shows a Community-Gadgets production block divided into six sub-blocks.

Figure 12–4 Sample Management Stack in a Production Environment

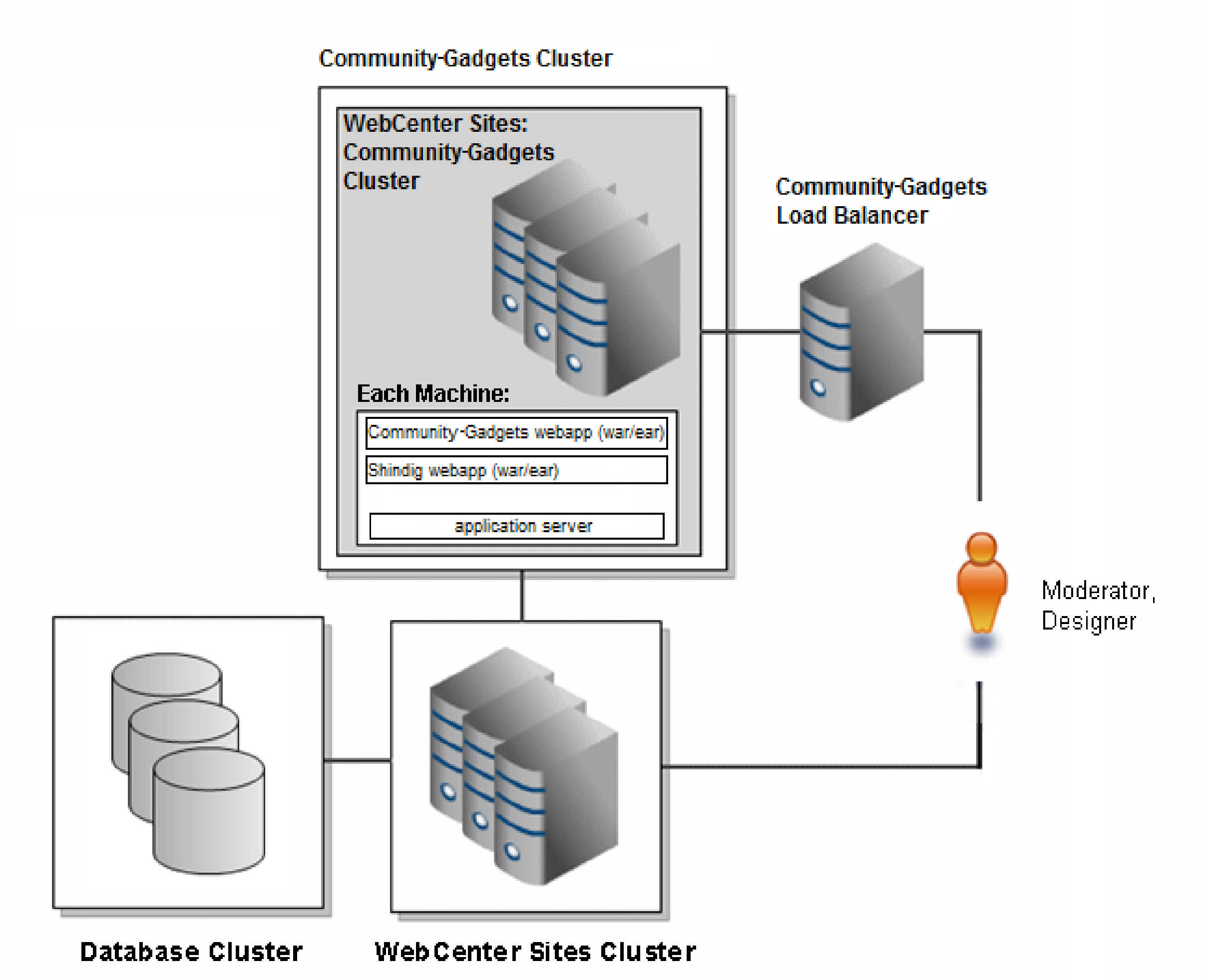

Community-Gadgets Cluster
WebCenter Sites: Community-Gadgets Cluster
Each Machine:
Community-Gadgets webapp (war/ear)
Shindig webapp (war/ear)
application server
Community-Gadgets Load Balancer
Moderator, Designer
Database Cluster
WebCenter Sites Cluster

Figure 12–4 shows a clustered version of the management stack illustrated in Figure 12–3.

In Figure 12–4, the management Community-Gadgets (Block B in Figure 12–3) is now clustered. The management WebCenter Sites application (Block D in Figure 12–3) is also clustered and uses a database cluster.

Figure 12–5 Sample Production (Delivery) Stack in a Production Environment

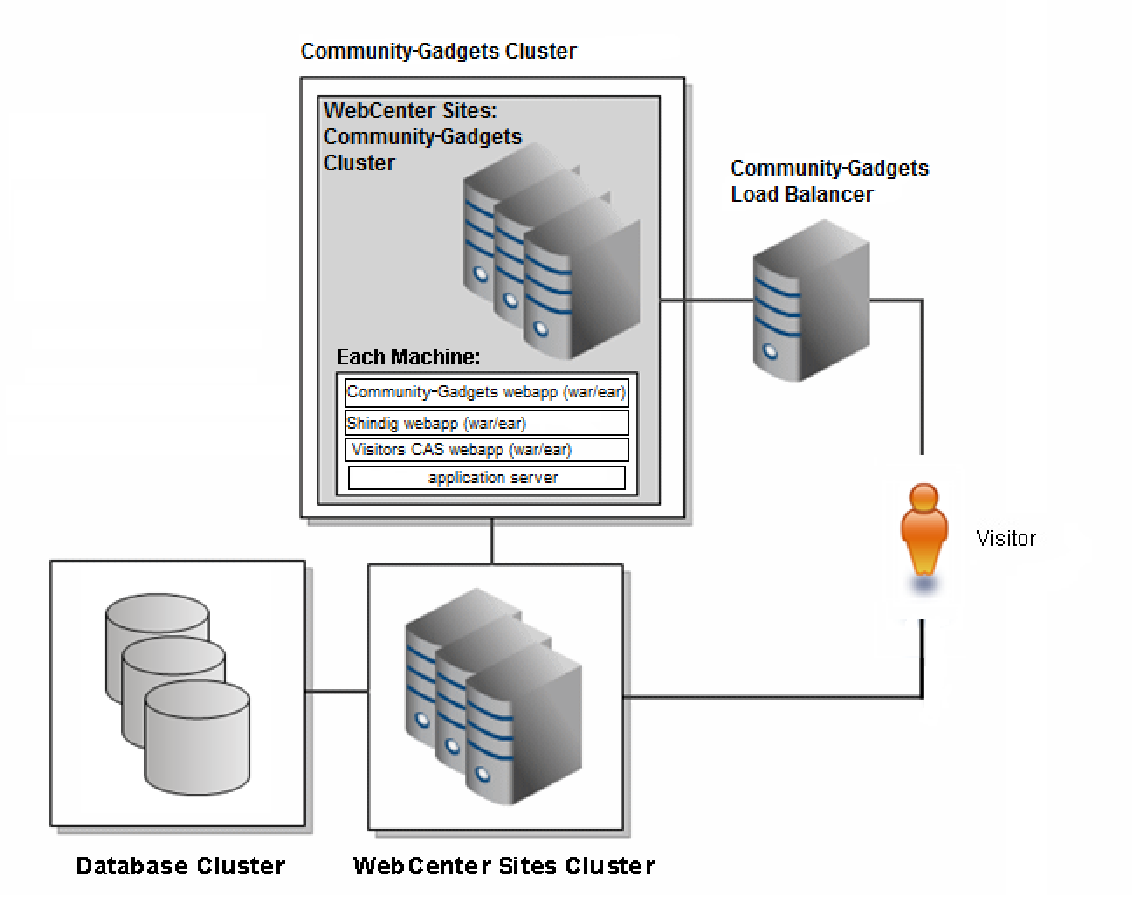
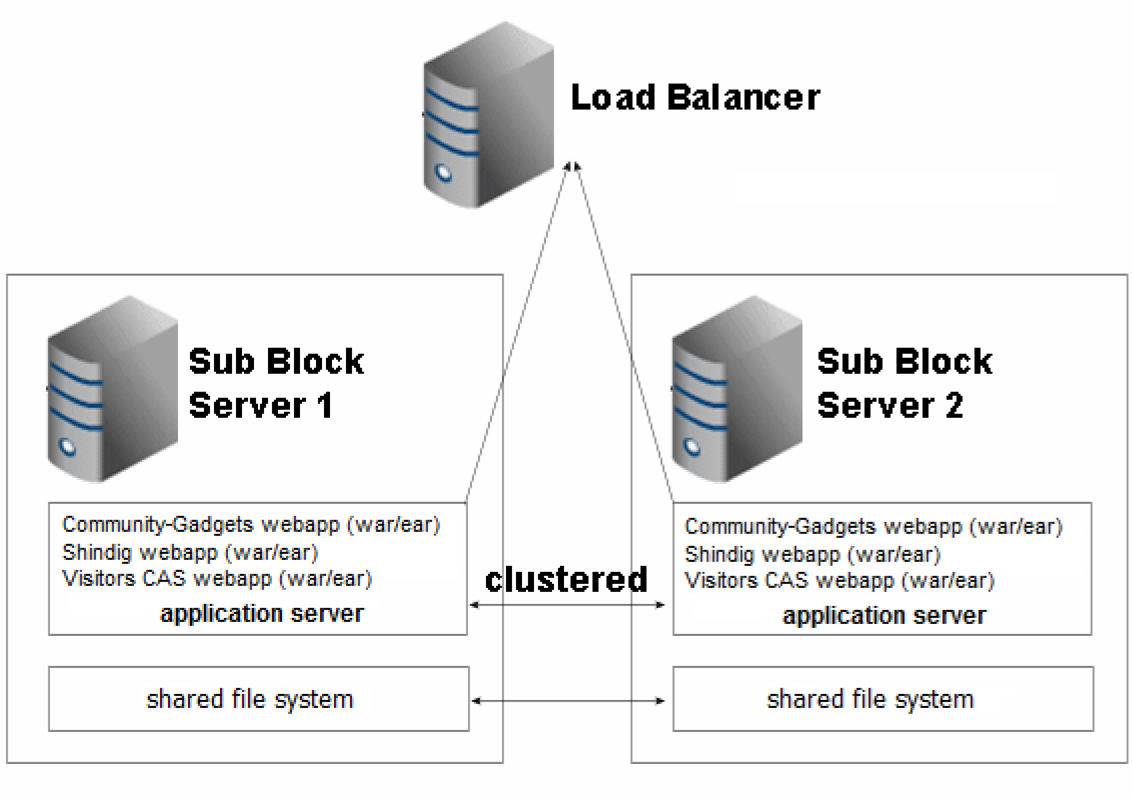

Community-Gadgets Cluster
WebCenter Sites: Community-Gadgets Cluster
Each Machine:
Community-Gadgets webapp (war/ear)
Shindig webapp (war/ear)
Visitors CAS webapp (war/ear)
application server
Community-Gadgets Load Balancer
Visitor
Database Cluster
WebCenter Sites Cluster

Figure 12–5 shows a clustered version of the production stack illustrated in Figure 12–3.

In Figure 12–5, production Community-Gadgets (Block A in Figure 12–3) is now clustered. The production WebCenter Sites application (Block C in Figure 12–3) is also clustered and uses a database cluster.

Figure 12–6 Community-Gadgets Production Block Consisting of Two Sub-Blocks



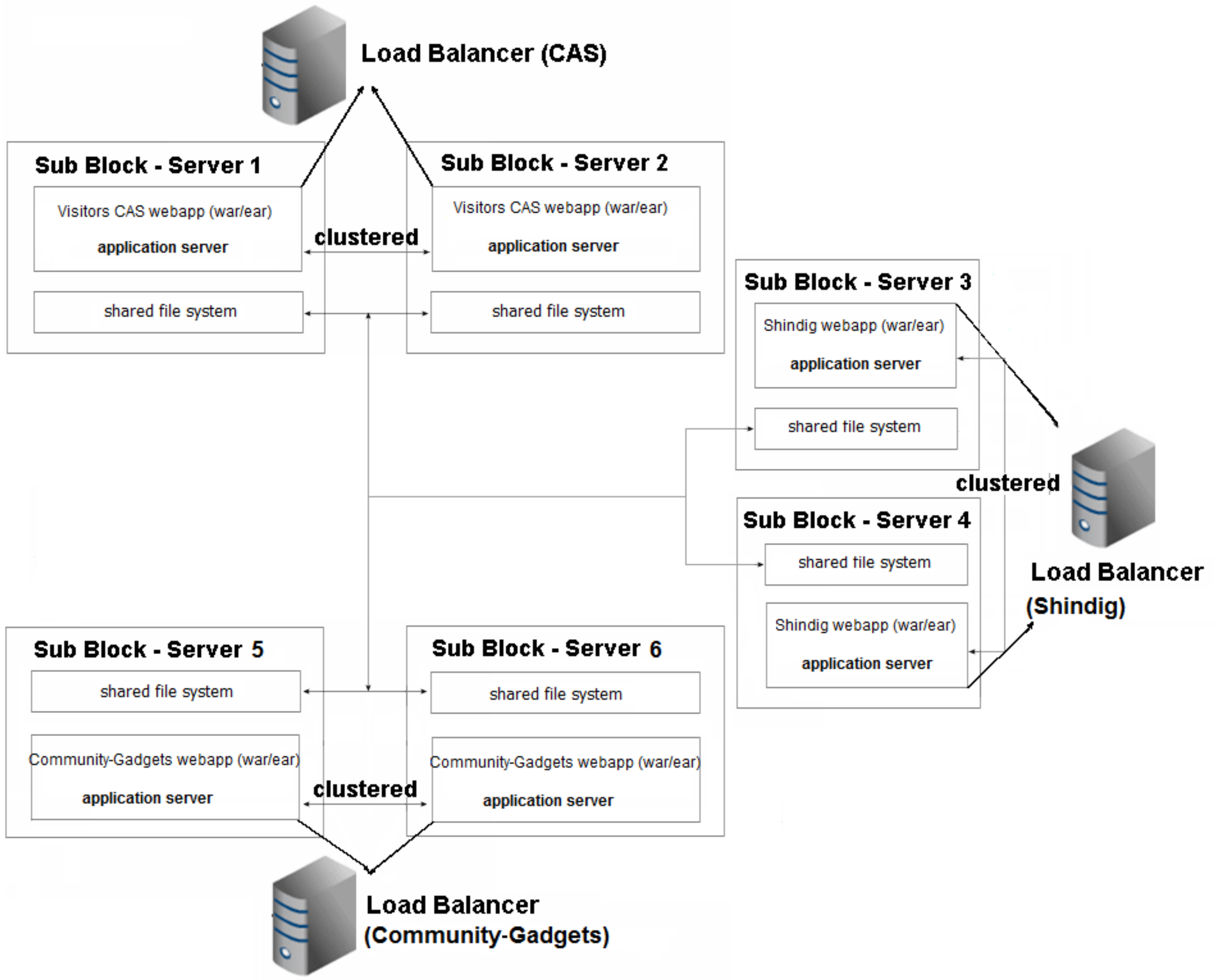
In Figure 12–6, the Community-Gadgets production block (Block A in Figure 12–3) is divided into a cluster of two servers (which can be expanded to any number of servers simply by duplicating the sub-block, as shown in Figure 12–7).

- Critical to the functionality of each sub-block is the fact that the application servers are fully clustered and include session failover.
- The file system is duplicated on each instance, as each cluster member requires unique configuration files for Community-Gadgets, Shindig, and Visitors CAS.

> **Note:** The example in Figure 12–6 uses a shared file system. A shared file system is not required and the relevant directories can simply be copied, if that is preferred.

- A load balancer has been introduced in front of the cluster members to provide failover in case of a failure.

Figure 12–7 Community-Gadgets Production Block Consisting of Six Sub-Blocks



In Figure 12–7, the Community-Gadgets production block (Block A in Figure 12–3) is divided into six sub-blocks, on six servers. The Community-Gadgets production block consists of three independent clusters: one for Visitors CAS, one for Shindig, and one for Community-Gadgets. While this kind of breakdown is possible, it is not recommended. Typically, a cluster of two sub-blocks, as shown in Figure 12–6, provides the required capabilities and failover, and with less administrative overhead.

- In Figure 12–7, critical to the functionality of each sub-block is the fact that the application servers are fully clustered and include session failover.
- All servers access the same shared file system for configuration information.

> **Note:** The example in Figure 12–7 uses a shared file system. A shared file system is not required and the relevant directories can simply be copied over if that is preferred.

- Three load balancers have been introduced: one for Community-Gadgets, one for the Shindig application, and one for Visitors CAS.

13

Prerequisites for Installing Community-Gadgets

This chapter provides prerequisites for installing Community-Gadgets and lists the path and directory naming conventions used in this guide.

This chapter contains the following sections:

- Section 13.1, "Before Installing Community-Gadgets"
- Section 13.2, "Path and Directory Naming Conventions"

13.1 Before Installing Community-Gadgets

Before installing Community-Gadgets, complete the prerequisite steps in this section.

This section contains the following steps:

- Section 13.1.1, "Prerequisites for All Installations"
- Section 13.1.2, "Prerequisites for Clustered Environments"
- Section 13.1.3, "Prerequisites for LDAP Configuration"

13.1.1 Prerequisites for All Installations

- Installing Community-Gadgets requires expertise with WebCenter Sites, the WEM Framework, application servers, and the process of installing and configuring enterprise-level software.
- Review the *Oracle WebCenter Sites Certification Matrix* and Community-Gadgets release notes for the latest information about supported platforms for Community-Gadgets and installation procedures.
- Read this guide to determine your installation options:
  - Community-Gadgets can be installed to work with either its native identity provider (the database used by the WEM Framework) or an LDAP identity provider.
  - Community-Gadgets can be installed via the graphical installer or silent installer.
  - Community-Gadgets can be installed as a single node or in a cluster.
  - All mention of the Central Authentication Service (CAS) application on the production (delivery) server refers to Visitors CAS. Optionally, WebCenter Sites on the management and production systems can be integrated with Oracle Access Manager (OAM).

- Ensure that Community-Gadgets has a dedicated application server for each management node and each production node.

  > **Note:** Installing Community-Gadgets on a single application server is not supported.

- Community-Gadgets requires JDK 1.6 or a later version. Ensure that the `JAVA_HOME` environment variable is set to the path of JDK 1.6 or a later version which is listed in the *Oracle WebCenter Sites Certification Matrix*.

  > **Note:** On Unix/Linux/Solaris platforms, add the following JVM parameter to each Community-Gadgets application server if X Server is not present or is not configured correctly:
  > `-Djava.awt.headless=true`

- If you are using a read-only LDAP, then create the following role for Community-Gadgets:
  - Role: Moderator
  - Description: Site Moderator

  Other roles (such as GeneralAdmin, SiteAdmin, and Designer) used by Community-Gadgets exist in WebCenter Sites by default.

- Ensure that you have a fully functional WebCenter Sites environment, consisting of a content management system and a production (delivery) system where WebCenter Sites is integrated with CAS or OAM.

  > **Note:** After installing Community-Gadgets, you will configure Community-Gadgets to use the same CAS or OAM to communicate with WebCenter Sites.

- Prepare your WebCenter Sites systems to host the Community WEM application and the Gadgets WEM application:

  1. On the management WebCenter Sites, create or select the following content management sites:
     - `<app_site>`, to which you will assign the Community user interface and the Gadgets user interface once Community-Gadgets is installed. You can create or select multiple sites (Community user and Gadgets user interfaces can be assigned to any content management site and any number of sites).
     - `<base_site>`, to which you will assign the Gadgets Global Catalog interface, used to register gadgets for distribution to other content management sites. You will specify this site for Community-Gadgets during the Community-Gadgets installation process. Only one site can be chosen.
  2. Mirror publish the sites to the WebCenter Sites production system (before installing Community-Gadgets).

> **Note:** Identically named content management sites must be created to support communication between production Community-Gadgets and management Community-Gadgets through WebCenter Sites.
>
> For example:
>
> - On the management side, each `<app_site>` will be used to display the Community and Gadgets interfaces, where users can configure the appearance of widgets and gadgets, and set visitors' permissions to the widgets and gadgets. The same settings will also be applied on the production (delivery) side.
> - On the production (delivery) side, each `<app_site>` will be used to store visitor-generated content. On the management side, the same sites will display visitor-generated content in the Community interface for moderation and related tasks.
>
> Assigning the management Community WEM application or Gadgets WEM application to a site on the WebCenter Sites management system automatically assigns the production Community-Gadgets instance to the same site on the WebCenter Sites production system. Production Community-Gadgets can then communicate with the management Community-Gadgets.

- On the WebCenter Sites production system, enable searches as follows: Start the Lucene search engine. Configure search indexing, using the steps in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

13.1.2 Prerequisites for Clustered Environments

If you are installing a clustered environment application, ensure you have the load balancer configured.

13.1.3 Prerequisites for LDAP Configuration

If you are using an LDAP identity provider, complete the steps below:

1. Do one of the following:
   - If you wish to use a new LDAP server, install and configure a supported LDAP server.
   - If you are using an existing LDAP server, create a new Base `dn` to provide for Community-Gadgets visitors.
2. Create an LDIF file and replace `{ldap_basedn}` with the value for your system. You can import multiple users by copying the # add user entry for each user.

For example:

```
dn: {ldap_basedn}
objectclass: dcObject
objectclass: organization
dc: oracle
description: OpenLDAP pre_cg_setup
o: Oracle Software

# add user
dn: cn=<user_name>,{ldap_basedn}
objectClass: inetOrgPerson
```

```
objectClass: top
userPassword: <user_password>
cn: <user_name>
sn: <user_name>
displayName: <user_name>
mail: <user_email>
description: <user_description>
```

13.2 Path and Directory Naming Conventions

Table 13–1 defines path and directory naming conventions used in Community-Gadgets installation procedures.

Table 13–1 Path and Directory Names Used in This Guide

Convention	Description
`<cg_install_dir>`	Path to the directory where Community-Gadgets will be installed.
`<WL_HOME>`	Path to the directory where WebLogic is installed. The path includes the name of the directory.
`<TC_HOME>`	Path to the directory where Tomcat is installed. The path includes the name of the directory.
`<WS_HOME>`	Path to the directory where WebSphere is installed. The path includes the name of the directory.
`<ServerA>`	Represents server 'A' with unique host name and IP address.
`<ServerB>`	Represents server 'B' with unique host name and IP address.
`<cgA_`*`Mi`*`>`	Path to the application server on which management Community-Gadgets (including Shindig) will be deployed. ■ 'A' stands for the server on which the application server is installed (in this case, `<ServerA>`). ■ 'M' stands for management Community-Gadgets. ■ '`i`' stands for the *`i`*th instance of management Community-Gadgets.
`<cgB_`*`Pi`*`>`	Path to the application server instance on which production Community-Gadgets (including Shindig and Visitors CAS) will be deployed. ■ 'B' stands for the server on which the application server is installed (in this case, `<ServerB>`). ■ 'P' stands for production Community-Gadgets. ■ '`i`' stands for the *`i`*th instance of production Community-Gadgets.
`<base_site>`	The content management site, which will be used for the Gadgets Global Catalog interface. And for storing visitors profiles data. It is specified as "Community-Gadgets application site" during the Community-Gadgets installation process on one of the installer screens. Only one site can be chosen.
`<app_site>`	The content management site, which will be used for the Community user interface and the Gadgets user interface. More than one site is supported.

14

Configuring Application Servers for Community-Gadgets

This chapter contains procedures for configuring application servers to support non-clustered and clustered deployments of the Community-Gadgets application.

This chapter contains the following sections:

14.1 Overview of Community-Gadgets Deployment

The Community-Gadgets application can be deployed in many ways. For illustration purposes, we use a dual-server configuration, as follows:

- **Non-clustered deployments**

  In this guide, we deploy Community-Gadgets on two servers. Management Community-Gadgets is deployed on `<ServerA>` and production Community-Gadgets is deployed on `<ServerB>`.

- **Clustered deployments**

  Community-Gadgets can be deployed as either a vertically or horizontally clustered application. In this guide, we vertically cluster Community-Gadgets as follows: management Community-Gadgets is deployed as a cluster on `<ServerA>`, and production Community-Gadgets is deployed as a cluster on `<ServerB>`. Horizontal clustering can be achieved by replicating the above configuration on as many additional servers as needed.

14.2 Configuring Application Servers for Non-Clustered Deployments

This section provides steps for configuring selected application servers to support the deployment of a non-clustered Community-Gadgets application.

This section covers the following configurations:

Note: In this section, you will configure an application server to support a non-clustered Community-Gadgets. As discussed in Chapter 12, "Overview of Community-Gadgets," you have multiple ways to configure such a non-clustered system. For illustration purposes, the steps in this section describe the deployment of the Community-Gadgets application on two servers, as explained in Section 14.1, "Overview of Community-Gadgets Deployment."

This section contains the following topics:

- Section 14.2.1, "Configuring Apache Tomcat"
- Section 14.2.2, "Configuring Oracle WebLogic Application Server"
- Section 14.2.3, "Configuring IBM WebSphere Application Server"

14.2.1 Configuring Apache Tomcat

In this section, you will configure Apache Tomcat to support deployment of a non-clustered Community-Gadgets application, as follows:

1. On `<ServerA>`:
   a. Create a Tomcat instance (`<cgA_M1>`) on which to deploy the management Community-Gadgets application.
   b. Edit `catalina.sh` in `<cgA_M1>/bin` by adding the following line after the first comment block:

      ```
      CATALINA_HOME=<cgA_M1>
      ```

2. On `<ServerB>`:
   a. Create a Tomcat instance (`<cgB_P1>`) on which to deploy the production Community-Gadgets application.
   b. Edit `catalina.sh` in `<cgB_P1>/bin` by adding the following line after the first comment block:

      ```
      CATALINA_HOME=<cgB_P1>
      ```

3. Continue to Chapter 15, "Generating Community-Gadgets Deployment Files."

14.2.2 Configuring Oracle WebLogic Application Server

In this section, you will configure WebLogic Application Server to support deployment of non-clustered Community-Gadgets.

Note: Machines and managed servers can be created as a part of the WebLogic domain configuration utility, or they can be created separately from the WebLogic Administration Console of the corresponding domain. If you need detailed steps on configuring WebLogic domains, refer to the Oracle WebLogic Application Server documentation.

To support the deployment of a non-clustered, management Community-Gadgets application, complete the following steps:

1. On <ServerA>, use the domain configuration utility to create a domain, a new machine, and a new managed server.

   For example, if you are using Linux:

   ```
   cd <WL_HOME>/wlserver_10.3/common/bin
   ./config.sh
   ```

   For reference, this guide uses the configurations described in Table 14–1, Table 14–2, and Table 14–3.

Table 14–1 WebLogic Domain Configuration Example for Management Community-Gadgets

Property	Value
Domain Name	<cgMgmt>
Admin Server	<ServerA>
Admin Server Port	For example: 7001 Or use your own value as appropriate for your configuration.

Table 14–2 WebLogic Machine Configuration Example for Management Community-Gadgets

Property	Value
Machine Name	<wlcgA_M>
Listen Address	<ServerA>
Listen Port	For example: 5556 Or use your own value as appropriate for your configuration.

Table 14–3 WebLogic Managed Server Configuration Example for Management Community-Gadgets

Property	Value
Managed Server	<cgA_M1>
Domain	<cgMgmt>
Listen Address	<ServerA>
Listen Port	For example: 7003 Or use your own value as appropriate for your configuration.
Machine	<wlcgA_M>

2. Similarly, configure WebLogic Application Server as follows to support deployment of a non-clustered, production Community-Gadgets application:

   On <ServerB>, use the domain configuration utility to create a domain, a new machine, and a new managed server.

   For reference, this guide uses the configurations described in Table 14–4, Table 14–5, and Table 14–6.

Table 14–4 WebLogic Domain Configuration Example for Production Community-Gadgets

Property	Value
Domain Name	`<cgProd>`
Admin Server	`<ServerB>`
Admin Server Port	For example: `7001` Or use your own value as appropriate for your configuration.

Table 14–5 WebLogic Machine Configuration Example for Production Community-Gadgets

Property	Value
Machine Name	`<wlcgB_P>`
Listen Address	`<ServerB>`
Listen Port	For example: `5556` Or use your own value as appropriate for your configuration.

Table 14–6 WebLogic Managed Server Configuration Example for Production Community-Gadgets

Property	Value
Managed Server	`<cgB_P1>`
Domain	`<cgProd>`
Listen Address	`<ServerB>`
Listen Port	For example: `7003` Or use your own value as appropriate for your configuration.
Machine	`<wlcgB_P>`

3. For each server in steps 1 and 2:

   a. Select **Enable Tunneling**.

   b. Select **None** for Hostname Verification.

4. For each domain, create a directory for staging the Community-Gadget application. In this guide, we created a directory named `applications` in the following paths:

```
<WL_HOME>/user_projects/domains/cgMgmt/applications
<WL_HOME>/user_projects/domains/cgProd/applications
```

5. Continue to Chapter 15, "Generating Community-Gadgets Deployment Files."

14.2.3 Configuring IBM WebSphere Application Server

In this section, you will configure IBM WebSphere Application Server to support deployment of a non-clustered Community-Gadgets application.

> **Note:** If you need detailed steps on configuring IBM WebSphere Application Server, refer to the vendor's documentation.

1. Create a node on which to deploy the management Community-Gadgets application, and federate that node to your Deployment Manager.

   If you do not have a Deployment Manager configured, then create a Deployment Manager and Application Server profile and federate the node to the Deployment Manager profile you just created.

   For reference, this guide uses the configurations described in Table 14–7 and Table 14–8.

Table 14–7 WebSphere Profile Configuration Example for Management Community-Gadgets

Property	Value
Deployment Manager Profile	`<Dmgr01>`
Host Server	`<ServerA>`
Deployment Manager Admin Server Port	For example: `9060` Or use your own value as appropriate for your configuration.

Table 14–8 WebSphere Application Server Configuration Example for Management Community-Gadgets

Property	Value
Application Server Name	`<cgA_M1>`
Profile	`<AppSrv01>`
Node	`<ServerA_Node01>`
Listen Address	`<ServerA>`
Listen Port	For example: `9080` Or use your own value as appropriate for your configuration.

2. Similarly, create a node on which to deploy the production Community-Gadgets application, and federate that node to your Deployment Manager.

   For reference, this guide uses the configurations described in Table 14–9 and Table 14–10.

Table 14–9 WebSphere Profile Configuration Example for Production Community-Gadgets

Property	Value
Deployment Manager Profile	`<Dmgr01>`
Host Server	`<ServerA>`
Deployment Manager Admin Server Port	For example: `9060` Or use your own value as appropriate for your configuration.

Table 14–10 WebApplication Server Profile Configuration Example for Production Community-Gadgets

Property	Value
Application Server Name	`<cgB_P1>`

Table 14–10 (Cont.) WebApplication Server Profile Configuration Example for Production Community-Gadgets

Property	Value
Profile	`<AppSrv02>`
Node	`<ServerB_Node02>`
Listen Address	`<ServerB>`
Listen Port	For example: `9080` Or use your own value as appropriate for your configuration.

3. Continue to Chapter 15, "Generating Community-Gadgets Deployment Files."

14.3 Configuring Application Servers for Clustered Deployments

This section describes the configuration of supported application servers to support clustered deployments of the Community-Gadgets application.

This section covers these clustered deployment configurations:

- Section 14.3.1, "Configuring Apache Tomcat (Clustered)"
- Section 14.3.2, "Configuring Oracle WebLogic Application Server (Clustered)"
- Section 14.3.3, "Configuring IBM WebSphere Application Server (Clustered)"

> **Note:** In this section, you will configure an application server to support a clustered Community-Gadgets installation. As discussed in Chapter 12, "Overview of Community-Gadgets," you have multiple ways to configure a clustered system. For illustration purposes, the steps in this section describe vertical clustering of the Community-Gadgets application, as explained in Section 14.1, "Overview of Community-Gadgets Deployment."

14.3.1 Configuring Apache Tomcat (Clustered)

In this section, you will configure Apache Tomcat to support deployment of a clustered Community-Gadgets application.

1. On `<ServerA>`:

   a. Create a Tomcat instance (`<cgA_M1>`) on which to deploy the management Community-Gadgets application. Edit `catalina.sh` in `<cgA_M1>/bin` by adding the following line after the first comment block:

   ```
   CATALINA_HOME=<cgA_M1>
   ```

   b. Create a Tomcat instance (`<cgA_M2>`) on which to deploy the management Community-Gadgets application. Edit `catalina.sh` in `<cgA_M2>/bin` by adding the following line after the first comment block:

   ```
   CATALINA_HOME=<cgA_M2>
   ```

   c. Edit the `server.xml` file ensure there are no port conflicts.

2. On `<ServerB>`:

a. Create a Tomcat instance (<cgB_P1>) on which to deploy the production Community-Gadgets application. Edit catalina.sh in <cgB_P1>/bin by adding the following line after the first comment block:

```
CATALINA_HOME=<cgB_P1>
```

b. Create a Tomcat instance (<cgB_P2>) on which to deploy the production Community-Gadgets application. Edit catalina.sh in <cgB_P2>/bin by adding the following line after the first comment block:

```
CATALINA_HOME=<cgB_P2>
```

c. Edit the server.xml file to ensure there are no port conflicts.

After you complete the above steps, your Tomcat configurations should look something like the example in Table 14–11.

Table 14–11 Tomcat Configuration Example for the Community-Gadgets Application

Host	Tomcat Instance	Port Number
<ServerA>	<cgA_M1>	8080
<ServerA>	<cgA_M2>	8180
<ServerB>	<cgB_P1>	8080
<ServerB>	<cgB_P2>	8180

3. Configure the application server cluster by editing the server.xml file. Follow the steps below for <ServerA> and <ServerB>.

a. For each cluster member, add clustering ability by adding the following code to the server.xml file (refer to Apache Tomcat documentation for information about the code):

```
<Cluster className=
"org.apache.catalina.ha.tcp.SimpleTcpCluster"
channelSendOptions="8">
<Manager className=
"org.apache.catalina.ha.session.DeltaManager"
expireSessionsOnShutdown="false"
notifyListenersOnReplication="true"/>

<Channel className=
"org.apache.catalina.tribes.group.GroupChannel">
<Membership className=
"org.apache.catalina.tribes.membership.McastService"
address="228.0.0.4"
port="45564"
frequency="500"
dropTime="3000"/>

<Receiver className=
"org.apache.catalina.tribes.transport.nio.NioReceiver"
address="auto"
port="4000"
autoBind="100"
selectorTimeout="5000"
maxThreads="6"/>

<Sender className=
"org.apache.catalina.tribes.transport.
```

```
ReplicationTransmitter">
<Transport className=
"org.apache.catalina.tribes.transport.nio.
PooledParallelSender"/>
</Sender>

<Interceptor className=
"org.apache.catalina.tribes.group.interceptors.
TcpFailureDetector"/>
<Interceptor className=
"org.apache.catalina.tribes.group.interceptors.
MessageDispatch15Interceptor"/>
</Channel>

<Valve className=
"org.apache.catalina.ha.tcp.ReplicationValve" filter=""/>
<Valve className=
"org.apache.catalina.ha.session.
JvmRouteBinderValve"/>

<Deployer className=
"org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/tmp/war-temp/"
deployDir="/tmp/war-deploy/"
watchDir="/tmp/war-listen/"
watchEnabled="false"/>

<ClusterListener className=
"org.apache.catalina.ha.session.
JvmRouteSessionIDBinderListener"/>
<ClusterListener className=
"org.apache.catalina.ha.session.
ClusterSessionListener"/>
</Cluster>
```

b. To enable load balancing using AJP, locate the Engine line in the server.xml file, then add a jvmRoute for the server instance. For example, to include a jvmroute in the line <Engine name="Catalina" defaultHost="localhost"> and name the server jvm1, add jvmRoute="jvm1" as follows: <Engine name="Catalina" defaultHost="localhost" **jvmRoute="jvm1"**>.

Note that the each server must have a unique jvmRoute name.

c. Verify that all Tomcat instances belonging to the same cluster have the same values for multicast address and port in the Membership tag. For example, <cgB_P1> and <cgB_P2> must have the same values for multicast address and port.

d. Verify that all Tomcat instances belonging to the different cluster have a multicast address or port number (in the Membership tag) that is unique across clusters. For example, all Tomcat instances in the <cgA_M1>/<cgA_M2> cluster must have a multicast address or port number that differs from the multicast address or port number of Tomcat instances in the <cgB_P1>/<cgB_P2> cluster.

e. For a vertical cluster, verify that all Tomcat instances located on the same server have different values for the port parameter in the Receiver tag.

f. If you are using a multi-homed server, enter the exact IP address or host name (instead of auto) as the value for the address parameter in the Receiver tag.

4. On all Tomcat instances, add the `<distributable/>` tag in the `<web-app>` tag to the `<TC_HOME>/conf/web.xml` file.
5. Continue to Chapter 15, "Generating Community-Gadgets Deployment Files."

14.3.2 Configuring Oracle WebLogic Application Server (Clustered)

In this section, you will configure WebLogic Application Server to support deployment of a clustered Community-Gadgets application.

> **Note:** Machines and managed servers can be created as a part of the WebLogic domain configuration utility, or they can be created separately from the WebLogic Administration Console of the corresponding domain. If you need detailed steps on configuring WebLogic domains, refer to the Oracle WebLogic Application Server documentation.

1. To support the deployment of a clustered, management Community-Gadgets application, do the following:

   On `<ServerA>`, use the domain configuration utility to create a domain, a new machine, new managed servers, and the cluster.

   For example, if you are using Linux:

   ```
   cd <WL_HOME>/wlserver_10.3/common/bin
   ./config.sh
   ```

   For reference, this guide uses the configurations described in Table 14–12, Table 14–13, Table 14–14, and Table 14–15.

Table 14–12 WebLogic Domain Configuration Example for Management Community-Gadgets (Clustered)

Property	Value
Domain Name	`<cgMgmt>`
Admin Server	`<ServerA>`
Admin Server Port	For example: `7001` Or use your own value as appropriate for your configuration.

Table 14–13 WebLogic Machine Configuration Example for Management Community-Gadgets (Clustered)

Property	Value
Machine Name	`<wlcgA_M>`
Listen Address	`<ServerA>`
Listen Port	For example: `5556` Or use your own value as appropriate for your configuration.

Table 14–14 WebLogic Managed Server Configuration Example for Management Community-Gadgets (Clustered)

Property	Value
Managed Server 1	`<cgA_M1>`
Domain	`<cgMgmt>`
Listen Address	`<ServerA>`
Listen Port	For example: `7003` Or use your own value as appropriate for your configuration.
Machine	`<wlcgA_M>`
Managed Server 2	`<cgA_M2>`
Domain	`<cgMgmt>`
Listen Address	`<ServerA>`
Listen Port	For example: `7005` Or use your own value as appropriate for your configuration.
Machine	`<wlcgA_M>`

Table 14–15 WebLogic Cluster Configuration Example for Management Community-Gadgets (Clustered)

Property	Value
Cluster Name	`<wlcg_M>`
Cluster Address	`<ServerA>`
Cluster Messaging Mode	Unicast (or Multicast depending on your environment)
Managed Servers as part of Cluster	`<cgA_M1>`, `<cgA_M2>`

2. Similarly, configure WebLogic Application Server to support the deployment of a clustered production Community-Gadgets application.

   On `<ServerB>`, use the domain configuration utility to create a domain, a new machine, new managed servers, and the cluster.

   For reference, this guide uses the configurations described in Table 14–16, Table 14–17, Table 14–18, and Table 14–19.

Table 14–16 WebLogic Domain Configuration Example for Production Community-Gadgets (Clustered)

Property	Value
Domain Name	`<cgProd>`
Admin Server	`<ServerB>`
Admin Server Port	For example: `7001` Or use your own value as appropriate for your configuration.

Table 14–17 WebLogic Machine Configuration Example for Production Community-Gadgets (Clustered)

Property	Value
Machine Name	`<wlcgB_P>`
Listen Address	`<ServerB>`
Listen Port	For example: `5556` Or use your own value as appropriate for your configuration.

Table 14–18 WebLogic Managed Server Configuration Example for Production Community-Gadgets (Clustered)

Property	Value
Managed Server 1	`<cgB_P1>`
Domain	`<cgProd>`
Listen Address	`<ServerB>`
Listen Port	For example: `7003` Or use your own value as appropriate for your configuration.
Machine	`<wlcgB_P>`
Managed Server 2	`<cgB_P2>`
Domain	`<cgProd>`
Listen Address	`<ServerB>`
Listen Port	For example: `7005` Or use your own value as appropriate for your configuration.
Machine	`<wlcgB_P>`

Table 14–19 WebLogic Cluster Configuration Example for Production Community-Gadgets (Clustered)

Property	Value
Cluster Name	`<wlcg_P>`
Cluster Address	`<ServerB>`
Cluster Messaging Mode	Unicast (or Multicast depending on your environment)
Managed Servers as part of Cluster	`<cgB_P1>`, `<cgB_P2>`

3. For each managed server created on `<ServerA>` and `<ServerB>`:
   a. Select **Enable Tunneling**.
   b. Select **None** for Hostname Verification.
4. For each domain, create a directory for staging the Community application. In this guide, we created a directory named `applications` in the following paths:

```
<WL_HOME>/user_projects/domains/cgMgmt/applications
<WL_HOME>/user_projects/domains/cgProd/applications
```

5. For each cluster, go to its Advanced Cluster Configuration section and select the **WebLogic Plug-In Enabled** parameter.

6. Continue to Chapter 15, "Generating Community-Gadgets Deployment Files."

14.3.3 Configuring IBM WebSphere Application Server (Clustered)

In this section, you will configure WebSphere Application Server to support deployment of a clustered Community-Gadgets application.

> **Note:** If you need detailed steps on configuring IBM WebSphere Application Server, or details on creating profiles or federating nodes, refer to the vendor's documentation.

1. Create a node on which to deploy the management Community-Gadgets application, and federate that node to your Deployment Manager.

   If you do not have a Deployment Manager configured, then create a Deployment Manager and Application Server profile and federate the node to the Deployment Manager profile you just created.

   For reference, this guide uses the configurations described in Table 14–20 and Table 14–21.

Table 14–20 WebSphere Profile Configuration Example for Management Community-Gadgets (Clustered)

Property	Value
Deployment Manager Profile	`<Dmgr01>`
Host Server	`<ServerA>`
Deployment Manager Admin Server Port	For example: `9060` Or use your own value as appropriate for your configuration.

Table 14–21 WebSphere Application Server Configuration Example for Management Community-Gadgets (Clustered)

Property	Value
Application Server Name	`<cgA_M1>`
Profile	`<AppSrv01>`
Node	`<ServerA_Node01>`
Listen Address	`<ServerA>`
Listen Port	For example: `9080` Or use your own value as appropriate for your configuration.
Application Server Name	`<cgA_M2>`
Profile	`<AppSrv01>`
Node	`<ServerA_Node01>`
Listen Address	`<ServerA>`
Listen Port	For example: `9081` Or use your own value as appropriate for your configuration.

2. Similarly, create a node on which to deploy the production Community-Gadgets application, and federate that node to your Deployment Manager.

   For reference, this guide uses the configurations described in Table 14–22 andTable 14–23.

Table 14–22 WebSphere Profile Configuration Example for Production Community-Gadgets (Clustered)

Property	Value
Deployment Manager Profile	`<Dmgr01>`
Host Server	`<ServerA>`
Deployment Manager Admin Server Port	For example: `9060` Or use your own value as appropriate for your configuration.

Table 14–23 WebSphere Application Server Configuration Example for Production Community-Gadgets (Clustered)

Property	Value
Application Server 1 Name	`<cgB_P1>`
Profile	`<AppSrv02>`
Node	`<ServerB_Node02>`
Listen Address	`<ServerB>`
Listen Port	For example: `9080` Or use your own value as appropriate for your configuration.
Application Server 2 Name	`<cgB_P2>`
Profile	`<AppSrv02>`
Node	`<ServerB_Node02>`
Listen Address	`<ServerB>`
Listen Port	For example: `9081` Or use your own value as appropriate for your configuration.

3. Create two new clusters (see Table 14–24), using the WebSphere Administration Console.

   For example:

   ```
   http://<ServerA>:9060/ibm/console
   ```

Table 14–24 WebSphere Cluster Configuration Examples for Community-Gadgets

Configuration	Property	Value
Cluster 1	Cluster Name	`<wscg_M>`
Cluster 1	Configure HTTP memory-to-memory replication	`Yes`
Cluster 1	Cluster Members	`<cgA_M1>,<cgA_M2>`
Cluster 2	Cluster Name	`<wscg_P>`

Table 14–24 (Cont.) WebSphere Cluster Configuration Examples for

Configuration	Property	Value
Cluster 2	Configure HTTP memory-to-memory replication	`Yes`
Cluster 2	Cluster Members	`<cgB_P1>, <cgB_P2>`

4. For each managed server created on `<ServerA>` and `<ServerB>`, complete the following steps in the WebSphere Administration Console:

   a. Set memory-to-memory replication.

   b. Under Web Container Settings, add the following custom property:

   ```
   Name: HttpSessionCloneId
   Value: 11111111 (8-9 characters, unique for each managed server)
   ```

5. If you wish to customize replication domains, use the administrative console (select **Environment**, then select **Replication domains** in the left frame).

6. Add ports for all cluster members (go to **Environment**, then select **Virtual hosts**, then select **default_host**, then select **Host Aliases** and add the ports).

7. Continue to Chapter 15, "Generating Community-Gadgets Deployment Files."

15 Generating Community-Gadgets Deployment Files

This chapter describes how to generate the Community-Gadgets war/ ear files both graphically and silently.

This chapter contains the following sections:

- Section 15.1, "Generating the Community-Gadgets Deployment Files Graphically"
- Section 15.2, "Generating the Community-Gadgets Deployment Files Silently"
- Section 15.3, "Verifying the Community-Gadgets Installation Directory Structure"

15.1 Generating the Community-Gadgets Deployment Files Graphically

The graphical installer is a configuration wizard that creates the Community-Gadgets installation directory and generates war/ear files for deployment.

Follow these steps to generate the Community-Gadgets deployment files graphically:

> **Note:** This section also provides the steps you need to take if your management instance of WebCenter Sites is integrated with OAM.

1. Download the Community-Gadgets distribution zip file to your server and extract it into a temporary directory. The extracted folder contains the cgInstall.bat and cgInstall.sh files, which are needed to run the Community-Gadgets installer.
2. Create an installation directory in which the Community-Gadgets installer will create the necessary folders and files.
3. Set JAVA_HOME to the path of JDK 1.6 or a later version.

   For example:

   ```
   export JAVA_HOME=/opt/jdk1.6.0_43
   ```

4. On remote UNIX environments, set the DISPLAY environment variable.

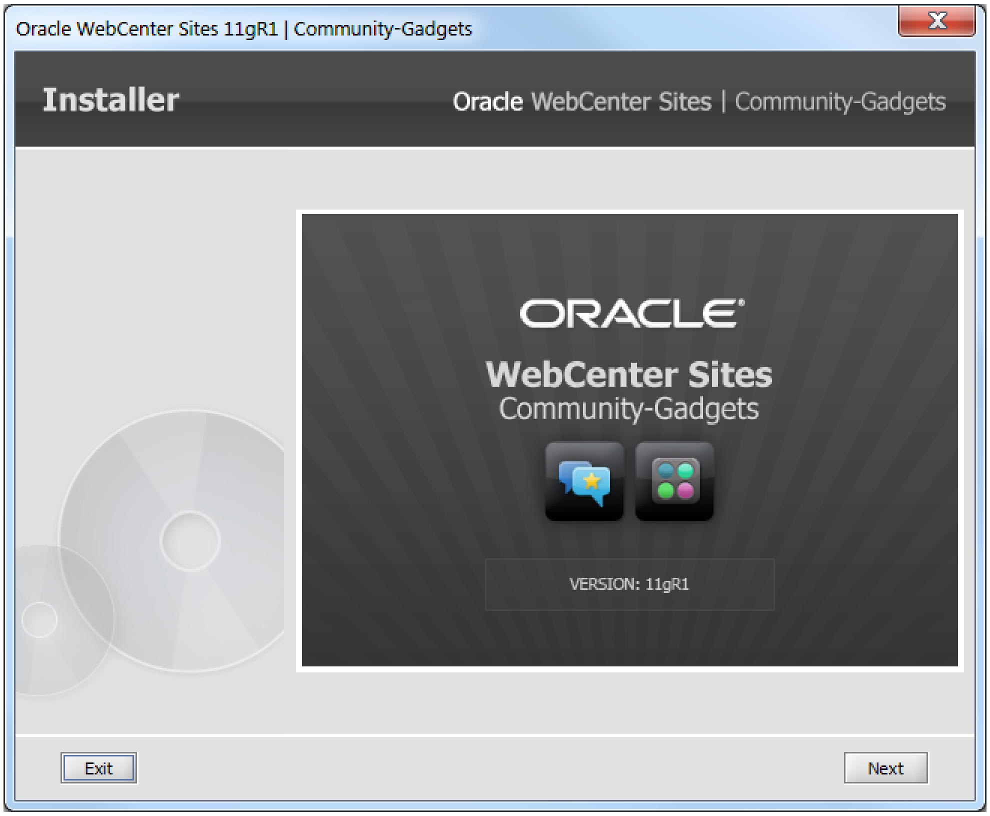
   For example:

   ```
   export DISPLAY=10.120.15.20:0.0
   ```

5. Execute the installer script:
   - On Windows: cgInstall.bat
   - On UNIX: cgInstall.sh

6. The first screen displayed to you is the welcome screen (Figure 15–1). Click **Next**.

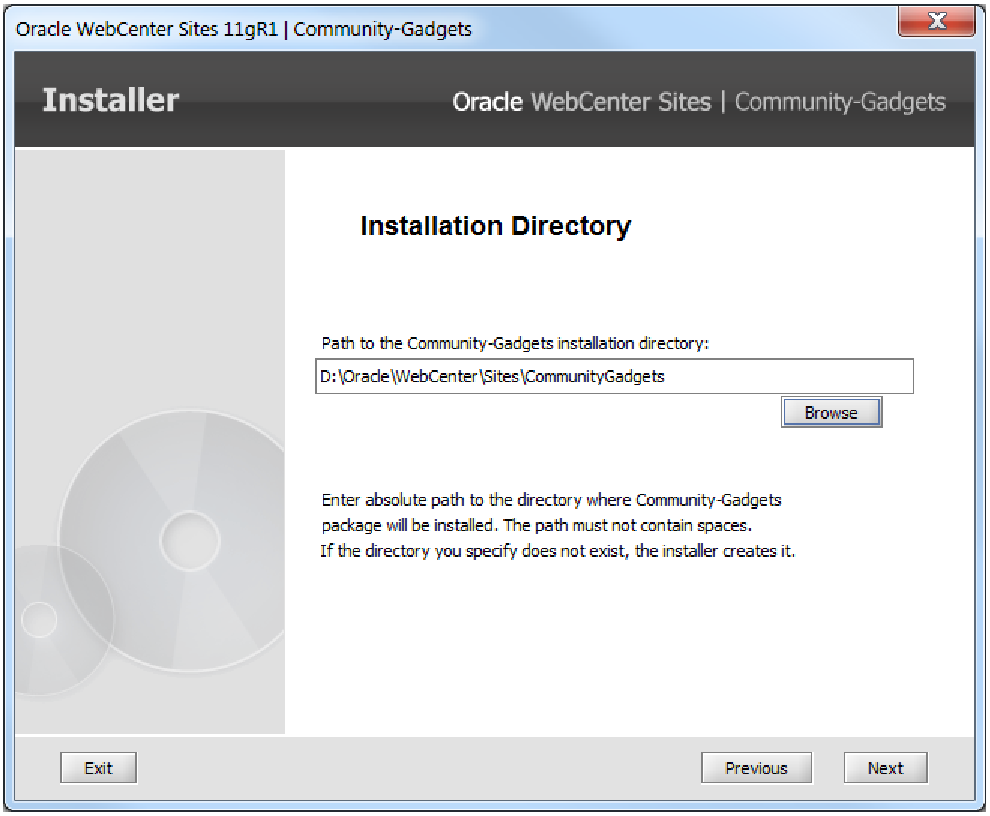
Figure 15–1 Installer Welcome


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
ORACLE
WebCenter Sites
Community-Gadgets
VERSION: 11gR1
Exit
Next

7. On the Installation Directory screen (Figure 15–2), specify the path to the directory (created in step 2) where the Community-Gadgets application will be installed, or click **Browse** and navigate to a directory.

   The path must not contain spaces. If the directory you specify does not exist, the installer creates it.

Figure 15–2 Installation Directory


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Browse
Exit
Previous
Next

Installation Directory

Path to the Community-Gadgets installation directory:

D:\Oracle\WebCenter\Sites\CommunityGadgets

Enter absolute path to the directory where Community-Gadgets package will be installed. The path must not contain spaces.
If the directory you specify does not exist, the installer creates it.

8. On the Management CG Information screen (Figure 15–3):

   - Select the **http** or **https** (HTTP over SSL) protocol.
   - Enter the host name or IP address of the management Community-Gadgets application server. If installing a cluster, enter the host name or IP address of the load balancer.

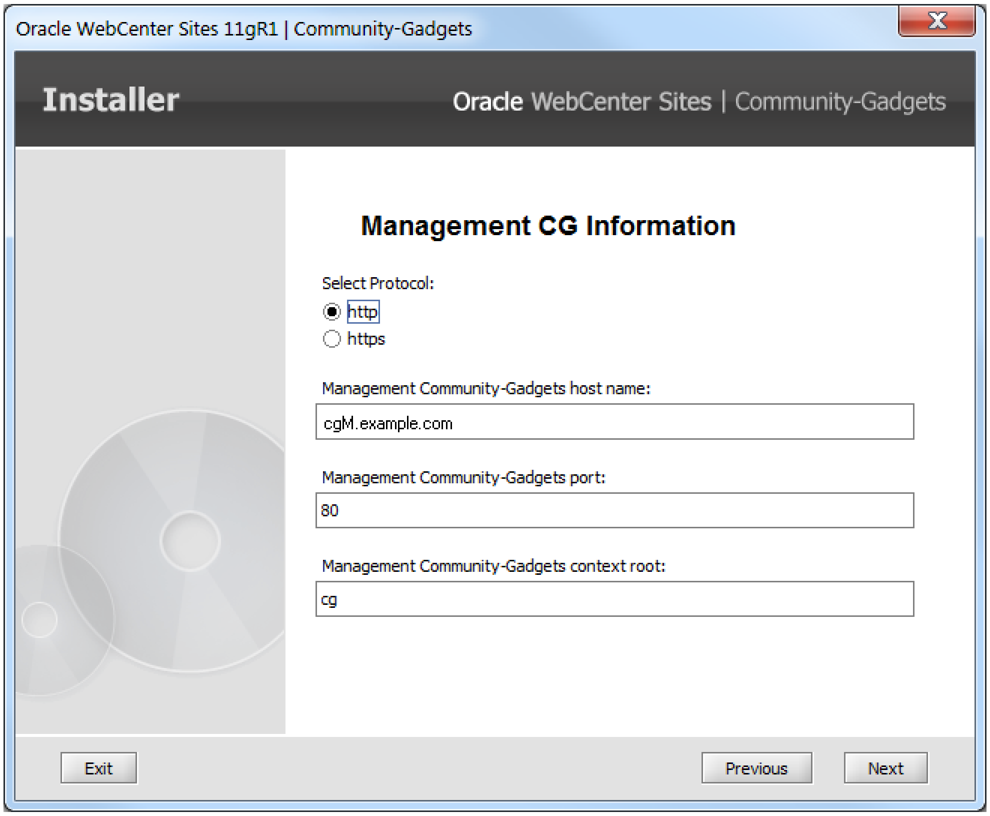
     Note: If WebCenter Sites on the management system is integrated with Oracle Access Manager, specify the name or IP address of the Oracle HTTP Server host.

   - Enter the port number of the management Community-Gadgets application server. If installing a cluster, enter the port number of the load balancer.

     Note: If WebCenter Sites on the management system is integrated with Oracle Access Manager, specify the port of the Oracle HTTP Server.

   - Enter the management Community-Gadgets application context root.

Figure 15–3 Management CG Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets

Management CG Information

Select Protocol:

● http

○ https

Management Community-Gadgets host name: cgM.example.com

Management Community-Gadgets port: 80

Management Community-Gadgets context root: cg

9. On the Management Shindig Information screen (Figure 15–4):

   - Select the **http** or **https** (HTTP over SSL) protocol.
   - Enter the host name or IP address of the management Shindig application server. If installing a cluster, enter the host name or IP address of the load balancer.

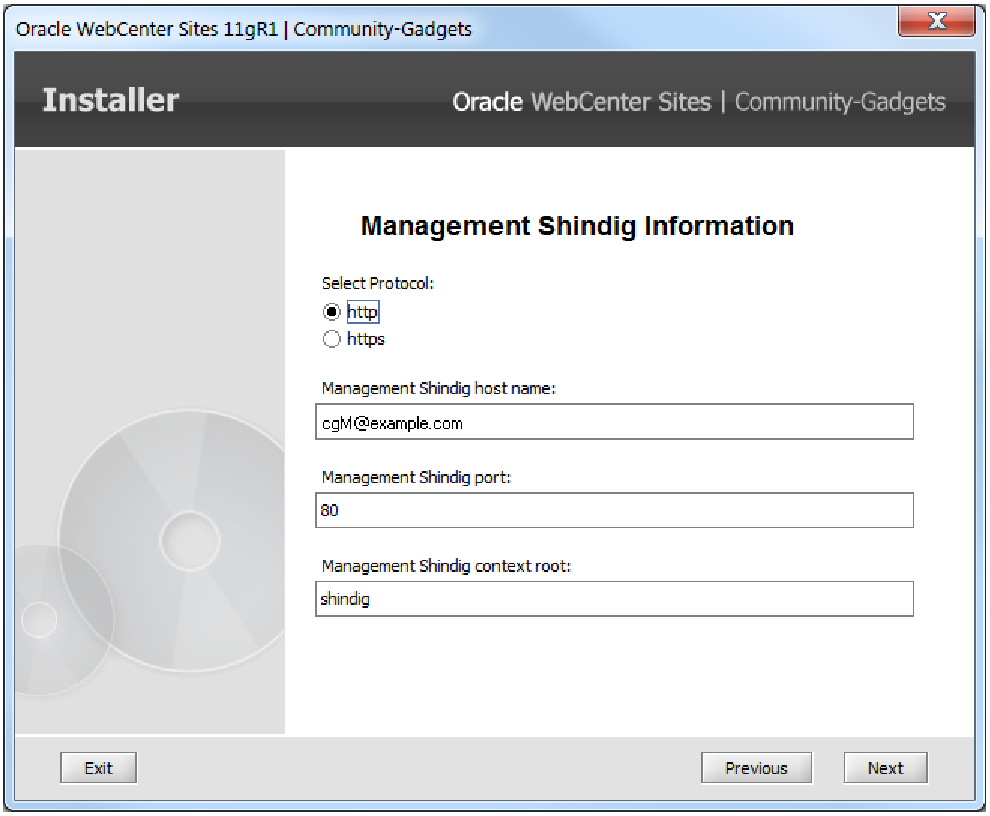
     Note: If WebCenter Sites on the management system is integrated with Oracle Access Manager, specify the name or IP address of the Oracle HTTP Server host.

   - Enter the port number of the management Shindig application server. If installing a cluster, enter the port number of the load balancer.

     Note: If WebCenter Sites on the management system is integrated with Oracle Access Manager, specify the port of the Oracle HTTP Server.

   - Enter the management Shindig application context root.

Figure 15–4 Management Shindig Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets

Installer

Oracle WebCenter Sites | Community-Gadgets

Management Shindig Information

Select Protocol:

● http

○ https

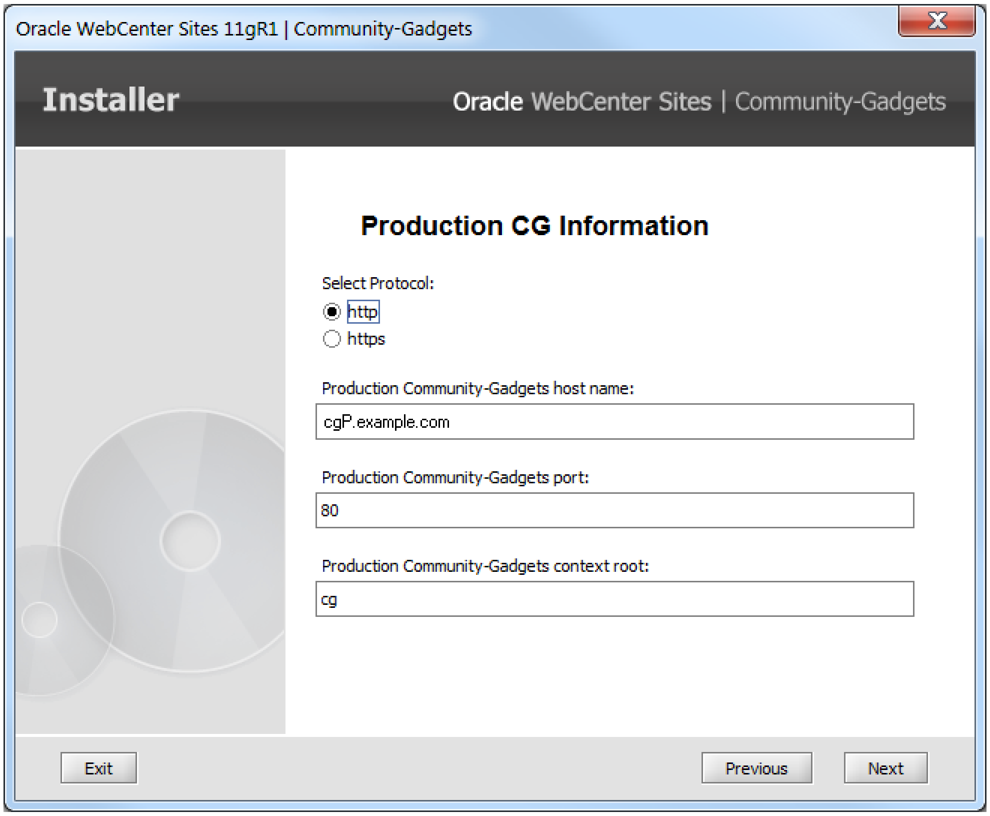
Management Shindig host name: cgM@example.com

Management Shindig port: 80

Management Shindig context root: shindig

10. On the Production CG Information screen (Figure 15–5):
    - Select the **http** or **https** (HTTP over SSL) protocol.
    - Enter the host name or IP address of the production Community-Gadgets application server. If installing a cluster, enter the host name or IP address of the load balancer.
    - Enter the port number of the production Community-Gadgets application server. If installing a cluster, enter the port number of the load balancer.
    - Enter the production Community-Gadgets application context root.

Figure 15–5 Production CG Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets

Production CG Information

Select Protocol:

● http

○ https

Production Community-Gadgets host name: cgP.example.com

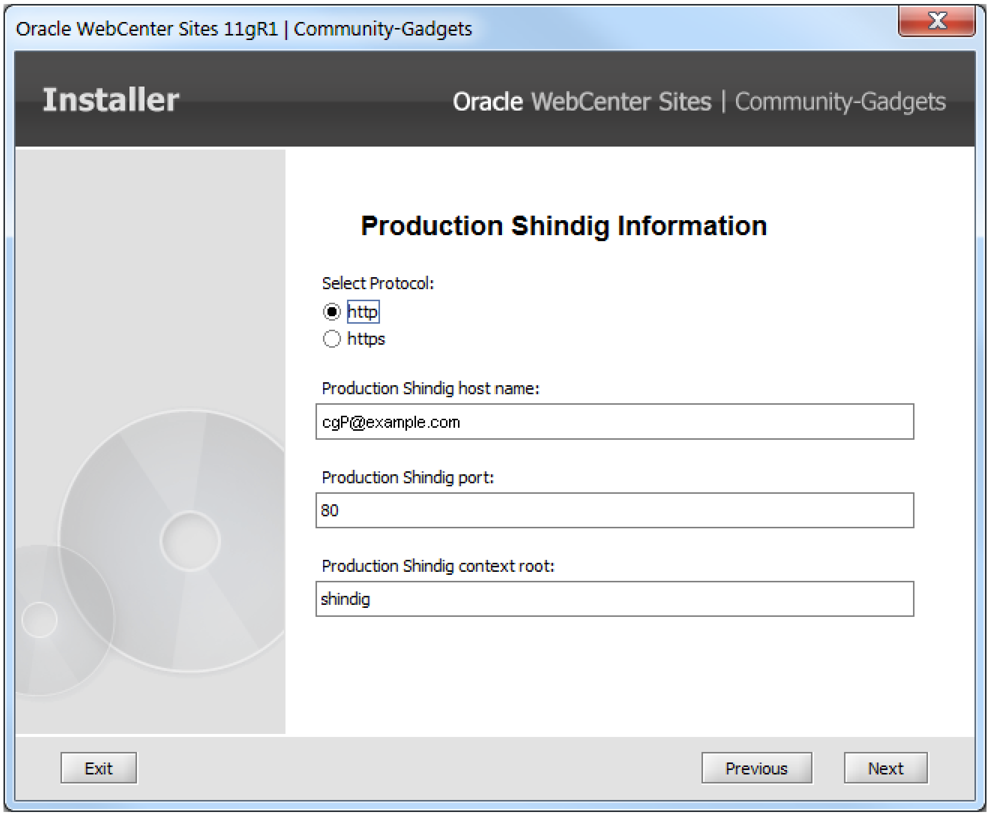
Production Community-Gadgets port: 80

Production Community-Gadgets context root: cg

11. On the Production Shindig Information screen (Figure 15–6):

    - Select the http or https (HTTP over SSL) protocol.
    - Enter the host name or IP address of the production Shindig application server. If installing a cluster, enter the host name or IP address of the load balancer.
    - Enter the port number of the production Shindig application server. If installing a cluster, enter the port number of the load balancer.
    - Enter the production Shindig application context root.

Figure 15–6 Production CG Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets

Installer

Oracle WebCenter Sites | Community-Gadgets

Production Shindig Information

Select Protocol:

● http

○ https

Production Shindig host name: cgP@example.com

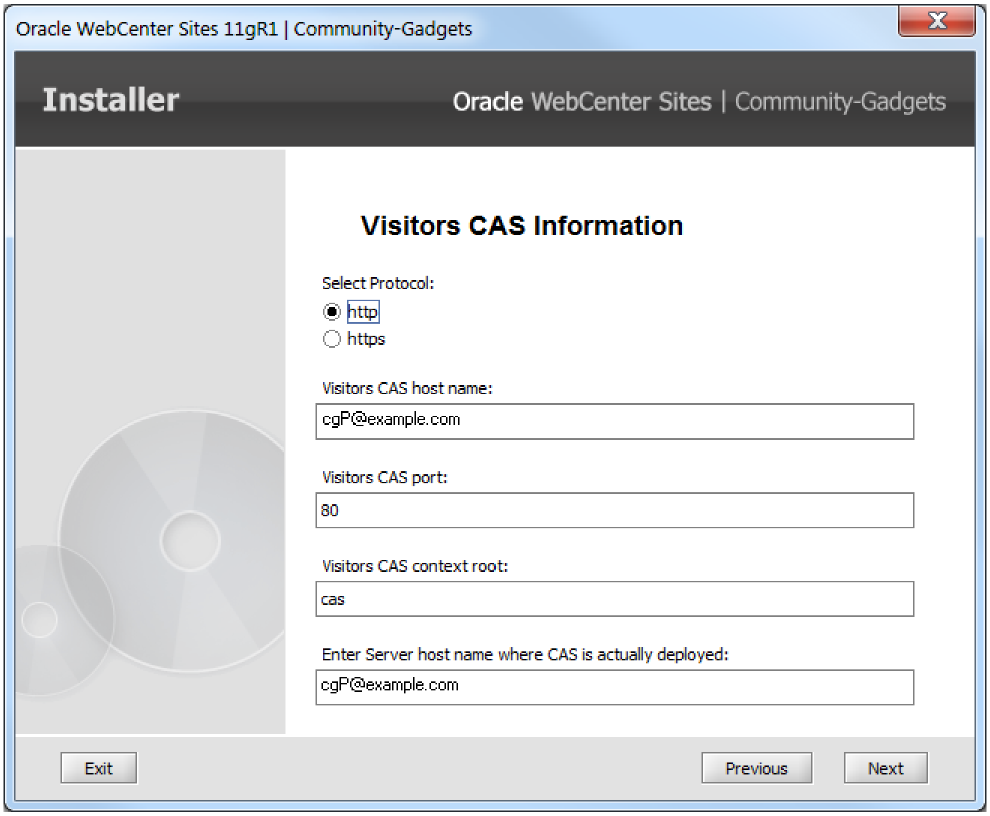
Production Shindig port: 80

Production Shindig context root: shindig

12. On the Visitors CAS Server Information screen (Figure 15–7):

    - Select the **http** or **https** (HTTP over SSL) protocol.
    - Enter the host name or IP address of the Visitors CAS application server. If installing a cluster, enter the host name or IP address of the load balancer.
    - Enter the port number of the Visitors CAS application server. If installing a cluster, enter the port number of the load balancer.
    - Enter the Visitors CAS application context root.
    - Enter the host name or IP address of the server where Visitors CAS will be deployed.
        - For a non-clustered installation, the value of this field is identical to the value of the field above (Visitors CAS host name).
        - For a clustered installation, the value of this field is the host name of the Visitors CAS application server. (Ensure that the value of the above field, Visitors CAS host name, is set to the host name or IP address of the load balancer.)

Figure 15–7 Visitor CAS Server Information


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Visitors CAS Information

Select Protocol:

● http

○ https

Visitors CAS host name: cgP@example.com

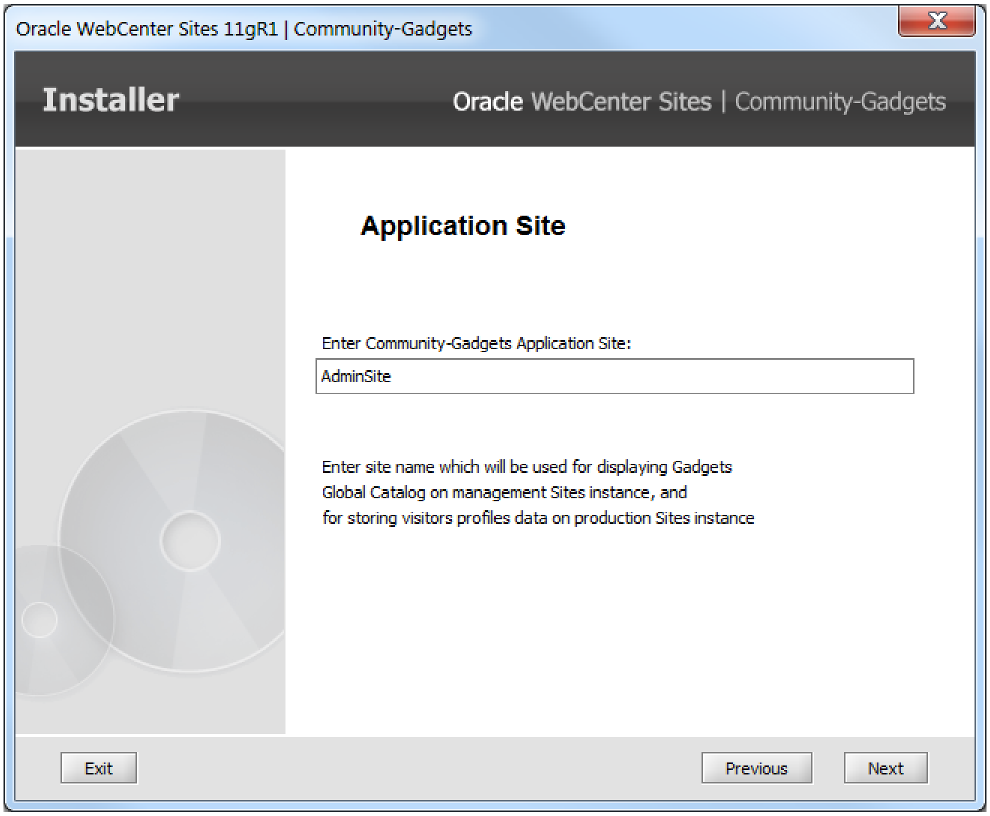
Visitors CAS port: 80

Visitors CAS context root: cas

Enter Server host name where CAS is actually deployed: cgP@example.com

13. On the Application Site screen (Figure 15–8), enter the name of the `<base_site>` that will be used for displaying the Gadgets Global Catalog interface on the management WebCenter Sites instance, and for storing visitors profiles data on the production Sites instance. (The Gadgets user interface and Community user interface can be available on any site, `<app_site>`.)

Figure 15–8 Application Site


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Application Site

Enter Community-Gadgets Application Site:

AdminSite

Enter site name which will be used for displaying Gadgets
Global Catalog on management Sites instance, and
for storing visitors profiles data on production Sites instance

14. On the Management Sites Information screen (Figure 15–9):

- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the management WebCenter Sites application server.

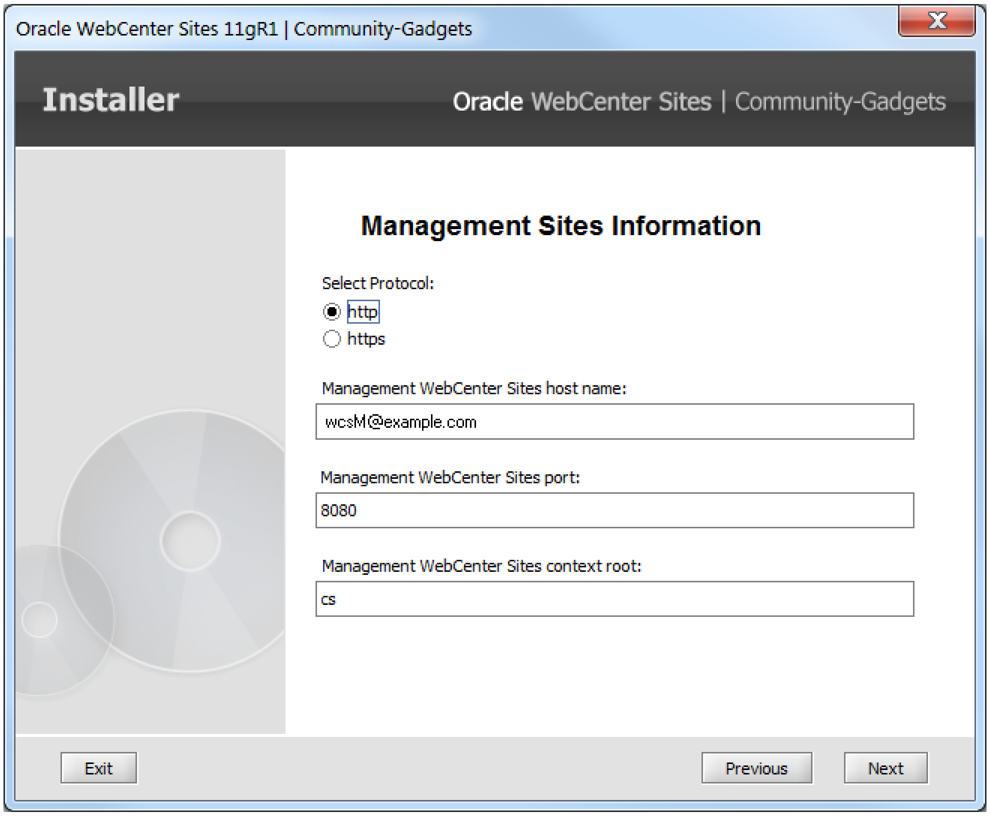
  Note: If WebCenter Sites on the management system is integrated with Oracle Access Manager, specify the name or IP address of the Oracle HTTP Server host.

- Enter the port number of the management WebCenter Sites application server.

  Note: If WebCenter Sites on the management system is integrated with Oracle Access Manager, specify the port of the Oracle HTTP Server.

- Enter the management WebCenter Sites application context root.

Figure 15–9 Management Sites Information


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Management Sites Information

Select Protocol:

● http

○ https

Management WebCenter Sites host name: wcsM@example.com

Management WebCenter Sites port: 8080

Management WebCenter Sites context root: cs

15. On the Management Sites Admin User screen (Figure 15–10):

    - Enter your general administrator user name, which you use to log in to the WebCenter Sites management system.

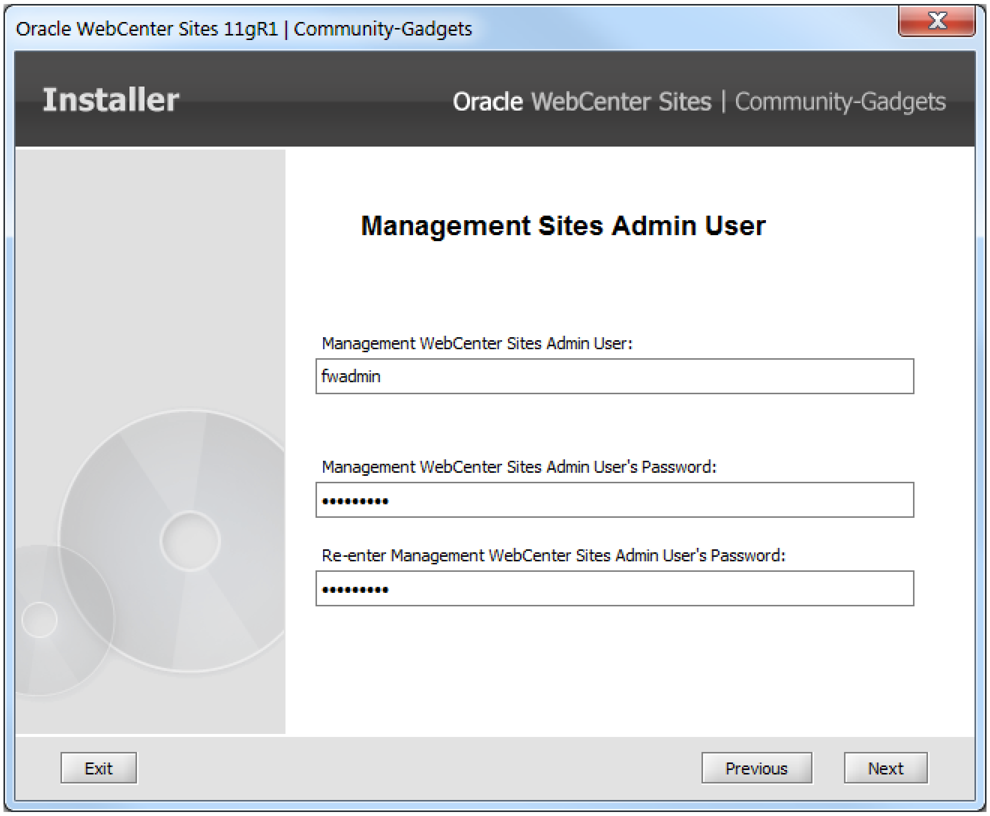
      The user name is case-sensitive and allows from 8 to 64 alphanumeric characters. The default user is: `fwadmin`

    - Enter your administrator password.

      The password is case-sensitive and allows from 8 to 64 alphanumeric characters.

    - For verification re-enter your password.

Figure 15–10 Management Sites Admin User


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Management Sites Admin User

Management WebCenter Sites Admin User: fwadmin

Management WebCenter Sites Admin User's Password: •••••••••

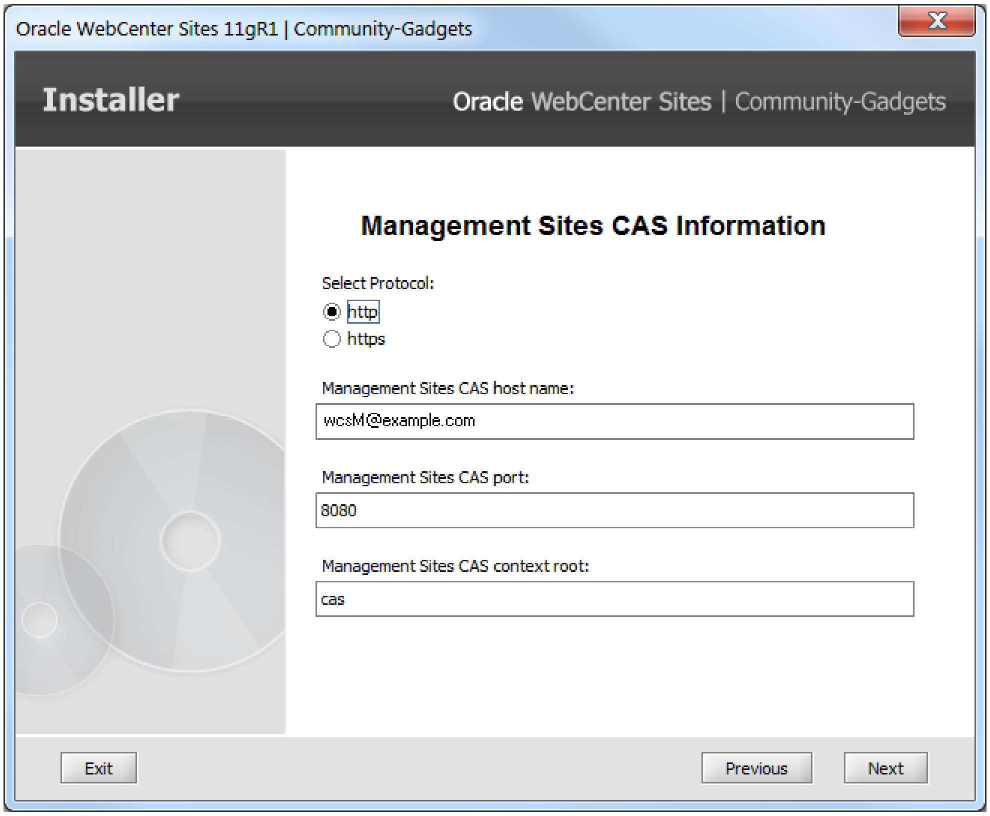
Re-enter Management WebCenter Sites Admin User's Password: •••••••••

16. On the Management Sites CAS Information screen (Figure 15–11):

> **Note:** If WebCenter Sites on the management system is integrated with Oracle Access Manager, accept the default values, click **Next**, and continue to step 17.

- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the management Sites CAS application server.
- Enter the port number of the management Sites CAS application server.
- Enter the management Sites CAS application context root.

Figure 15–11 Management Sites CAS Information


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Management Sites CAS Information

Select Protocol:
● http
○ https

Management Sites CAS host name: wcsM@example.com

Management Sites CAS port: 8080

Management Sites CAS context root: cas

17. On the Production Sites Information screen (Figure 15–12):

- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the production WebCenter Sites application server.

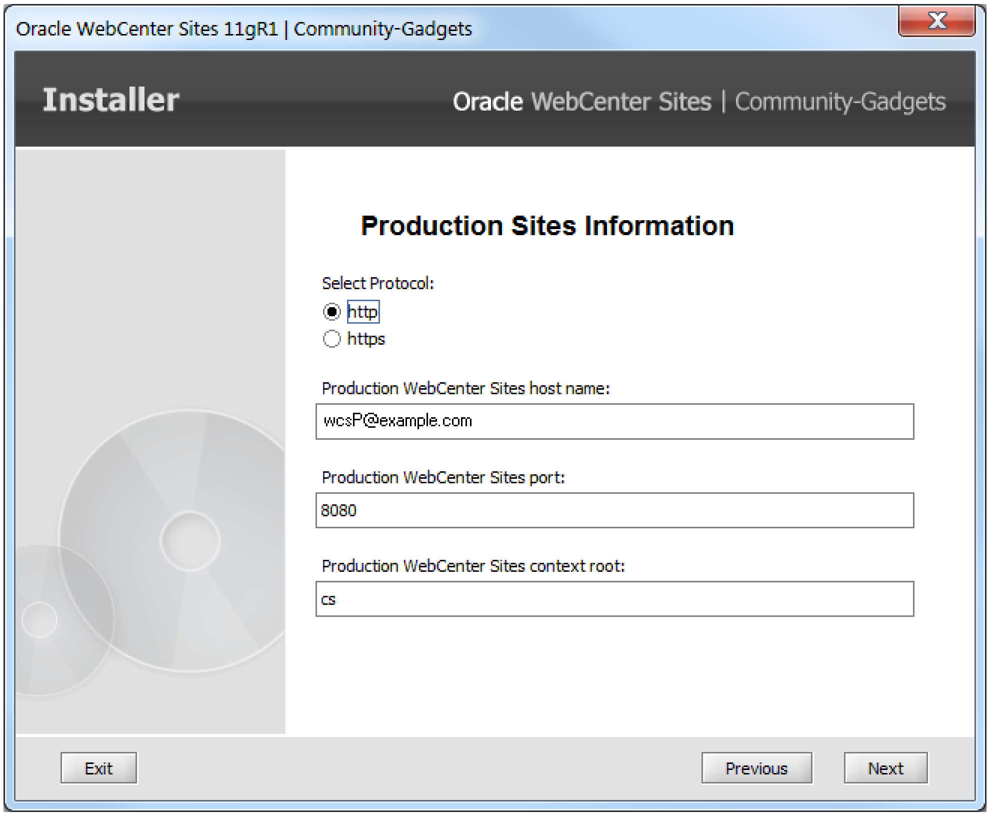
> **Note:** If WebCenter Sites on the production system is integrated with Oracle Access Manager, specify the name or IP address of the Oracle HTTP Server host.

- Enter the port number of the production WebCenter Sites application server.

> **Note:** If WebCenter Sites on the production system is integrated with Oracle Access Manager, specify the port of the Oracle HTTP Server host.

- Enter the production WebCenter Sites application context root.

Figure 15–12 Production Sites Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets

Production Sites Information

Select Protocol:

● http

○ https

Production WebCenter Sites host name: wcsP@example.com

Production WebCenter Sites port: 8080

Production WebCenter Sites context root: cs

18. On the Production Sites Admin User screen (Figure 15–13):

- Enter your general administrator user name, which you use to log in to the WebCenter Sites production system.

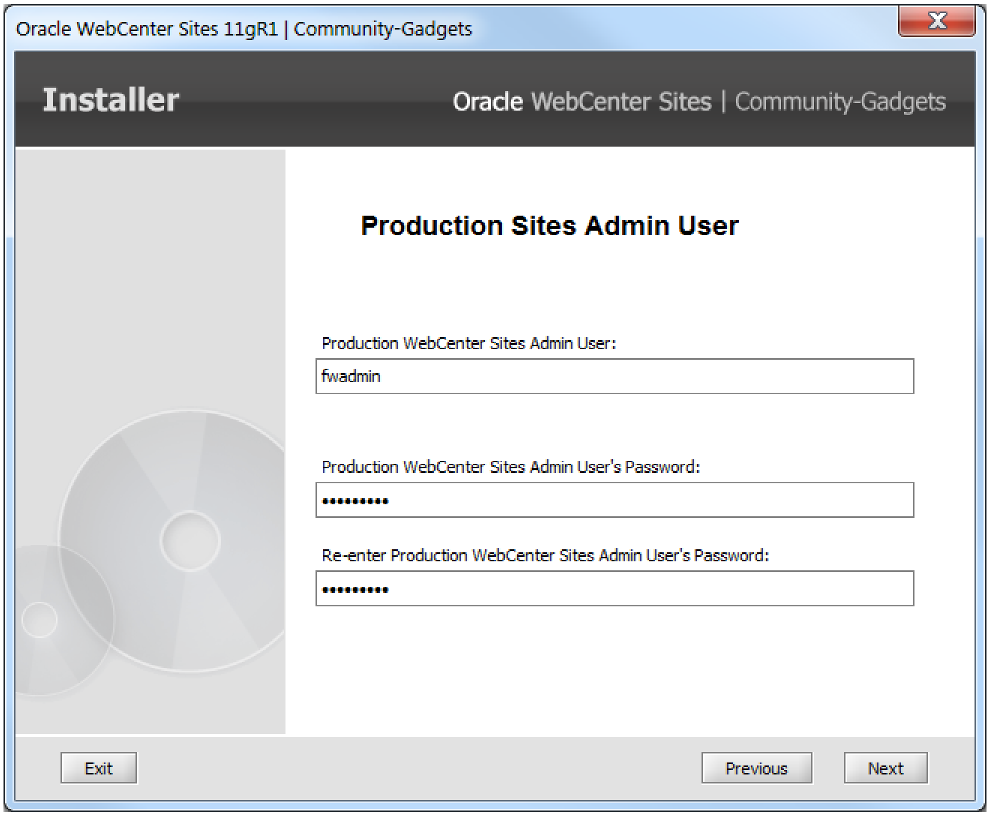
  The user name is case-sensitive and allows from 8 to 64 alphanumeric characters. The default user: `fwadmin`

- Enter your administrator password.

  The password is case-sensitive and allows from 8 to 64 alphanumeric characters.

- For verification, re-enter your password.

Figure 15–13 Production Sites Admin User



Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets

Production Sites Admin User

Production WebCenter Sites Admin User: fwadmin

Production WebCenter Sites Admin User's Password: •••••••••

Re-enter Production WebCenter Sites Admin User's Password: •••••••••

19. On the Production Sites Satellite User screen (Figure 15–14):

- Enter the user name for the Satellite Server that is configured for the production WebCenter Sites application.

  The user name is case-sensitive and allows from 8 to 64 alphanumeric characters. The default user is: `SatelliteServer`

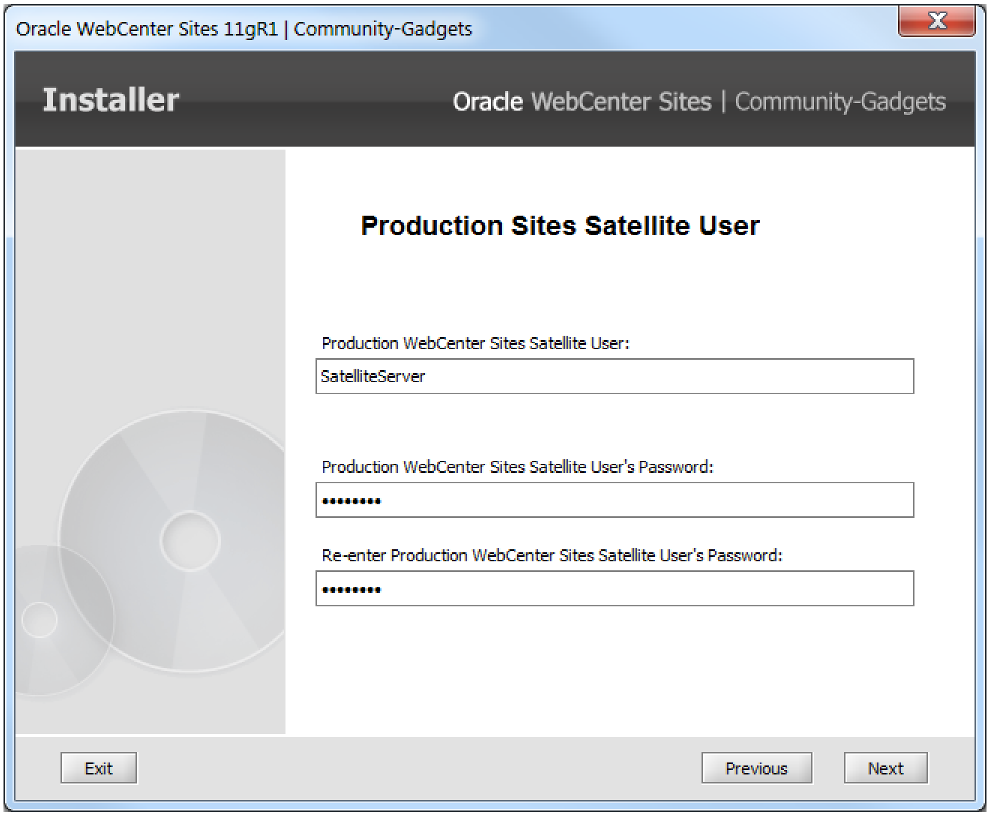
- Enter the Satellite Server user's password.

  The password is case-sensitive and allows from 8 to 64 alphanumeric characters.

- For verification, re-enter the password.

Satellite Server is a caching engine that enables WebCenter Sites to serve pre-rendered, static versions of pages that remain unchanged since a visitor's last request (otherwise, WebCenter Sites would generate the pages dynamically upon each request). Satellite Server caching improves the performance of your WebCenter Sites system.

Figure 15–14 Production Sites Satellite User


Oracle WebCenter Sites 11gR1 | Community-Gadgets
X
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Production Sites Satellite User

Production WebCenter Sites Satellite User:
SatelliteServer

Production WebCenter Sites Satellite User's Password:
••••••••

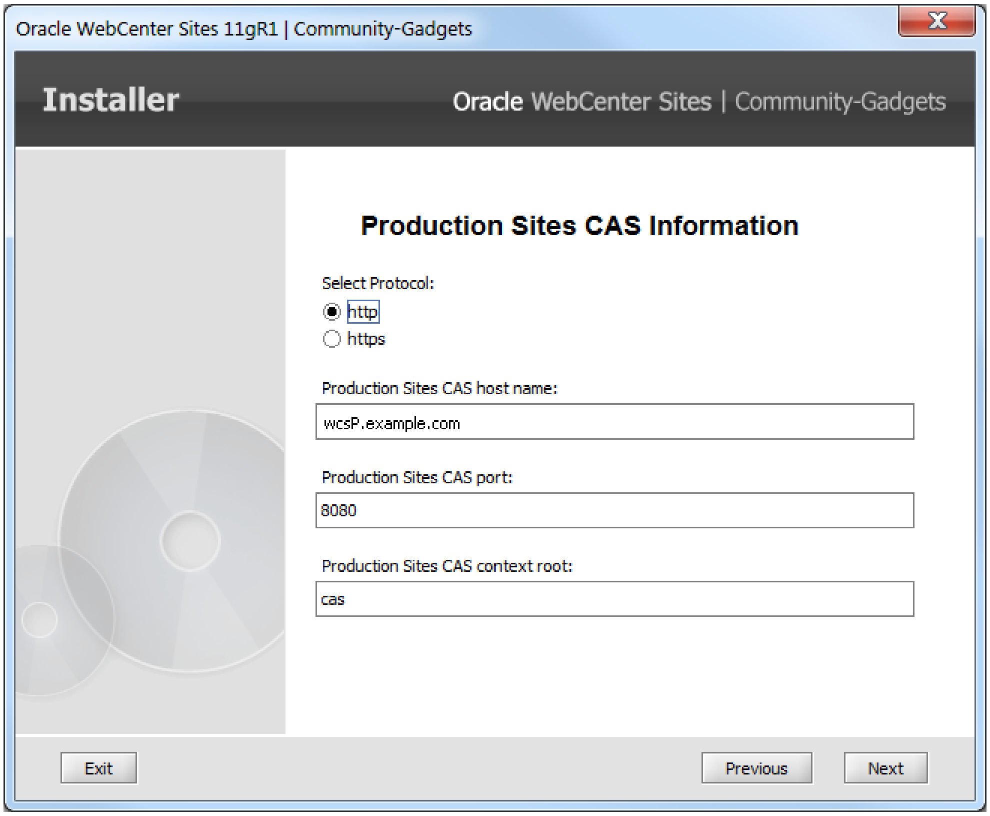
Re-enter Production WebCenter Sites Satellite User's Password:
••••••••

20. On the Production Sites CAS Information screen (Figure 15–15):

> **Note:** If WebCenter Sites on the production system is integrated with Oracle Access Manager, accept the default values, click **Next**, and continue to step 21.

- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the production Sites CAS application server.
- Enter the port number of the production Sites CAS application server.
- Enter the production Sites CAS application context root.

Figure 15–15 Production Sites CAS Information


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets

Production Sites CAS Information

Select Protocol:

● http
○ https

Production Sites CAS host name: wcsP.example.com

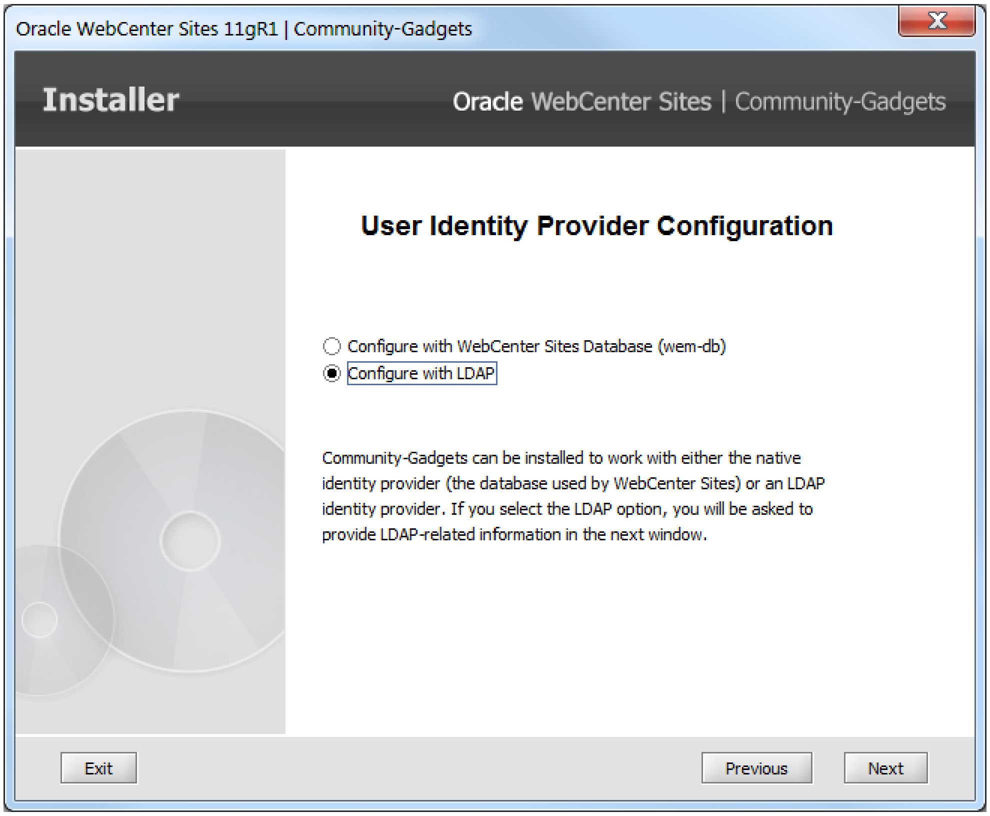
Production Sites CAS port: 8080

Production Sites CAS context root: cas

21. On the User Identity Provider Configuration screen, select one of these options (Figure 15–16):

    - **Configure with WebCenter Sites Database (wem-db)**
    - **Configure with LDAP**. If you select this option, the LDAP Server Information screen is displayed when you click **Next**.

Figure 15–16 User Identity Provider Configuration


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets

User Identity Provider Configuration

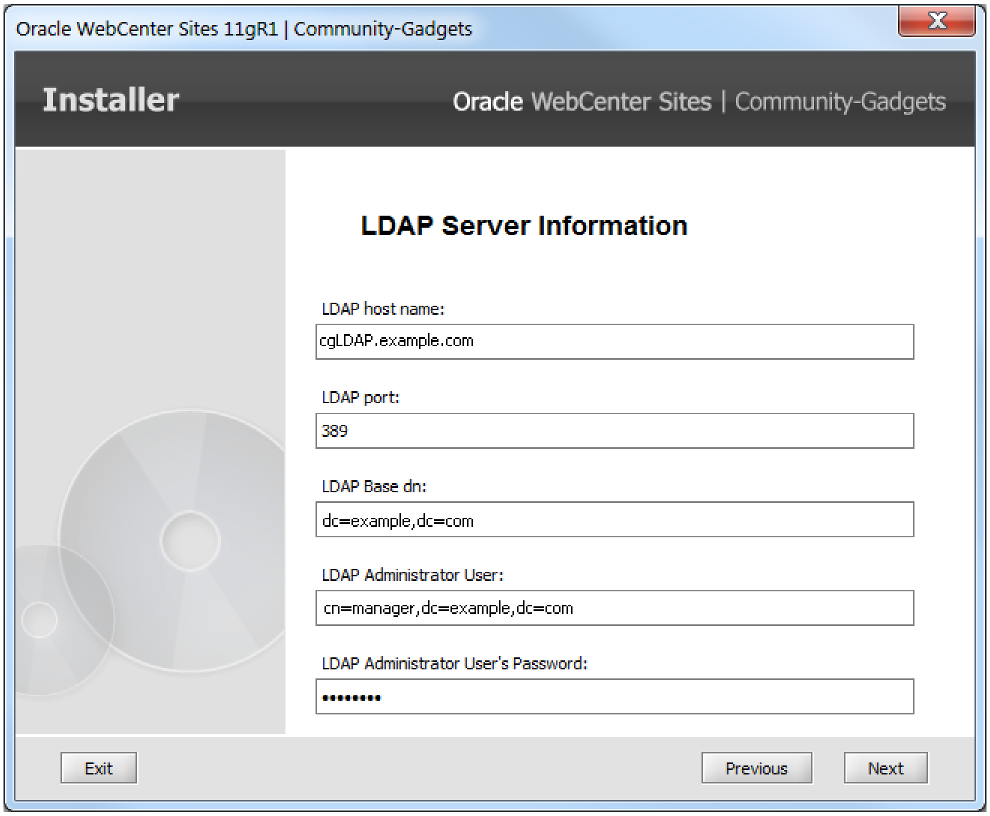
○ Configure with WebCenter Sites Database (wem-db)
● Configure with LDAP

Community-Gadgets can be installed to work with either the native identity provider (the database used by WebCenter Sites) or an LDAP identity provider. If you select the LDAP option, you will be asked to provide LDAP-related information in the next window.

22. If you selected Configure with LDAP in the previous step, provide the following values on the "LDAP Server Information" screen shown in Figure 15–17 (for examples, see Section 13.1.3, "Prerequisites for LDAP Configuration").

- LDAP server host name or IP address.
- LDAP server port.
- LDAP server Base DN.
- LDAP administrator user.
- LDAP administrator's password.

Figure 15–17 LDAP Server Information


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

LDAP Server Information

LDAP host name: cgLDAP.example.com

LDAP port: 389

LDAP Base dn: dc=example,dc=com

LDAP Administrator User: cn=manager,dc=example,dc=com

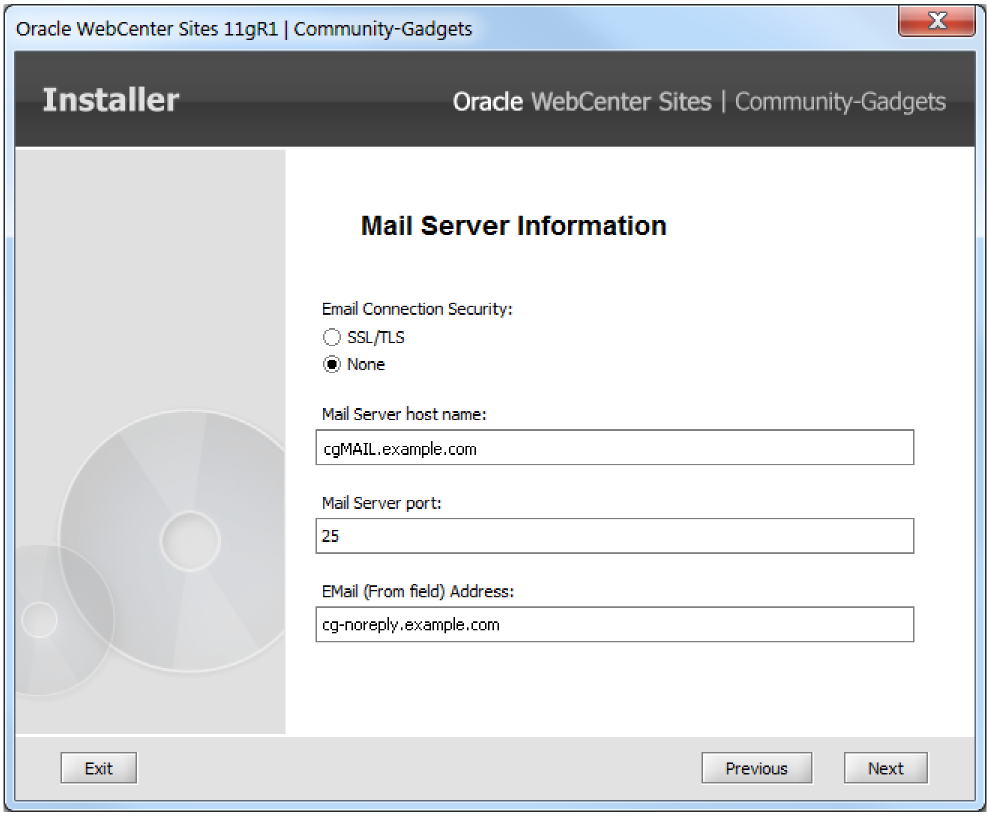
LDAP Administrator User's Password: ••••••••

23. On the Mail Server Information screen (Figure 15–18):

- Select **SSL/TLS** or **None** for email connection security.
- Enter the mail server host name or IP address.
- Enter the mail server port.
- Enter the email address to be used in the **From** field.

> **Note:** Mail server information is used by the Community-Gadgets application to communicate with visitors.

Figure 15–18 Mail Server Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets

Mail Server Information

Email Connection Security:

○ SSL/TLS

● None

Mail Server host name: cgMAIL.example.com

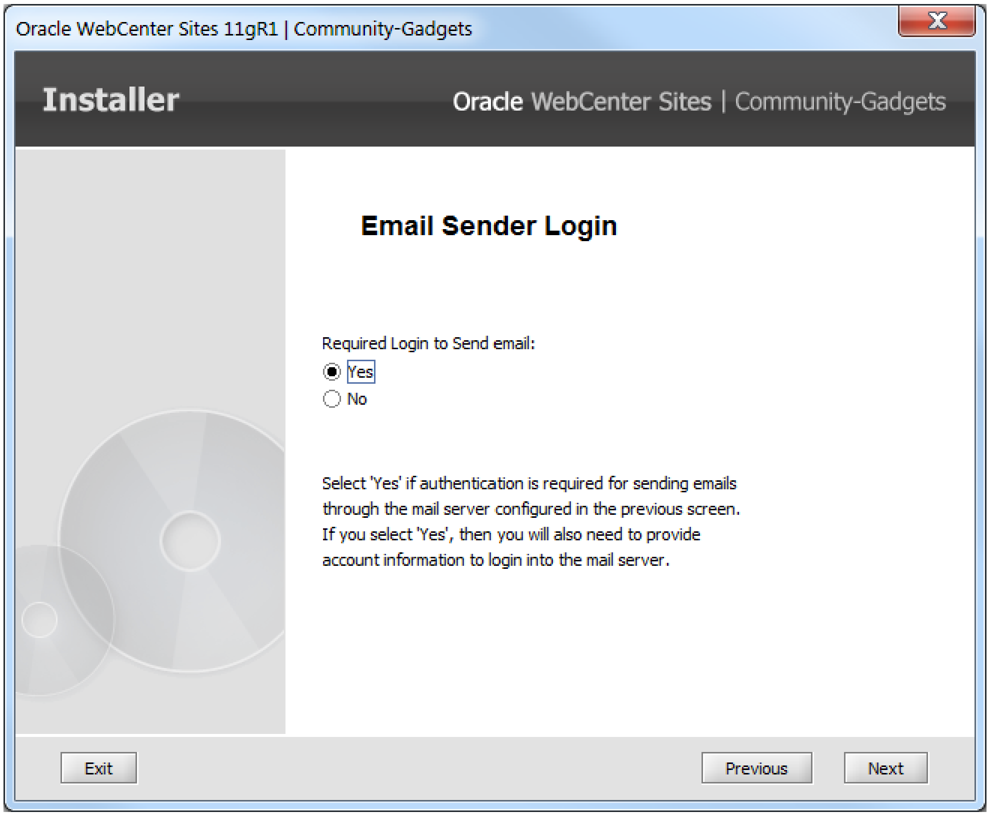
Mail Server port: 25

EMail (From field) Address: cg-noreply.example.com

24. On the Email Sender Login screen, select one of these options (Figure 15–19):

- **Yes**: Login is required to send mail. If you select this option, the Email Sender Information screen is displayed when you click **Next**.
- **No**: Login is not required to send mail.

Figure 15–19 Email Sender Login



Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Exit
Previous
Next

Email Sender Login

Required Login to Send email:

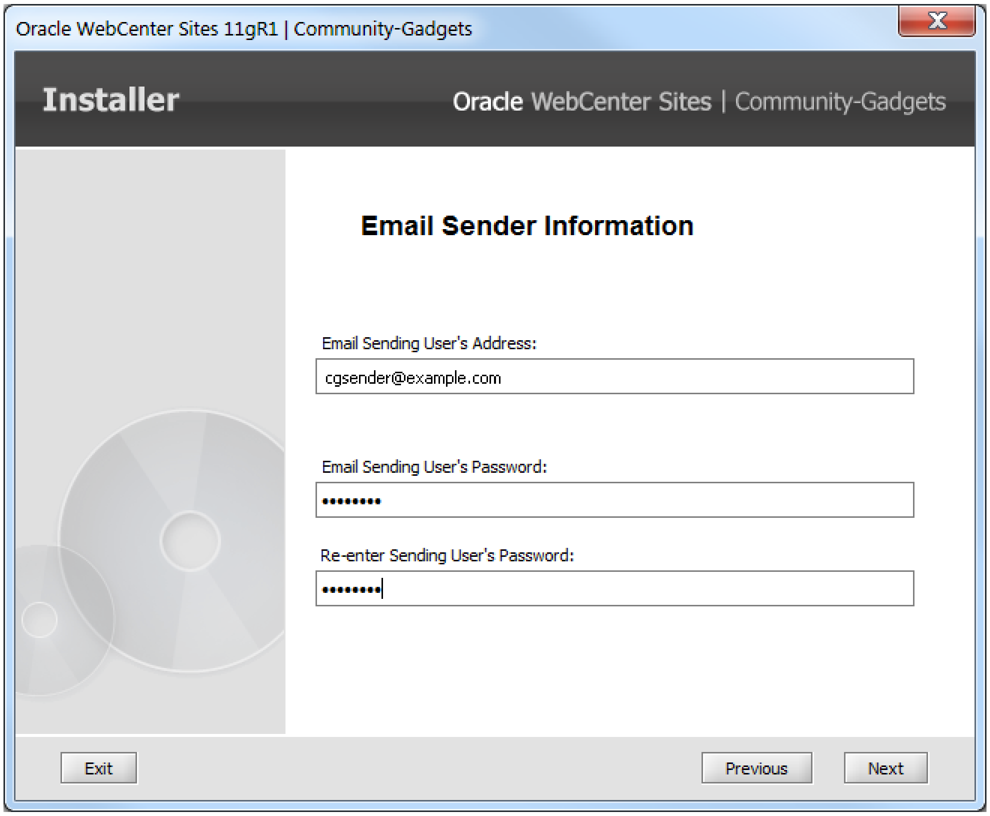
● Yes
○ No

Select 'Yes' if authentication is required for sending emails through the mail server configured in the previous screen. If you select 'Yes', then you will also need to provide account information to login into the mail server.

25. If you selected **Yes** in the previous step, provide these values on the Email Sender Information screen (Figure 15–20):

    - Email address of the sending user.
    - Password of the sending user.
    - For verification, re-enter the password of the sending user.

Figure 15–20 Email Sender Information



Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets

Email Sender Information

Email Sending User's Address: cgsender@example.com

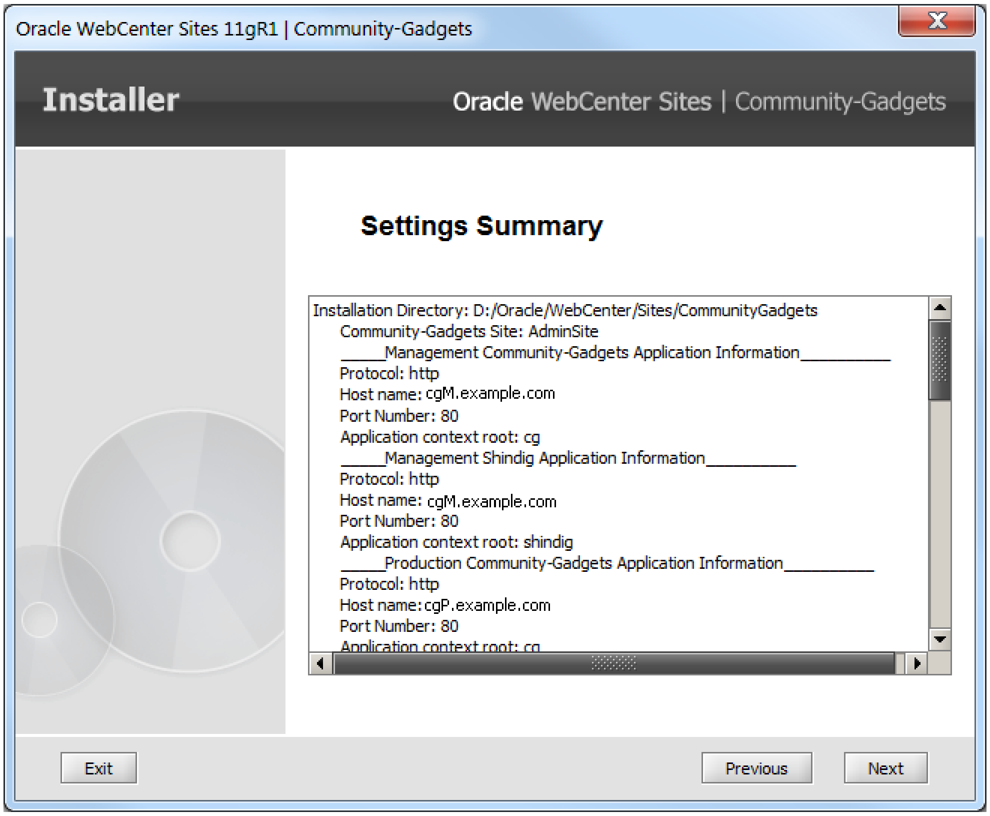
Email Sending User's Password: ••••••••

Re-enter Sending User's Password: ••••••••

26. The Settings Summary screen (Figure 15–21) summarizes the configuration choices you have made for this installation.

- Review the settings summary to make sure all options are configured correctly.
- If you find an option that needs to be modified, click **Previous** to return to the installer screen containing that option.

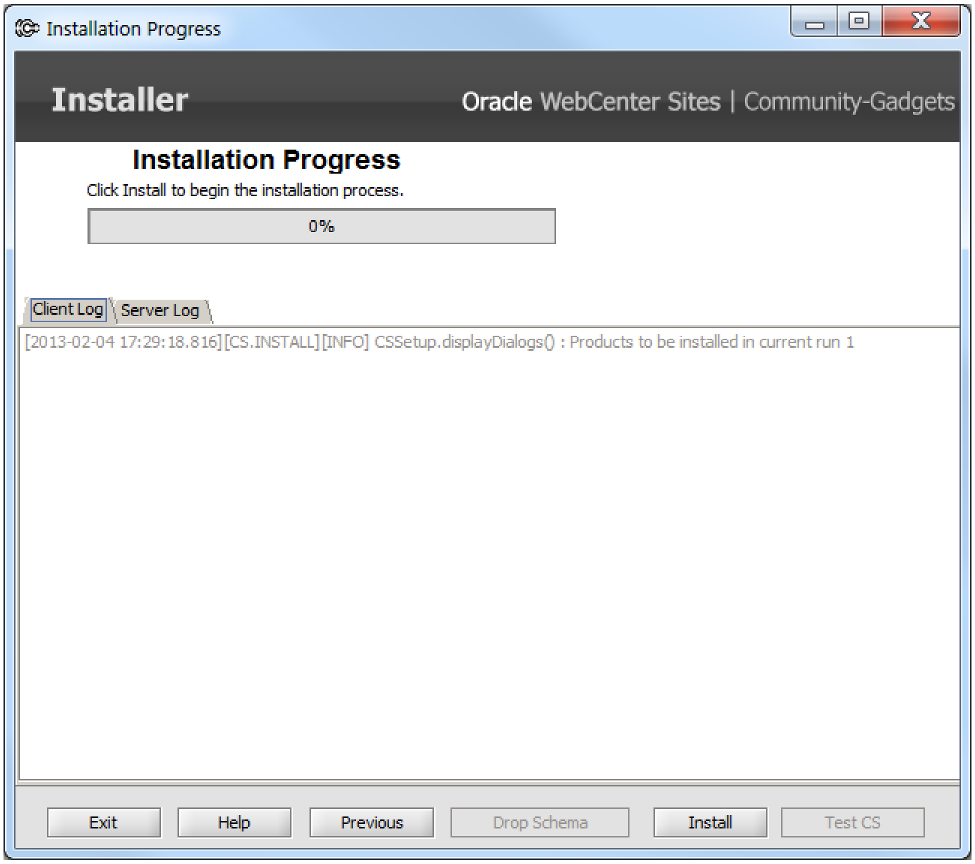
Figure 15–21 Settings Summary


Oracle WebCenter Sites 11gR1 | Community-Gadgets
Installer
Oracle WebCenter Sites | Community-Gadgets
Settings Summary
Exit
Previous
Next

```
Installation Directory: D:/Oracle/WebCenter/Sites/CommunityGadgets
    Community-Gadgets Site: AdminSite
    _____Management Community-Gadgets Application Information__________
    Protocol: http
    Host name: cgM.example.com
    Port Number: 80
    Application context root: cg
    _____Management Shindig Application Information__________
    Protocol: http
    Host name: cgM.example.com
    Port Number: 80
    Application context root: shindig
    _____Production Community-Gadgets Application Information__________
    Protocol: http
    Host name:cgP.example.com
    Port Number: 80
    Application context root: cg
```

27. On the Installation Progress screen (Figure 15–22), click **Install** to start the process of generating the deployment files. You can monitor the process from the **Client Log** tab.

Figure 15–22 Installation Progress



Installation Progress

Click Install to begin the installation process.

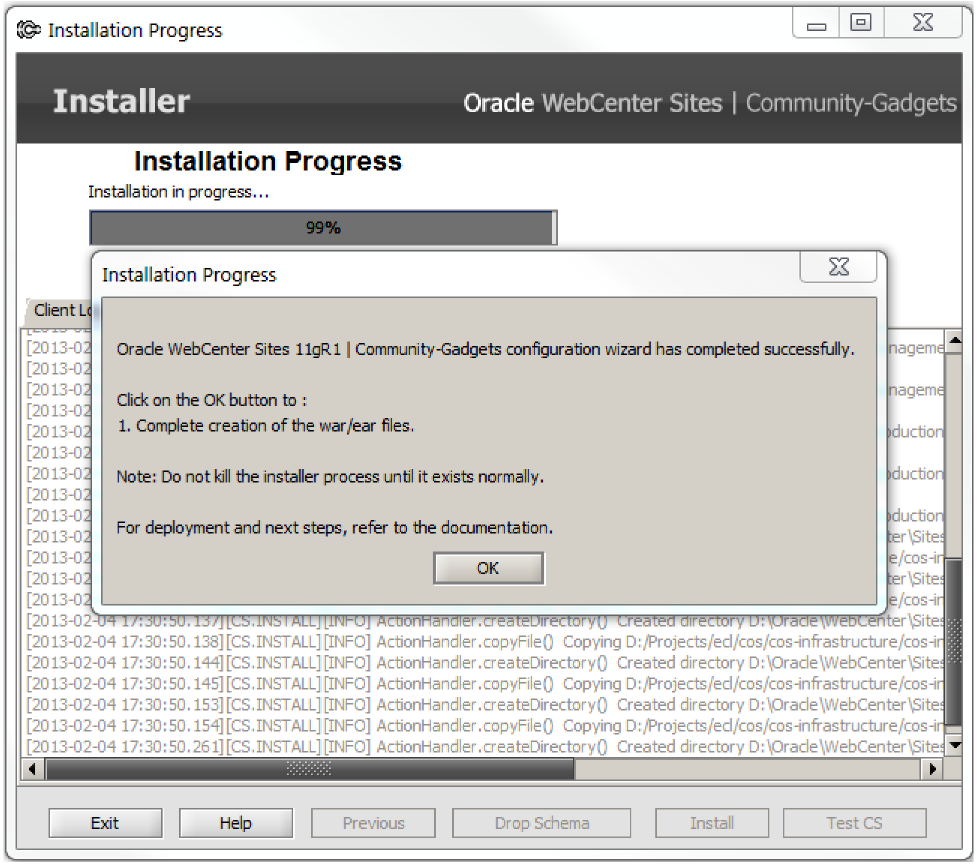
0%

Client Log | Server Log

```
[2013-02-04 17:29:18.816][CS.INSTALL][INFO] CSSetup.displayDialogs() : Products to be installed in current run 1
```

28. When the installer tasks are completed successfully, a confirmation note is displayed (Figure 15–23). Click **OK** to create the `war/ear` files for the Community-Gadgets web applications. Do not end the installer process. Allow it to exit normally.

Figure 15–23 Installation Progress



Oracle WebCenter Sites 11gR1 | Community-Gadgets configuration wizard has completed successfully.

Click on the OK button to :
1. Complete creation of the war/ear files.

Note: Do not kill the installer process until it exists normally.

For deployment and next steps, refer to the documentation.

```
[2013-02-04 17:30:50.137][CS.INSTALL][INFO] ActionHandler.createDirectory()  Created directory D:\Oracle\WebCenter\Sites
[2013-02-04 17:30:50.138][CS.INSTALL][INFO] ActionHandler.copyFile()  Copying D:/Projects/ecl/cos/cos-infrastructure/cos-in
[2013-02-04 17:30:50.144][CS.INSTALL][INFO] ActionHandler.createDirectory()  Created directory D:\Oracle\WebCenter\Sites
[2013-02-04 17:30:50.145][CS.INSTALL][INFO] ActionHandler.copyFile()  Copying D:/Projects/ecl/cos/cos-infrastructure/cos-in
[2013-02-04 17:30:50.153][CS.INSTALL][INFO] ActionHandler.createDirectory()  Created directory D:\Oracle\WebCenter\Sites
[2013-02-04 17:30:50.154][CS.INSTALL][INFO] ActionHandler.copyFile()  Copying D:/Projects/ecl/cos/cos-infrastructure/cos-in
[2013-02-04 17:30:50.261][CS.INSTALL][INFO] ActionHandler.createDirectory()  Created directory D:\Oracle\WebCenter\Sites
```

29. Continue to Section 15.3, "Verifying the Community-Gadgets Installation Directory Structure."

15.2 Generating the Community-Gadgets Deployment Files Silently

Follow these steps to generate the Community-Gadgets deployment files silently:

1. Extract the Community-Gadgets installer on the server where you will install the Community-Gadgets application.
2. The installer folder contains the sample `omii_sample.ini` file, which you can use for your silent installation.
   a. Edit the file by verifying the default values and entering additional values as required. Refer to the comments in the file for guidance.
   b. Save the file and copy it to a folder outside `<cg_install_dir>`.
3. Edit the `install.ini` file (in the extracted Community-Gadgets installer) as follows:
   - Set `loadfile=<path_to_the_copied_omii.ini_file>` (to point to the `omii.ini` file in step 2b).
   - Set `nodisplay=true`

> **Note:** Path to the file should include forward slashes ("/") as a path separator, even on the Windows platform. For example: `d:/CG/omii.ini`

4. Create an installation directory in which the Community-Gadgets installer will create the necessary folders and files.

   For example:

   ```
   /u01/software/Apps/CG
   ```

5. Set `JAVA_HOME` to the path of JDK 1.6 or a later version.

   For example:

   ```
   export JAVA_HOME=/opt/jdk1.6.0_43
   ```

6. Run the Community-Gadgets silent installer:

   a. Run the `cgInstall.sh` or `cgInstall.bat` script with the `-silent` parameter.

   b. Wait until the "`CG web-application files have been generated successfully`" message appears and the process is complete.

7. Continue to Section 15.3, "Verifying the Community-Gadgets Installation Directory Structure."

15.3 Verifying the Community-Gadgets Installation Directory Structure

When the installer completes its process, it creates the Community-Gadgets installation directory containing the generated deployment files.

1. Verify your Community-Gadgets installation directory and the locations of the generated deployment files. Refer to the directory structure shown in this step. The main subdirectories are:

   - `cg_licenses`: This subdirectory contains the necessary license files.
   - `deploy`: This subdirectory contains the `war` and `ear` files generated for the management and production sides of the Community-Gadgets application. This directory also contains standalone configuration files that should be added to the application server classpath. You are required to deploy the `war /ear` files manually.
   - `logs`: This subdirectory stores logs that will be created by the Community-Gadgets application during use.
   - `ominstallinfo`: This subdirectory contains information about the Community-Gadgets installation. The installation log and installation settings file are stored in this subdirectory.

   ```
   <cg_install_dir>
   |--cg_licenses
   | |--cas_thirdparty_licenses
   | |--shindig_thirdparty_licenses
   |--deploy
   | |--management
   | | |--CommunityGadgets.ear
   | | |--cg.war
   | | |--Shindig.ear
   | | |--shindig.war
   ```

```
| | |--management_node1
| | | |--cas-cache.xml
| | | |--cos-cache.xml
| | | |--log4j-cos.properties
| | | |--log4j-shindig.properties
| | | |--setup_auth.properties
| | | |--setup_cos.properties
| | | |--setup_cs.properties
| | | |--setup_proxy.properties
| | | |--setup_shindig.properties
| | | |--wsdk_facilities.properties
| | | |--ESAPI.properties
| | | |--validation.properties
| | | |--wem_sso_config.xml
| | | |--oam_wem_sso_config_sample.xml
| | | |--gas_features.txt
| |--production
| | |--cas.ear
| | |--cas.war
| | |--CommunityGadgets.ear
| | |--cg.war
| | |--Shindig.ear
| | |--shindig.war
| |--production_node1
| | | |--cas.properties
| | | |--customBeans.xml
| | | |--host.properties
| | | |--jbossTicketCacheReplicationConfig.xml
| | | |--cas-cache.xml
| | | |--cos-cache.xml
| | | |--log4j-cos.properties
| | | |--log4j.properties
| | | |--log4j-shindig.properties
| | | |--setup_auth.properties
| | | |--setup_cos.properties
| | | |--setup_cs.properties
| | | |--setup_proxy.properties
| | | |--setup_shindig.properties
| | | |--wsdk_facilities.properties
| | | |--ESAPI.properties
| | | |--validation.properties
| | | |--wem_sso_config.xml
| | | |--oam_wem_sso_config_sample.xml
| | | |--gas_features.txt
|--logs
|--ominstallinfo
| |--install_log.log
| |--management_cgwar.log
| |--management_shindigwar.log
| |--omii.ini
| |--omproduct.ini
| |--production_caswar.log
| |--production_cgwar.log
| |--production_shindigwar.log
```

2. Continue to Chapter 16, "Completing Community-Gadgets Pre-Deployment Steps."

16

Completing Community-Gadgets Pre-Deployment Steps

This chapter describes the steps you must complete before you can deploy Community-Gadgets. It is assumed that you have successfully generated the Community-Gadgets deployment file.

This chapter contains the following sections:

16.1 Configuring inCache Invalidation

1. Run Sites Explorer and connect to the production WebCenter Sites server using the `ContentServer` user. (For instructions on logging in to Sites Explorer, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.)
2. Open the `SystemSatellite` table in Sites Explorer:
   a. Copy the first row of the table.
   b. Paste the row into the table, once for each management node of the Community-Gadgets application and once for each production node of the Community-Gadgets application.
   c. In the new rows, change the values of the following fields to match the values for your installation:
      - **description**: `<node name>`
      - **protocol**: `<http or https>`
      - **host:** `<host name or IP address of the management or production Community-Gadgets application server>`
      - **port:** `<port number of the management or production Community-Gadgets application server>`

- **flushservletpath:** /cg/FlushServer

> **Note:** If the standard Community-Gadgets application context root is not used, use:
>
> ```
> /<new context root>/FlushServer
> ```

- **username**: <of the production Satellite Server user> (the default is SatelliteServer)
- **password**: <password of the above user>

3. Continue to Section 16.2, "Copying Installer-Generated Configuration Files."

16.2 Copying Installer-Generated Configuration Files

The Community-Gadgets installer created property files and other configuration files that will be read by the application server to start the deployed Community-Gadgets application correctly.

1. For easier management and scalability, we recommend copying the files such that each instance of the Community-Gadgets application has its own set of files:

   a. In <cg_install_dir>, create a directory called <cg-standalone-configs>.

   b. Within this newly created directory, create a subdirectory for each management node and each production node. The subdirectories will hold the configuration files for those nodes. These directories can be either on a shared file system or on an individual server, depending on you architecture.

   c. For each subdirectory that you created in the step above, add its path to your application server's classpath.

   For example, in Linux:

   ```
   mkdir <cg_install_dir>/<cg-standalone-configs>
   cd <cg_install_dir>/<cg-standalone-configs>
   mkdir <cgA_M1> <cgA_M2> <cgB_P1> <cgB_P2> ...
   ```

2. Copy the installer-generated configuration and property files to the node subdirectories as follows:

   For each Management Community-Gadgets application node:

   a. Copy the configuration files from <cg_install_dir>/deploy/management/management_node1 to the subdirectories that you created in step 1.b for the management Community-Gadgets application.

   For example, in Linux:

   ```
   cd <cg_install_dir>/deploy/management/management_node1
   cp -p ./* <cg_install_dir>/<cg-standalone-configs>/<cgA_M1>
   ```

   b. If needed, change the log file names by modifying log4j-cos.properties and log4j-shindig.properties for each management Community-Gadgets application on a particular node.

   For each Production Community-Gadgets application node:

   1. Copy the configuration files from <cg_install_dir>/deploy/production/production_node1 to the subdirectories that you

created in step 1.b for the production Community-Gadgets application.

For example, in Linux:

```
cd <cg_install_dir>/deploy/production/production_node1
cp -p ./* <cg_install_dir>/<cg-standalone-configs>/<cgB_P1>
```

2. Edit `jbossTicketCacheReplicationConfig.xml` by going to the UDP tag and doing the following:

   – For a non-clustered application, set the value of `mcast_port` to a port number that is unique relative to the other CAS applications on the subnet.

   – For a clustered application, set the value of `ip_ttl` to `8` and change the `bind_addr` to the host IP address of the Visitors CAS cluster member. Ensure that all cluster members are listening on the same mcast address and port.

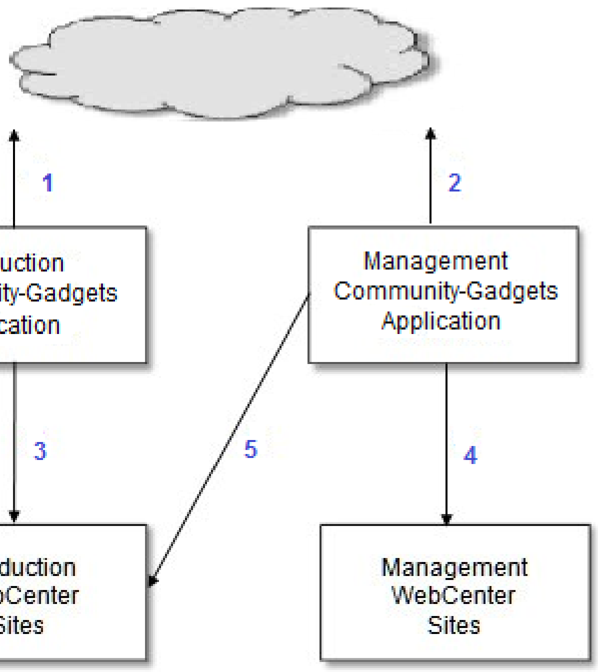
3. If needed, change the log file names by modifying `log4j-cos.properties`, `log4j-shindig.properties`, and `log4j.properties` for each production Community-Gadgets application on a particular node.

3. Complete the rest of the steps in this chapter as necessary for your system. When you have finished, continue to Chapter 17, "Deploying Community-Gadgets."

16.3 Proxy Connection Settings

The Community-Gadgets application supports up to five proxy connection settings (disabled by default) as shown in Figure 16–1.

Figure 16–1 Proxy Connection Settings Supported by Community-Gadgets


1
2
Production Community-Gadgets Application
Management Community-Gadgets Application
3
5
4
Production WebCenter Sites
Management WebCenter Sites

- Proxy connection 1 is an Internet connection proxy that supports communication from the production Community-Gadgets application to the Internet. This proxy connection is used for external authentication via third-party services such as Facebook, Twitter, Google, and Janrain.

- Proxy connection 2 is an Internet connection proxy that supports communication from the management Community-Gadgets application to the Internet. This proxy connection is used for access to external gadgets descriptors during gadgets registration.
- Proxy connection 3 supports communication from the production Community-Gadgets application to the production WebCenter Sites application.
- Proxy connection 4 supports communication from the management Community-Gadgets application to the management WebCenter Sites application.
- Proxy connection 5 supports communication from the management Community-Gadgets application to the production WebCenter Sites application.

If you wish to configure proxy connections, do the following:

1. Proxy connections are configured in the `setup_proxy.properties` file located in the `deploy/production/production_node1` and `deploy/management/management_node1` folders, or inside the folders you created in Section 16.2, "Copying Installer-Generated Configuration Files." The `setup_proxy.properties` file contains five sections with parameters for each proxy connection.

   - If proxies are not used, all proxy sections by default have the `<proxy>.proxy.attrs.enabled` parameter set to `false`.
   - If a proxy is used, go to the relevant section and set the following parameters:
     - Set the `<proxy>.proxy.attrs.enabled` parameter to `true`.
     - Set the `<proxy>.proxy.attrs.host` parameter to the host name or IP address of the proxy.
     - Set the `<proxy>.proxy.attrs.port` parameter to the port of the proxy `cos-production_www.proxy.attrs.auth.user`
   - If a proxy requires authentication, set the `<proxy>.proxy.attrs.auth.required` parameter to `true` and specify the username and password in the `<proxy>.proxy.attrs.auth.user` and `<proxy>.proxy.attrs.auth.password` parameters.

   > **Note:** In the Internet proxies sections, it can be useful to add the host names of your management and production WebCenter Sites instances to the list of hosts for which a proxy should not be used in production and management.

   - For reference, a sample configuration section is given below:

     Sample configuration for a production Internet connection proxy (first section in the `setup_proxy.properties` file):

     ```
     # Use proxy server
     # Possible values: "true" or "false"
     cos-production_www.proxy.attrs.enabled=true
     # Proxy type
     # Possible values: "http" or "socks"
     cos-production_www.proxy.attrs.type=http
     # Proxy server host
     cos-production_www.proxy.attrs.host=www-proxy.example.com
     # Proxy server port.
     ```

```
cos-production_www.proxy.attrs.port=80
# Is authentication required for proxy server configured above.
cos-production_www.proxy.attrs.auth.required=true
# User proxy account to log in to the proxy server
cos-production_www.proxy.attrs.auth.user=user
# Password of the above user.
cos-production_www.proxy.attrs.auth.password=proxypassword
# List of hosts for which proxy should not be used
cos-production_www.proxy.attrs.non_proxy_hosts=
```

2. Complete the rest of the steps in this chapter as necessary for your system. When you have finished, continue to Chapter 17, "Deploying Community-Gadgets.".

16.4 Adding Community-Gadgets URLs to the WebCenter Sites customBeans.xml file

1. In your installation, navigate to `<sites_install_dir>/bin/customBeans.xml`.
2. Open the `customBeans.xml` file.
3. In the `RequestParamValidator` bean section, add the Community-Gadgets URLs inside `validUrls` list as follows:

   - On the WebCenter Sites management system, enter the following for each management Community-Gadgets instance:

     ```
     <value>{protocol}://{cg_mgmt_node_host}:{cg_mgmt_node_port}/{cg_
     context}/*</value>
     ```

   - On the WebCenter Sites production system, enter the following for each production Community-Gadgets instance:

     ```
     <value>{protocol}://{cg_prod_node_host}:{cg_prod_node_port}/{cg_
     context}/*</value>
     ```

   - In case of clustered installation, additional value is required for Community-Gadgets Load Balancer.

   Example 16–1 shows a sample `customBeans.xml` file.

Example 16–1 Sample customBeans.xml File

```
<bean id="RequestParamValidator"
class="com.fatwire.cas.web.RequestParamValidator">
  <property name="validUrls">
    <list>
      <!-- Sites URL -->
      <value>http://192.0.2.1:8080/cs/*</value>
      <!-- Community-Gadgets  node 1 -->
      <value>http://192.0.2.2:8080/cg/*</value>
      <!-- Community-Gadgets  node 2 -->
      <value>http://192.0.2.3:8080/cg/*</value>
      <!-- Community-Gadgets  Load Balancer 1 -->
      <value>http://192.0.2.4:80/cg/*</value>
    </list>
  </property>
</bean>
```

16.5 Enabling Shindig Features

Gadgets that are either developed for iGoogle or use iGoogle features may require you to enable extra Shindig features in order for those gadgets to render properly. If you enable these Shindig features, the Community-Gadgets application downloads the third-party libraries necessary for rendering the gadgets that require them from the Internet.

Note: To enable downloading of third-party libraries, you must have access to the Internet. If you use a proxy for Internet access, configure the proper connections settings for both management and production Internet access in the `setup_proxy.properties` files. For more information, see Section 16.3, "Proxy Connection Settings."

1. If you wish to enable additional Shindig features for the Community-Gadgets application, uncomment the following lines in the `gas_features.txt` file located in the `<cg_install_dir>/deploy/management/management_node1` and `<cg_install_dir>/deploy/production/production_node1` folders (or in the folders you created in Section 16.2, "Copying Installer-Generated Configuration Files"):

   ```
   # features/analytics/feature.xml
   # features/com.google.gadgets.analytics/feature.xml
   # features-extras/org.jquery.core-1.4.2/feature.xml
   ```

2. Complete the rest of the steps in this chapter as necessary for your system. When you have finished, continue to Chapter 17, "Deploying Community-Gadgets."

16.6 If Your Management and/or Production WebCenter Sites Is Integrated with OAM

1. If your management and/or production instance of WebCenter Sites is integrated with OAM, configure the Community-Gadgets instance to work with the same OAM. For instructions, see *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.
2. Continue to Chapter 17, "Deploying Community-Gadgets."

17
Deploying Community-Gadgets

This chapter describes the steps for deploying Community-Gadgets on supported application servers.

This chapter contains the following sections:

- Section 17.1, "Overview"
- Section 17.2, "Deploying a Non-Clustered Community-Gadgets Application"
- Section 17.3, "Deploying a Clustered Community-Gadgets Application"

17.1 Overview

When deploying Community-Gadgets, you will do the following:

- On the management system, you will deploy the management Community-Gadgets web application and the management Shindig web application (for gadgets rendering).
- On the production system, you will deploy the production Community-Gadgets web application, the production Shindig web application (for gadgets rendering), and the Visitors CAS web application (for authentication).

Instructions are available in this chapter.

> **Note:** Before proceeding with this chapter, ensure you have completed requirements procedures in the previous chapters of Part IV, "Installing Oracle WebCenter Sites: Community-Gadgets."

17.2 Deploying a Non-Clustered Community-Gadgets Application

This section describes the steps for deploying a non-clustered Community-Gadgets application to supported application servers.

This section covers the following deployment configurations:

- Section 17.2.1, "Deploying on Apache Tomcat"
- Section 17.2.2, "Deploying on Oracle WebLogic Application Server"
- Section 17.2.3, "Deploying on IBM WebSphere Application Server"

17.2.1 Deploying on Apache Tomcat

In this section, you will deploy a non-clustered Community-Gadgets application on Tomcat.

Note: If you need detailed steps on how to deploy an application on Apache Tomcat, refer to the vendor's documentation.

1. Before deploying the Community-Gadgets application, ensure that each application server's classpath points to the respective Community-Gadgets configuration directories (created in Section 16.2, "Copying Installer-Generated Configuration Files").

   For reference, this guide uses the configurations described in Table 17–1.

Table 17–1 Classpath Configuration for Tomcat Servers (Non-clustered)

Tomcat Server	Server Classpath
`<cgA_M1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgA_M1>:${CLASSPATH}"`
`<cgB_P1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgB_P1>:${CLASSPATH}"`

2. The Community-Gadgets installer creates `war` and `ear` files in the `<cg_install_dir>/deploy/management` directory and the `<cg_install_dir>/deploy/production` directory. Deploy the Community-Gadgets application either through the Tomcat administration console, or by copying the exploded `war` files to the `webapps` directory. Locations of the `war` files are shown in Table 17–2.

Note: In the `catalina.sh` (or `catalina.bat`) file, ensure that the line `"CLASSPATH="` is commented out.

- **On UNIX:** `#CLASSPATH=`
- **On Windows:** `REM CLASSPATH=`

Table 17–2 war File Locations for Tomcat (Non-clustered)

Application Server Instances	Location of WAR Files	Sample Deployment Locations

`<cgA_M1>`	`<cg_install_dir>/deploy/management/cg.war`
	`<cg_install_dir>/deploy/management/shindig.war`
		`<cgA_M1>/webapps/cg`
		`<cgA_M1>/webapps/shindig`

`<cgB_P1>`	`<cg_install_dir>/deploy/production/cg.war`
	`<cg_install_dir>/deploy/production/shindig.war`
	`<cg_install_dir>/deploy/production/cas.war`
		`<cgB_P1>/webapps/cg`
		`<cgB_P1>/webapps/shindig`
		`<cgB_P1>/webapps/cas`

Note: If the CG context root is changed at the time of installation, then in the tomcat `webapps` the CG deploy folder name should be the same as the CG Context root.

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 13, "Prerequisites for Installing Community-Gadgets."

4. Start the production and management deployment servers.

   For example, in Linux:

   On `<ServerB>`:

   ```
   cd <cgB_P1>/bin
   ./catalina.sh run
   ```

   On `<ServerA>`:

   ```
   cd <cgA_M1>/bin
   ./catalina.sh run
   ```

5. Continue to Chapter 18, "Verifying and Implementing Community-Gadgets."

17.2.2 Deploying on Oracle WebLogic Application Server

In this section, you will deploy a non-clustered Community-Gadgets application on Oracle WebLogic Application Server.

> **Note:** Consider the following:
>
> - If you need detailed steps on deploying web applications on Oracle WebLogic Application Server, refer to the Oracle WebLogic Application Server documentation.
> - If the Community-Gadgets application will be deployed on WebLogic application server, add the following JVM parameter to `JAVA_OPTIONS` of WebLogic application server: `-DUseSunHttpHandler=true`

1. Before deploying the Community-Gadgets application, ensure that each application server's classpath points to the respective Community-Gadgets configuration directories (created in Section 16.2, "Copying Installer-Generated Configuration Files.")

   For reference, this guide uses the configurations described in Table 17–3.

Table 17–3 Classpath Configuration for Managed Servers (WebLogic, Non-clustered)

Managed Servers	Server Classpath
`<cgA_M1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgA_M1>:${CLASSPATH}"`
`<cgB_P1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgB_P1>:${CLASSPATH}"`

2. The Community-Gadgets installer creates `war` and `ear` files in the `<cg_install_dir>/deploy/management` and `<cg_install_dir>/deploy/production` directories. Extract the contents of the `war` files and deploy them via the command line or the Administration Console. Locations of the `war` files are shown in Table 17–4.

Table 17–4 war File Locations for WebLogic (Non-clustered)

Managed Server	Location of war Files	Sample Deployment Location

`<cgA_M1>`	`<cg_install_dir>/deploy/management/cg.war`
	`<cg_install_dir>/deploy/management/shindig.war`
		`<WL_HOME>/user_projects/domains/<cgMgmt>/applications/cg`
		`<WL_HOME>/user_projects/domains/<cgMgmt>/applications/shindig`

`<cgB_P1>`	`<cg_install_dir>/deploy/production/cg.war`
	`<cg_install_dir>/deploy/management/shindig.war`
	`<cg_install_dir>/deploy/production/cas.war`
		`<WL_HOME>/user_projects/domains/<cgProd>/applications/cg`
		`<WL_HOME>/user_projects/domains/<cgProd>/applications/shindig`
		`<WL_HOME>/user_projects/domains/<cgProd>/applications/cas`

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 13, "Prerequisites for Installing Community-Gadgets."

4. Copy the following files from the Community-Gadgets application to its application server's JRE: `jaxb-api-2.2.2.jar` and `jaxb-impl-2.2.3-1.jar`. To get these files, unzip `cos.war`. These JARs are located in the `WEB-INF/lib` directory.

   The files must be placed into the `<WebLogic_JDK>/jre/lib/endorsed` folder (if the `endorsed` folder does not exist, create it).

5. Start the production and management deployment servers.

   For example, in Linux:

```
cd <WL_HOME>/user_projects/domains/<cgProd>/bin
./startManagedServer <cgB_P1> http://<ServerB>:7001

cd <WL_HOME>/user_projects/domains/<cgMgmt>/bin
./startManagedServer <cgA_M1> http://<ServerA>:7001
```

6. Continue to Chapter 18, "Verifying and Implementing Community-Gadgets."

17.2.3 Deploying on IBM WebSphere Application Server

In this section, you will deploy a non-clustered Community-Gadgets application on IBM WebSphere Application Server.

> **Note:** If you need detailed steps on deploying web applications on IBM WebSphere Application Server, refer to the vendor's documentation.

1. Before deploying the Community-Gadgets application, ensure that each application server's classpath points to the respective Community-Gadgets configuration directories (created in Section 16.2, "Copying Installer-Generated Configuration Files."). To do so:

  a. Log in to the deployment manager (for example, access `http://<ServerA>:9060/ibm/console`).

  b. For each server, select **Java and Process Management**, then select **Process Definition**, then select **Java Virtual Machine** and edit the `CLASSPATH` field.

For reference, this guide uses the configurations described in Table 17–5.

Table 17–5 Classpath Configuration for WebSphere (Non-clustered)

Managed Server	Server Classpath
`<cgA_M1>`	`/<cg_install_dir>/<cg-standalone-configs>/<cgA_M1>`
`<cgB_P1>`	`/<cg_install_dir>/<cg-standalone-configs>/<cgB_P1>`

2. The Community-Gadgets installer creates `war` and `ear` files in the `<cg_install_dir>/deploy/management` directory and the `<cg_install_dir>/deploy/production` directory. Locations of the `ear` files are shown in Table 17–6.

Table 17–6 EAR File Locations for WebSphere (Non-clustered)

Managed Server	Location of EAR Files

`<cgA_M1>`	`<cg_install_dir>/deploy/management/CommunityGadgets.ear`
	`<cg_install_dir>/deploy/management/Shindig.ear`

`<cgB_P1>`	`<cg_install_dir>/deploy/production/CommunityGadgets.ear`
	`<cg_install_dir>/deploy/production/Shindig.ear`
	`<cg_install_dir>/deploy/production/cas.ear`

3. Complete the following steps in the WebSphere Admin Console for Community-Gadgets applications, Shindig applications, and the Visitors CAS application:

   a. Select **Applications**, then select **Application Types**, then select **WebSphere enterprise applications**, then select ***application_name***.

   b. Under Web Module Properties, select **Session management**.

   c. Under General Properties, select **Enable Cookies** and change the value for Cookie path from `/` to `/<application-context-root>` (the default value for the Community-Gadgets application is `/cg`, for Shindig it is `/shindig`, and for the Visitors CAS application it is`/cas`).

   d. Click **Apply and Save**.

   e. Under General Properties, select **Override session management**.

   f. Click **Apply and Save**.

4. Copy the following files from the Community-Gadgets application to its application server's JRE: `axb-api-2.2.2.jar`, `jaxb-impl-2.2.3-1.jar` and `asm-3.1.jar`. The files must be placed into the `<WebSphere_JDK>/jre/lib/endorsed` folder (if the `endorsed` folder does not exist, create it).

5. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 13, "Prerequisites for Installing Community-Gadgets."

6. Start the Deployment Manager profile and nodes.

For example, in Linux:

On <ServerA>:

```
<WS_HOME>/bin
./startManager.sh -profileName <Dmgr01>
./startNode.sh -profileName <AppSrv01>
```

On <ServerB>:

```
./startNode.sh -profileName <AppSrv02>
```

7. Start the production and management servers.

   For example, in Linux:

   On <ServerB>:

   ```
   ./startServer.sh cgB_P1 -profileName <AppSrv02>
   ```

   On <ServerA>:

   ```
   ./startServer.sh cgA_M1 -profileName <AppSrv01>
   ```

8. Continue to Chapter 18, "Verifying and Implementing Community-Gadgets."

17.3 Deploying a Clustered Community-Gadgets Application

This section describes the steps for deploying a clustered Community-Gadgets application to supported application servers.

This section covers the following deployment configurations:

- Section 17.3.1, "Deploying on Apache Tomcat (Clustered)"
- Section 17.3.2, "Deploying on Oracle WebLogic Application Server (Clustered)"
- Section 17.3.3, "Deploying on IBM WebSphere Application Server (Clustered)"

17.3.1 Deploying on Apache Tomcat (Clustered)

In this section, you will deploy a clustered Community-Gadgets application on Tomcat.

> **Note:** If you need detailed steps on how to deploy an application on Apache Tomcat, refer to the vendor's documentation.

1. Before deploying the Community-Gadgets application, ensure that each application server's classpath points to the respective Community-Gadgets configuration directories (created in Section 16.2, "Copying Installer-Generated Configuration Files").

   For reference, this guide uses the configurations described in Table 17–7.

Table 17–7 Classpath Configuration for Tomcat Servers (Clustered)

Managed Server	Server Classpath
<cgA_M1>	CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgA_M1>:${CLASSPATH}"

Table 17–7 (Cont.) Classpath Configuration for Tomcat Servers (Clustered)

Managed Server	Server Classpath
`<cgB_P1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgB_P1>:${CLASSPATH}"`
`<cgA_M2>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgA_M2>:${CLASSPATH}"`
`<cgB_P2>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgB_P2>:${CLASSPATH}"`

2. The Community-Gadgets installer creates `war` and `ear` files in the `<cg_install_dir>/deploy/management` and `<cg_install_dir>/deploy/production` directories. Deploy the Community-Gadgets application either through the Tomcat administration console, or by copying the exploded `war` files to the `webapps` directory. Locations of the `war` files are shown in Table 17–8.

Table 17–8 WAR File Locations for Tomcat (Clustered)

Application Server Instances	Location of WAR Files	Sample Deployment Location

`<cgA_M1>`	`<cg_install_dir>/deploy/management/cg.war`
	`<cg_install_dir>/deploy/management/shindig.war`
		`<cgA_M1>/webapps/cg`
		`<cgA_M1>/webapps/shindig`

`<cgB_P1>`	`<cg_install_dir>/deploy/production/cg.war`
	`<cg_install_dir>/deploy/production/shindig.war`
	`<cg_install_dir>/deploy/production/cas.war`
		`<cgB_P1>/webapps/cg`
		`<cgB_P1>/webapps/shindig`
		`<cgB_P1>/webapps/cas`

`<cgA_M2>`	`<cg_install_dir>/deploy/management/cg.war`
	`<cg_install_dir>/deploy/management/shindig.war`
		`<cgA_M2>/webapps/cg`
		`<cgA_M2>/webapps/shindig`

`<cgB_P2>`	`<cg_install_dir>/deploy/production/cg.war`
	`<cg_install_dir>/deploy/production/shindig.war`
	`<cg_install_dir>/deploy/production/cas.war`
		`<cgB_P2>/webapps/cg`
		`<cgB_P2>/webapps/shindig`
		`<cgB_P2>/webapps/cas`

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 13, "Prerequisites for Installing Community-Gadgets."

4. Start the production and management deployment servers.

   For example, in Linux:

   On `<ServerB>`:

```
cd <cgB_P1>/bin
./catalina.sh run
cd <cgB_P2>/bin
./catalina.sh run
```

On <ServerA>:

```
cd <cgA_M1>/bin
./catalina.sh run
cd <cgA_M2>/bin
./catalina.sh run
```

5. Continue to Chapter 18, "Verifying and Implementing Community-Gadgets."

17.3.2 Deploying on Oracle WebLogic Application Server (Clustered)

In this section, you will deploy a clustered Community-Gadgets application on Oracle WebLogic Application Server.

> **Note:** Consider the following:
>
> - If you need detailed steps on deploying web applications on Oracle WebLogic Application Server, refer to the Oracle WebLogic Application Server documentation.
> - Before deploying the Community-Gadgets application on WebLogic application server, add the following JVM parameter to `JAVA_OPTIONS` of WebLogic application server: `-DUseSunHttpHandler=true`

1. Before deploying the Community-Gadgets application, ensure that each application server's classpath points to the respective Community-Gadgets configuration directories (created in Section 16.2, "Copying Installer-Generated Configuration Files.").

   For reference, this guide uses the configurations described in Table 17–9.

Table 17–9 Classpath Configuration for Managed Servers (Clustered WebLogic)

Managed Server	Server Classpath
`<cgA_M1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgA_M1>:${CLASSPATH}"`
`<cgB_P1>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgB_P1>:${CLASSPATH}"`
`<cgA_M2>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgA_M2>:${CLASSPATH}"`
`<cgB_P2>`	`CLASSPATH="/<cg_install_dir>/<cg-standalone-configs>/<cgB_P2>:${CLASSPATH}"`

2. The Community-Gadgets installer creates `war` and `ear` files in the `<cg_install_dir>/deploy/management` and `<cg_install_dir>/deploy/production` directories. Extract the contents of the `war` files and deploy them via the command line or the Administration Console. Locations of the `war` files are shown in Table 17–10.

Table 17–10 WAR File Locations for WebLogic (Clustered)

Target Servers: All Cluster Members	Location of WAR Files	Sample Deployment Location

`<cgA_M1>`
`<cgA_M2>`
	`<cg_install_dir>/deploy/management/cg.war`
	`<cg_install_dir>/deploy/management/shindig.war`
		`<WL_HOME>/user_projects/domains/<cgMgmt>/applications/cg`
		`<WL_HOME>/user_projects/domains/<cgMgmt>/applications/shindig`

`<cgB_P1>`
`<cgB_P2>`
	`<cg_install_dir>/deploy/production/cg.war`
	`<cg_install_dir>/deploy/production/shindig.war`
	`<cg_install_dir>/deploy/production/cas.war`
		`<WL_HOME>/user_projects/domains/<cgProd>/applications/cg`
		`<WL_HOME>/user_projects/domains/<cgProd>/applications/shindig`
		`<WL_HOME>/user_projects/domains/<cgProd>/applications/cas`

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 13, "Prerequisites for Installing Community-Gadgets."

4. Copy the following files from the Community-Gadgets application to its application server's JRE: `jaxb-api-2.2.2.jar` file and `jaxb-impl-2.2.3-1.jar`. The files must be placed into the `<WebLogic_JDK>/jre/lib/endorsed` folder (if the `endorsed` folder does not exist, create it).

5. Start the production and management deployment servers.

   For example, in Linux:

   On `<ServerB>`:

```
cd <WL_HOME>/user_projects/domains/<cgProd>/bin
./startManagedServer <cgB_P1> http://<ServerB>:7001
cd <WL_HOME>/user_projects/domains/<cgProd>/bin
./startManagedServer <cgB_P2> http://<ServerB>:7001
```

   On `<ServerA>`:

```
cd <WL_HOME>/user_projects/domains/<cgMgmt>/bin
./startManagedServer <cgA_M1> http://<ServerA>:7001
cd <WL_HOME>/user_projects/domains/<cgMgmt>/bin
./startManagedServer <cgA_M2> http://<ServerA>:7001
```

6. Continue to Chapter 18, "Verifying and Implementing Community-Gadgets."

17.3.3 Deploying on IBM WebSphere Application Server (Clustered)

In this section, you will deploy a clustered Community-Gadgets application on IBM WebSphere Application Server.

Note: If you need detailed steps on deploying web applications on IBM WebSphere Application Server, refer to the vendor's documentation.

1. Before deploying the Community-Gadgets application, ensure that each application server's classpath points to the respective Community-Gadgets configuration directories (created in Section 16.2, "Copying Installer-Generated Configuration Files."). To do so:

   a. Log in to the deployment manager (for example, access `http://<ServerA>:9060/ibm/console`).

   b. For each server, select **Java and Process Management**, then select **Process Definition**, then select **Java Virtual Machine** and edit the `CLASSPATH` field.

      For reference, this guide uses the configurations described in Table 17–11.

Table 17–11 Classpath Configuration for WebSphere (Clustered)

Managed Server	Server Classpath
`<cgA_M1>`	`/<cg_install_dir>/<cg-standalone-configs>/<cgA_M1>`
`<cgB_P1>`	`/<cg_install_dir>/<cg-standalone-configs>/<cgB_P1>`
`<cgA_M2>`	`/<cg_install_dir>/<cg-standalone-configs>/<cgA_M2>`
`<cgB_P2>`	`/<cg_install_dir>/<cg-standalone-configs>/<cgB_P2>`

2. The Community-Gadgets installer creates `war` and `ear` files in the `<cg_install_dir>/deploy/management` and `<cg_install_dir>/deploy/production` directories. Locations of the `ear` files are shown in Table 17–12.

Table 17–12 EAR File Locations for WebSphere (Clustered)

Target Server	Location of EAR Files

`<wscg_M> cluster:`
`<cgA_M1>,<cgA_M2>`
	`<cg_install_dir>/deploy/management/CommunityGadgets.ear`
	`<cg_install_dir>/deploy/management/Shindig.ear`

`<wscg_P> cluster:`
`<cgB_P1>,<cgB_P2>`
	`<cg_install_dir>/deploy/production/CommunityGadgets.ear`
	`<cg_install_dir>/deploy/production/Shindig.ear`
	`<cg_install_dir>/deploy/production/cas.ear`

3. Complete the following steps in the WebSphere Admin Console for each Community-Gadgets application, Shindig application, and Visitors CAS application:

   a. Select **Applications**, then select **Application Types**, then select **WebSphere enterprise applications**, then select ***application_name***.

   b. Under Web Module Properties, select **Session management**.

   c. Under General Properties, select **Enable Cookies** and change the value for Cookie path from / to `/<application-context-root>` (the default value for the Community-Gadgets application is `/cg`, for Shindig is `/shindig`, and for the Visitors CAS application it is`/cas`).

   d. Click **Apply and Save**.

   e. Under General Properties, select **Override session management**.

   f. Click **Apply and Save**.

4. Copy the following files from Community-Gadgets to its application server's JRE: `axb-api-2.2.2.jar`, `jaxb-impl-2.2.3-1.jar` and `asm-3.1.jar`. The files must be

placed into the `<WebSphere_JDK>/jre/lib/endorsed` folder (if the `endorsed` folder does not exist, create it).

5. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 13, "Prerequisites for Installing Community-Gadgets."

6. Start the Deployment Manager profile and nodes.

   For example, in Linux:

   On `<ServerA>`:

   ```
   <WS_HOME>/bin
   ./startManager.sh -profileName <Dmgr01>
   ./startNode.sh -profileName <AppSrv01>
   ```

   On `<ServerB>`:

   ```
   ./startNode.sh -profileName <AppSrv02>
   ```

7. Start the production and management servers.

   For example, in Linux:

   On `<ServerB>`:

   ```
   ./startServer.sh cgB_P1 -profileName <AppSrv02>
   ./startServer.sh cgB_P2 -profileName <AppSrv02>
   ```

   On `<ServerA>`:

   ```
   ./startServer.sh cgA_M1 -profileName <AppSrv01>
   ./startServer.sh cgA_M2 -profileName <AppSrv01>
   ```

8. Continue to Chapter 18, "Verifying and Implementing Community-Gadgets."

18

Verifying and Implementing Community-Gadgets

Having deployed Community-Gadgets, you will verify the installation and display its interfaces by authorizing users to work with the Community WEM application and the Gadgets WEM application. You also have the option to integrate Community-Gadgets with the WebCenter Sites Contributor interface to deploy gadgets and widgets directly from the Contributor interface, as well to customize the deployed widgets.

This chapter contains the following sections:

18.1 Verifying Community-Gadgets URLs

In this section, you will verify that your newly installed Community-Gadgets application is operating correctly.

> **Note:** The URLs in this section are used only to verify that Community-Gadgets is correctly installed. The URLs cannot be used to invoke the Community WEM application or the Gadgets WEM application for management operations (an error will be displayed).
>
> For security, after completing the verification process, ensure that the URLs cannot be accessed externally. Internal access, however, may still be helpful for troubleshooting.

1. Start application servers in the following order:

> **Note:** In this section, we first start all servers, then verify all the URLs. For simplicity, you can start and test each server one at a time. This way, if a problem exists, you can correct it before starting the rest of the servers.

a. Start the WebCenter Sites load balancers (assuming your installation is clustered).

b. Start the WebCenter Sites application servers (production and management).

> **Note:** Ensure that the servers are properly started and you can login as the user provided to Community-Gadgets on both servers.
>
> It is recommended that WebCenter Sites on the production system is backed up before continuing with the Community-Gadgets installation.

c. Start the Community-Gadgets load balancers (assuming your installation is clustered).

d. Start a single Community-Gadgets (production) application server instance.

> **Note:** Monitor all logs on WebCenter Sites on the production system and the Community-Gadgets production server and applications. Resolve errors reported in the logs before continuing. Failing to do so may leave Community-Gadgets in an unknown state.

e. Start the remaining Community-Gadgets (production) application server instances.

f. Start a single Community-Gadgets (management) application server instance.

g. Start the remaining Community-Gadgets (management) application server instances.

2. At this point, all instances should be running, and you can verify the individual URLs, and then test the load balancer.

3. Verify the following for all deployed instances of Visitors CAS:

   a. Verify that the Visitors CAS application login form is displayed. You cannot yet log in:

   ```
   http://<production_cg_host>:<production_cg_port>/<production_cg_cas_context
   root>/login
   ```

   For example:

   ```
   http://192.0.2.1:8080/cas/login
   ```

   b. Verify the status of the Community-Gadgets application on each node of the production and management systems. It is best to start with production, then move to management:

   ```
   http://<cg_node_host>:<cg_node_port>/<cg_context_root>/status
   ```

   For example:

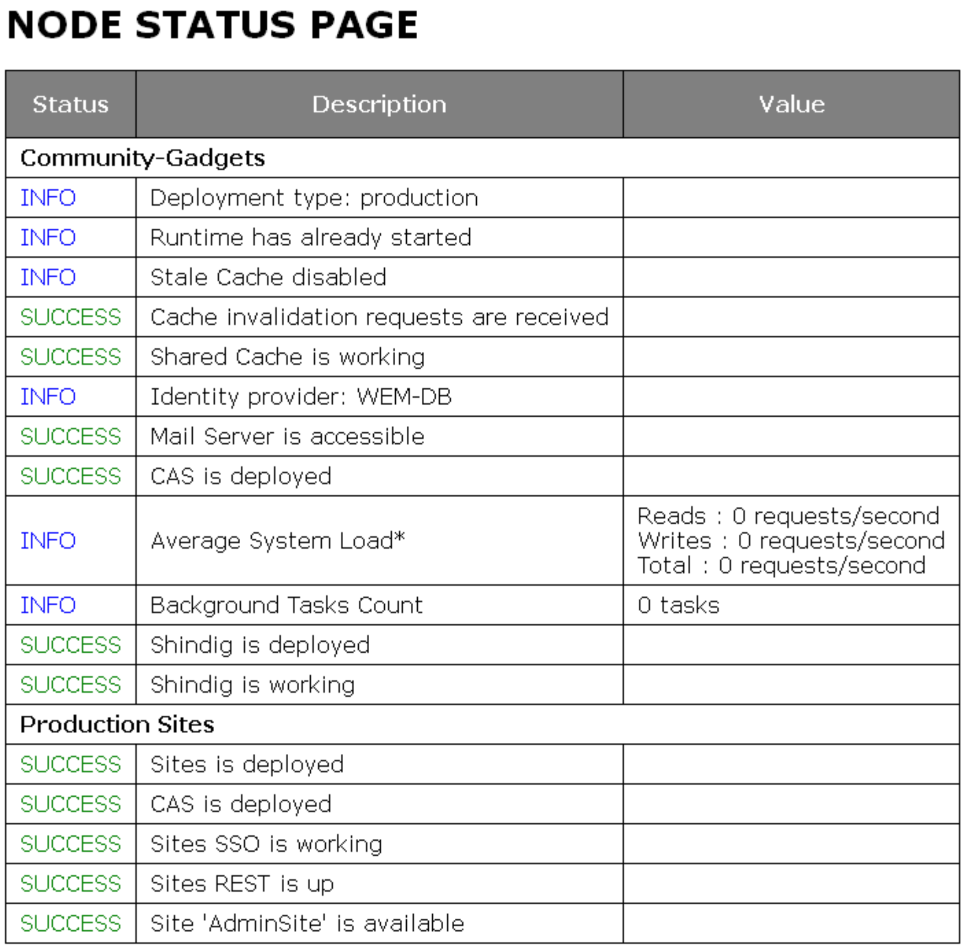
   Node1: `http://192.0.2.1:8080/cg/status`

   Node2: `http://192.0.2.1:8080/cg/status`

   A screen similar to Figure 18–1 should be displayed with all the options showing either `SUCCESS` or `INFO`. If any option displays another status (`ERROR` or `WARNING`) or the page does not load within 90 seconds, then you cannot continue. If the status for Lucene is "Lucene is not working", or the "Processing..." status is shown for more than 2 minutes, then you cannot continue. Configure Lucene engine on the WebCenter Sites production instances, as described in the section "Configuring Lucene Parameters" of the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Note: If you see any errors or warnings, review the Community-Gadgets log file and the application server's logs for details on what has failed.

Figure 18–1 Node Status Page



NODE STATUS PAGE

Status	Description	Value
Community-Gadgets		
INFO	Deployment type: production	
INFO	Runtime has already started	
INFO	Stale Cache disabled	
SUCCESS	Cache invalidation requests are received	
SUCCESS	Shared Cache is working	
INFO	Identity provider: WEM-DB	
SUCCESS	Mail Server is accessible	
SUCCESS	CAS is deployed	
INFO	Average System Load*	Reads : 0 requests/second Writes : 0 requests/second Total : 0 requests/second
INFO	Background Tasks Count	0 tasks
SUCCESS	Shindig is deployed	
SUCCESS	Shindig is working	
Production Sites		
SUCCESS	Sites is deployed	
SUCCESS	CAS is deployed	
SUCCESS	Sites SSO is working	
SUCCESS	Sites REST is up	
SUCCESS	Site 'AdminSite' is available	

4. On each Management node, log in to the Community user interface as a general administrator (the default user is `fwadmin`. If you entered the credentials of a different administrator during the installation process, use those credentials):

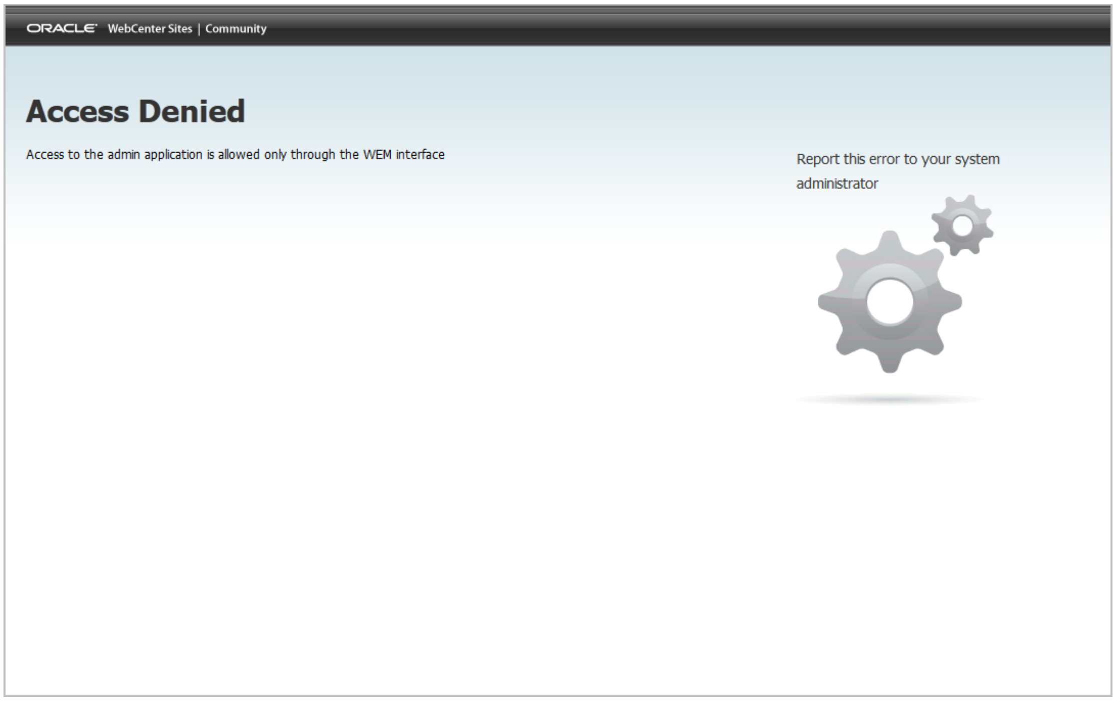
```
http://<cg_node_host>:<cg_node_port>/cg/admin/start.jsp?siteid=<Site to which
Community WEM application will be assigned>
```

   For example:

```
http://192.0.2.1:90/cg/admin/start.jsp?siteid=avisports
```

   Once logged in, you should see a message stating that you did not access the Community user interface through the WEM Framework (Figure 18–2).

Figure 18–2 Community User Interface Accessed from Outside the WEM Framework



Access Denied

Access to the admin application is allowed only through the WEM interface

Report this error to your system administrator

Note: If the Community user interface is not displayed, or you get an error message other than "`Access Denied`," then a login error exists and you cannot continue.

If any failures are seen, review the Community-Gadgets application's log file and the application server's logs for details on what has failed.

---

5. On each management node, log in to the Gadgets administrator interface (Global Catalog) as a general administrator (the default user is `fwadmin`. If you entered the credentials of a different administrator during the installation process, use those credentials):

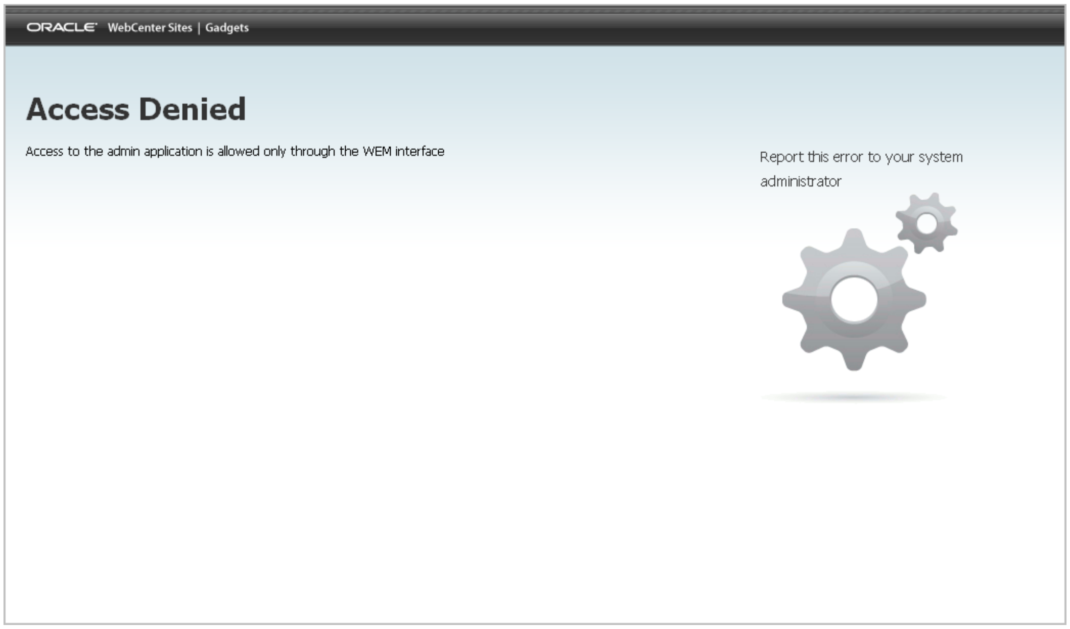
   ```
   http://<cg_node_host>:<cg_node_port>/cg/admin-gadgets/start.jsp?siteid=<Site to
   which Gadgets WEM application will be assigned>
   ```

   For example:

   ```
   http://192.0.2.1/cg/admin-gadgets/start.jsp?siteid=avisports
   ```

   Once logged in, you should see a message stating that you did not access the Gadgets administrator interface through the WEM Framework (Figure 18–3).

Figure 18–3 Gadgets Administrator Interface Accessed from Outside the WEM Framework



Access Denied

Access to the admin application is allowed only through the WEM interface

Report this error to your system administrator

Note: If the Gadgets administrator interface is not displayed, or you get an error message other than `"Access Denied,"` then a login error exists and you cannot continue.

If any failures are found, review the Community-Gadgets log file and the application server's logs for details on what has failed.

18.2 Registering the Community-Gadgets Application

During first launch (when the Community-Gadgets application server is first started), Community-Gadgets registers itself with the management WebCenter Sites instance. It creates two WEM applications:

- **Community** – This WEM application provides the Community user interface.
- **Gadgets** – This WEM application provides the Gadgets user interface and the Gadgets administrator interface (Global Catalog).

Your next steps are to display the interfaces of the Community WEM application and the Gadgets WEM application. For instructions, continue to Section 18.3, "Authorizing Users to Access Community and Gadgets Interfaces."

18.3 Authorizing Users to Access Community and Gadgets Interfaces

In this section, you will display the Community and Gadgets interfaces by authorizing users to work with the Community WEM application and Gadgets WEM application, and logging in as the users to access the interfaces.

Note: Detailed information about user authorization is available in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*. For a quick start, follow the steps below.

Throughout this section, you will be using the WEM Admin interface to authorize users as follows:

- Assign the Community WEM application and its users to the same site (`<app_site>`) via a common role. Sharing at least one role to an application and a user on the same site grants the user access to the application on that site.
- Assign the Gadgets WEM application and its users to the same site (`<app_site>`) via a common role.
- Assign the Gadgets WEM application to `<base_site>` to display the Gadgets administrator interface (Global Catalog).

To authorize users, complete the following steps:

> **Note:** If your management WebCenter Sites system is using read-only LDAP, manually add the `Moderator` role to the LDAP database. This role is required to give users moderator rights in the Community application. All other Community-Gadgets roles (`GeneralAdmin`, `SiteAdmin`, and `Designer`) are pre-defined in WebCenter Sites.

1. Assign the Community WEM application to `<app_site>`:
   a. Starting with the Apps page of the WEM Admin interface, point to the **Community** application name and select **Manage App**.
   b. In the Manage App screen, click **Assign to Sites**.
   c. Select the `<app_site>` site to which you are assigning the Community WEM application.
   d. Select the following roles for the Community application: **GeneralAdmin, SiteAdmin**, **Moderator**, and **Designer**
2. Assign the Gadgets WEM application to `<app_site>`:
   a. Starting with the Apps page of the WEM Admin interface, point to the **Gadgets** application name and select **Manage App**.
   b. In the Manage App screen, click **Assign to Sites**.
   c. Select the `<app_site>` site to which you are assigning the Gadgets WEM application.
   d. Select the following roles for the Gadgets application: **GeneralAdmin**, **SiteAdmin**, and **Designer**.
3. Assign the Gadgets WEM application to `<base_site>`:
   a. Starting with the Apps page of the WEM Admin interface, point to the **Gadgets** application name and select **Manage App**.
   b. In the Manage App screen, click **Assign to Sites**.
   c. Select the `<base_site>` site to which you are assigning the Gadgets WEM application.
   d. Select the **GeneralAdmin** role for the Gadgets application.
4. Authorize users to access the Community WEM application:
   a. In the menu bar, select **Sites**.

b. Point to the site to which the Community WEM application is assigned and select **Manage Site Users**.

c. In the Manage Site Users screen, complete one or both of the following steps, as necessary:

   a. To assign a user to the site and assign roles to the user, click **Assign Users**.

   b. To assign roles to an existing site user, point to the user's name and select **Assign Roles to User**.

      In the Assign Roles to User screen, select at least one of the following roles: **GeneralAdmin**, **SiteAdmin**, **Designer**, **Moderator**.

      Click **Save and Close**.

> **Note:** If you assigned the `SiteAdmin` role to any of the users, verify that the users are members of the REST security group called `SiteAdmin_AdminSite`. For instructions on adding a user to a REST security group, see the part, "Administering the Web Experience Management Framework" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

5. Authorize users to access the Gadgets WEM application:

   a. In the menu bar, select **Sites**.

   b. Point to the site to which the Gadgets WEM application is assigned and select **Manage Site Users**.

   c. In the Manage Site Users screen, complete one or both of the following, as necessary:

      – To assign a user to the site and assign roles to the user, click **Assign Users**.

      – To assign roles to an existing site user, point to the user's name and select **Assign Roles to User**.

        In the Assign Roles to User screen, select at least one of the following roles: **GeneralAdmin**, **SiteAdmin**, **Designer** for `<app_site>`. Select **GeneralAdmin** role for `<base_site>`. For permissions associated with each role, see the chapter, "Welcome to the Oracle WebCenter Sites: Community Application" in the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

        Click **Save and Close**.

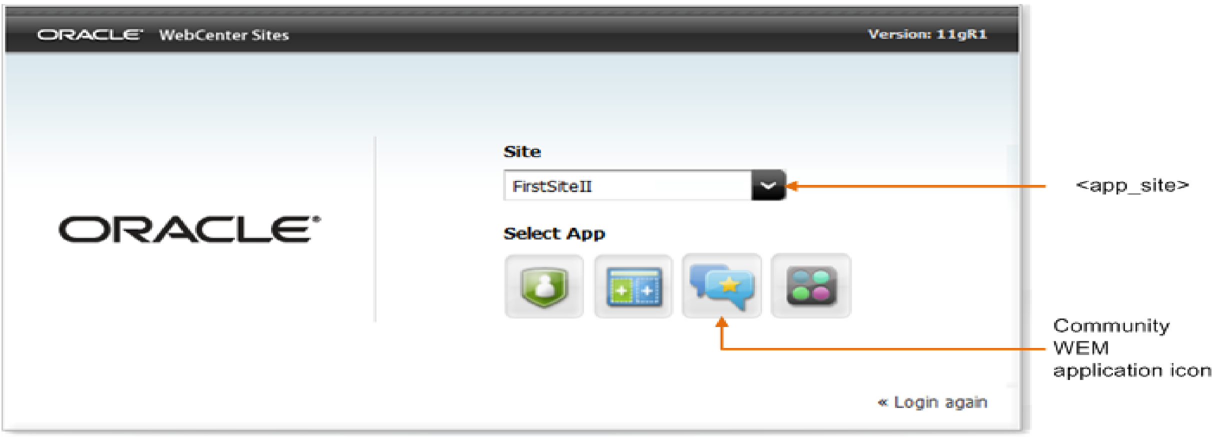
> **Note:** If you assigned the `SiteAdmin` role to any of the users, verify that the users are members of the REST security group called `SiteAdmin_AdminSite`. For instructions on adding a user to a REST security group, see the part, "Administering the Web Experience Management Framework" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

6. Verify each user's access to the Community WEM application on `<app_site>` as follows:

   a. Log in to WebCenter Sites as the Community user that you authorized in the steps above.

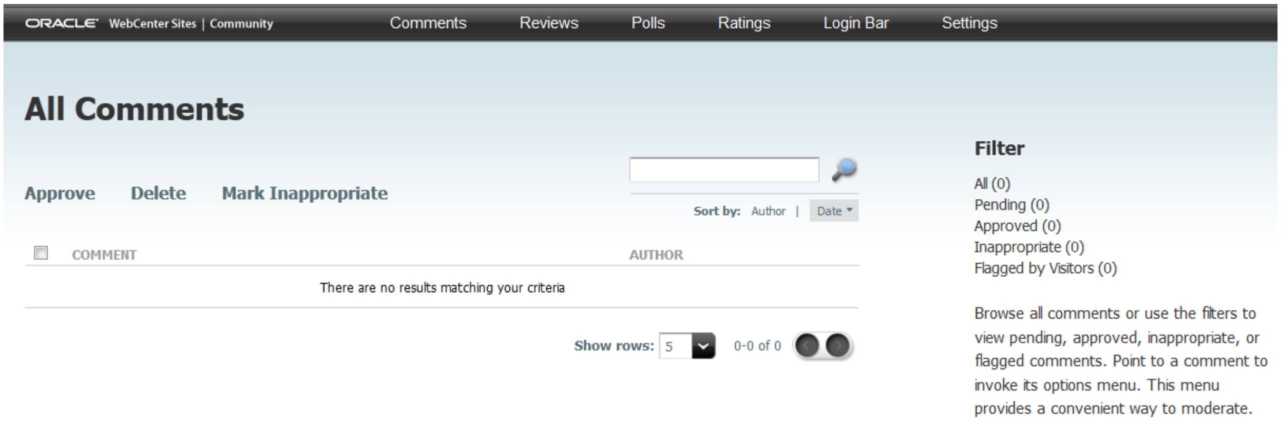
b. Select <app_site> and click the Community application icon, highlighted in Figure 18–4.

Figure 18–4 Site Drop-down List and Community Application Icon



c. The "All Comments" screen is displayed (Figure 18–5). For users with the Designer role, the "All Polls" screen is displayed. If you see the named screen, you have completed authorizing the current user to access the Community interface. Continue to the next step.

Figure 18–5 All Comments Screen



All Comments

There are no results matching your criteria

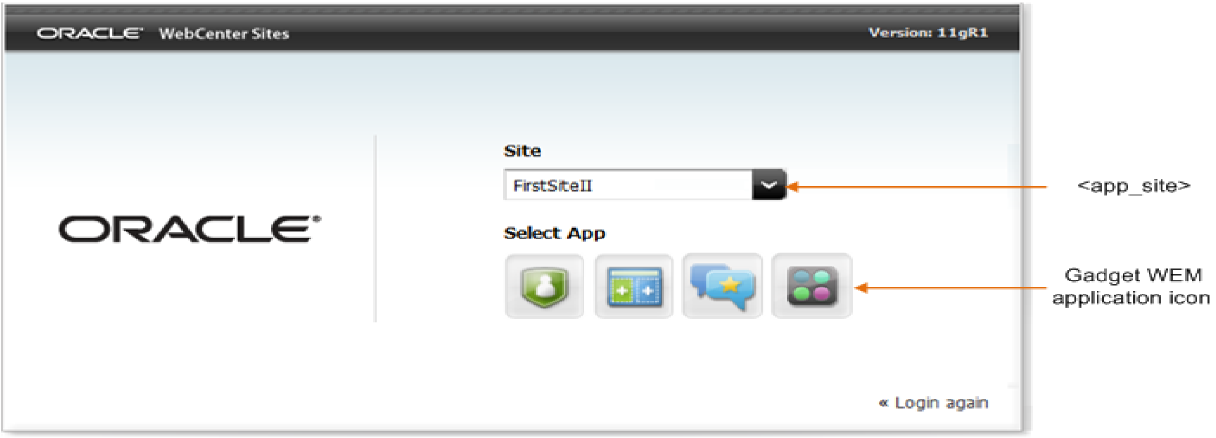
Browse all comments or use the filters to view pending, approved, inappropriate, or flagged comments. Point to a comment to invoke its options menu. This menu provides a convenient way to moderate.

7. Verify each user's access to the Gadgets WEM application on <app_site> and <base_site> as follows:

   a. Log in to WebCenter Sites as a Gadgets user that you authorized in the steps above.

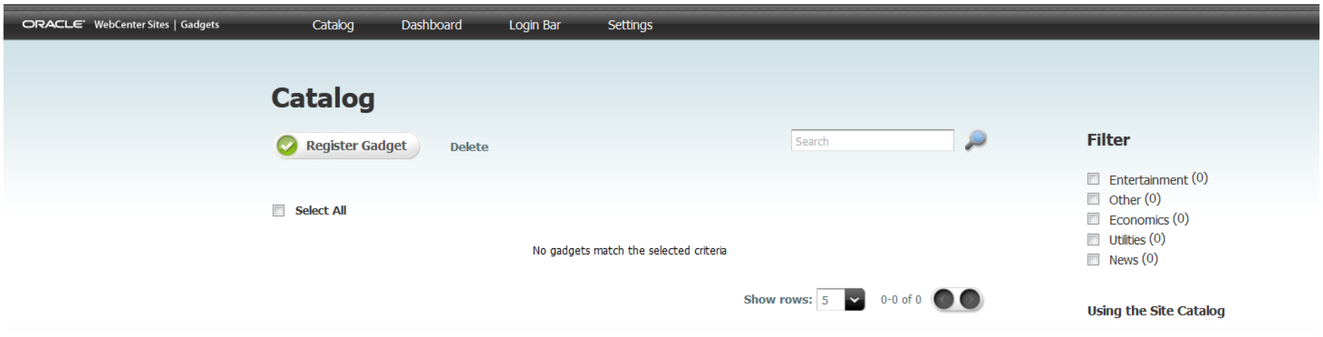
   b. Select <app_site> and click the Gadgets application icon, highlighted in Figure 18–6.

Figure 18–6 Site Drop-down List and Gadgets Application Icon



c. If you see the Catalog screen (Figure 18–7), you have completed authorizing the current user to access the Gadgets WEM application on <app_site>. Continue with the next step.

Figure 18–7 Gadgets User Interface

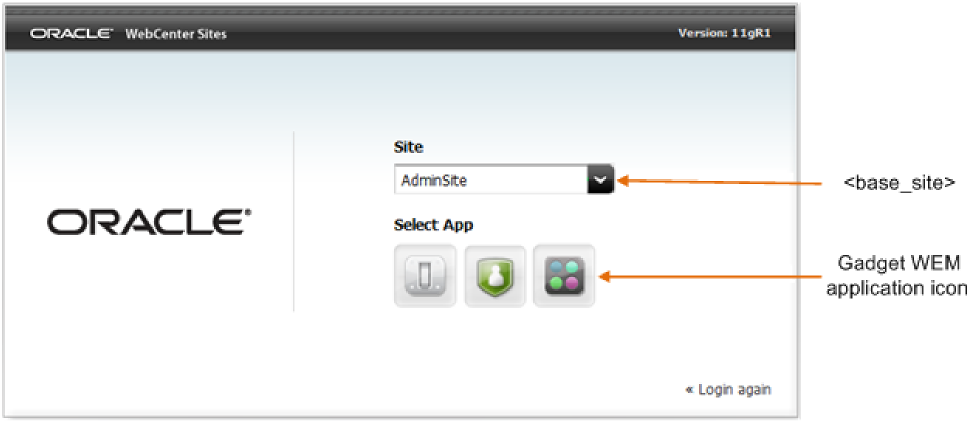


8. Verify the general administrator's access to the Gadgets WEM application on <base_site> as follows:

   a. Log in to WebCenter Sites as a general administrator.

   b. Select <base_site> and click the Gadgets application icon, highlighted in Figure 18–8.

Figure 18–8 Site Drop-down List and Gadgets Application Icon

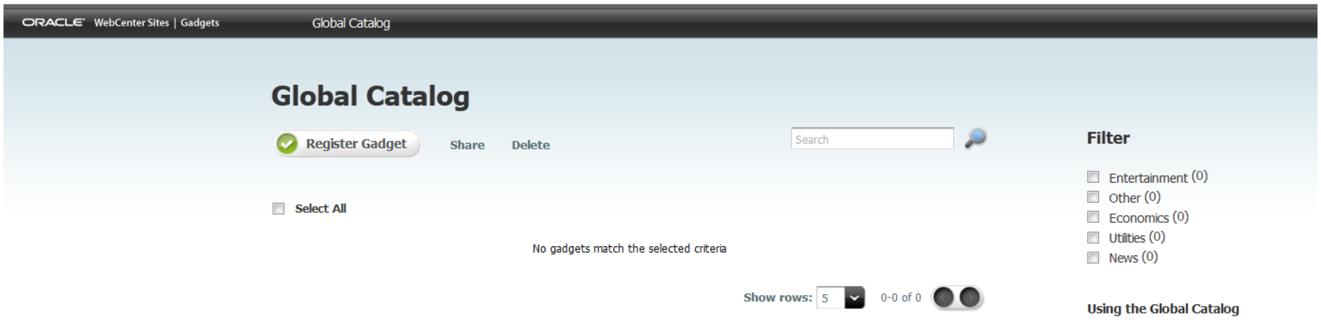


<base_site>

Gadget WEM application icon

c. The Gadgets administrator interface (Global Catalog) is displayed (Figure 18–9). Continue to the next step.

Figure 18–9 Gadgets Administrator Interface



9. If you wish to assign the Community WEM application or Gadgets WEM application to additional sites, first create or select the sites on the WebCenter Sites content management system, then mirror publish the sites to the WebCenter Sites production system, and repeat the steps in this section.

18.4 Integrating Community-Gadgets into the WebCenter Sites Contributor Interface

If you integrate the management Community-Gadgets web application into the WebCenter Sites Contributor interface, users can deploy widgets and gadgets directly from the WebCenter Sites Contributor interface.

To integrate Community-Gadgets into the WebCenter Sites Contributor Interface

1. Add the Community-Gadgets URLs to the `futuretense_xcel.ini` property file:

   a. Open the `futuretense_xcel.ini` file (located in the management WebCenter Sites installation folder) in the WebCenter Sites Property Editor.

   b. Set `cg.management.url` to specify the URL for the management Community-Gadgets web application.

   If you are not using the Property Editor to modify the file, the syntax of the property value is the following, where the colon character (:) must be escaped with a backslash (\):

```
cg.management.url=<protocol>\://<cg_mgmt_host>\:<cg_mgmt_port>/<cg_mgmt_
context>
```

For example:

```
cg.management.url=http\://cgmgmt.example.com\:8280/cg
```

c. Set the `cg.production.url` to specify the URL of the production Community-Gadgets web application.

   If you are not using the Property Editor to modify the file, the syntax of the property values is the following, where the colon character (:) must be escaped with a backslash (\):

```
cg.production.url=<protocol>\://<cg_prod_host>\:<cg_prod_port>/<cg_prod_
context>
```

For example:

```
cg.production.url=http\://cgprod.example.com\:8380/cg
```

2. Restart the application server for both the management and production WebCenter Sites instances.

3. If you have not assigned the Community and Gadgets WEM applications to the site on which you wish to integrate Community-Gadgets with the Contributor interface, follow the steps in Section 18.3, "Authorizing Users to Access Community and Gadgets Interfaces."

4. Enable the Community-Gadgets asset types and "Search" start menu items:

   a. Log in to a site on which you enabled either the Community interface or the Gadgets interface as a user assigned one of the following roles: **GeneralAdmin**, **SiteAdmin**, or **Designer**.

   b. Select the icon for either the Community interface or the Gadgets interface.

      Upon first access of either the Community interface or the Gadgets interface, the following Community-Gadgets asset types are automatically enabled on all the necessary sites: `CGGadget`, `CGPoll`, and `CGWidget`. The "Search" start menu items for these asset types are also enabled automatically.

5. Verify the Community-Gadgets integration with the Contributor interface:

   a. Log in to the WebCenter Sites Contributor interface as a user assigned one of the following roles: **GeneralAdmin**, **SiteAdmin**, or **Designer**.

   b. Verify that the **Community-Gadgets** tree pane is displayed in the left navigation pane and that the **Community-Gadgets** tree contains widgets and gadgets.

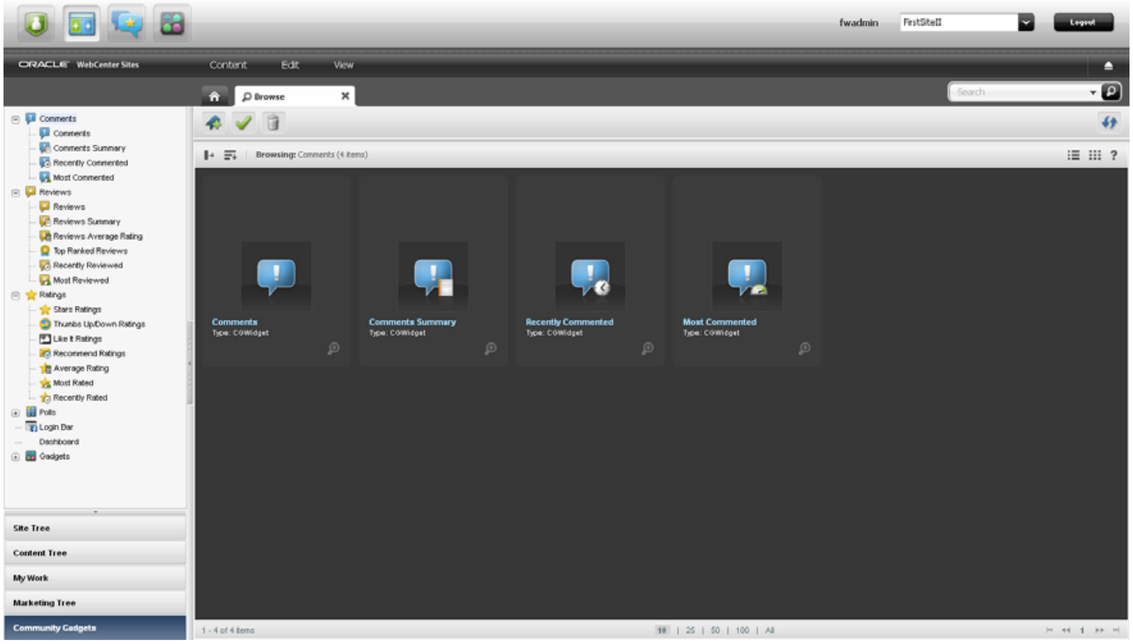
   c. Verify that the search field displays options for finding the following asset types: `CGGadget`, `CGWidget` and `CGPoll`.

   d. Perform a search for the `CGWidget` asset type and verify that the search results return relevant data.

The WebCenter Sites Contributor interface should look similar to the one shown in Figure 18–10. The **Community-Gadgets** tree should list the available widgets and gadgets for the site. Those widgets and gadgets can now be deployed directly from the Contributor interface. For more information about the **Community-Gadgets** tree, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

Note: If the Contributor interface does not display the **Community-Gadgets** tree, or if the Community-Gadgets asset types and/or "Search" start menu items are not accessible from the Contributor interface, verify the username and password for the `xcelerate.batchuser` and `xcelerate.batchpass` parameters in the `futuretense_xcel.ini` file. For more information about these parameters, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

Figure 18–10 Community-Gadgets Tree in the WebCenter Sites Contributor Interface



19 Adding Sample Gadgets

This chapter contains the following sections:

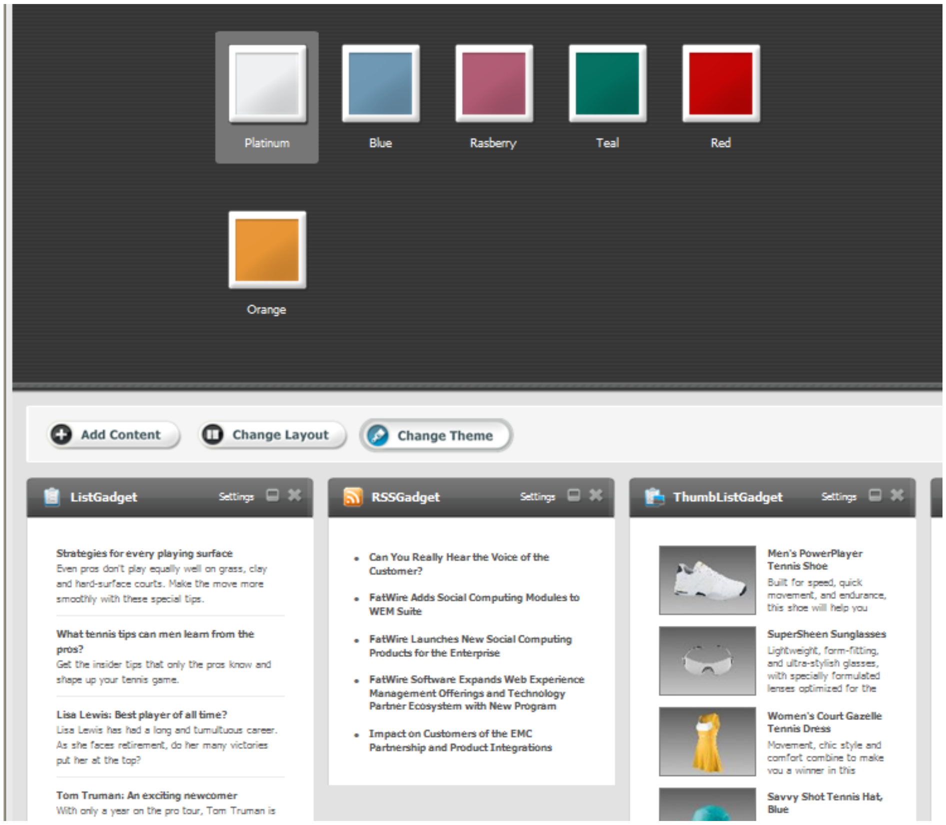
- Section 19.1, "Prerequisites for Installing Sample Gadgets"
- Section 19.2, "Installing the Sample Gadgets"

19.1 Prerequisites for Installing Sample Gadgets

WebCenter Sites ships with the following sample gadgets (Figure 19–1):

`ListGadget`, `RSSGadget`, `ThumbListGadget`, and `SlideshowGadget`.

Figure 19–1 WebCenter Sites Gadgets



Strategies for every playing surface
Even pros don't play equally well on grass, clay and hard-surface courts. Make the move more smoothly with these special tips.

What tennis tips can men learn from the pros?
Get the insider tips that only the pros know and shape up your tennis game.

Lisa Lewis: Best player of all time?
Lisa Lewis has had a long and tumultuous career. As she faces retirement, do her many victories put her at the top?

Tom Truman: An exciting newcomer
With only a year on the pro tour, Tom Truman is

- Can You Really Hear the Voice of the Customer?
- FatWire Adds Social Computing Modules to WEM Suite
- FatWire Launches New Social Computing Products for the Enterprise
- FatWire Software Expands Web Experience Management Offerings and Technology Partner Ecosystem with New Program
- Impact on Customers of the EMC Partnership and Product Integrations

Men's PowerPlayer Tennis Shoe
Built for speed, quick movement, and endurance, this shoe will help you

SuperSheen Sunglasses
Lightweight, form-fitting, and ultra-stylish glasses, with specially formulated lenses optimized for the

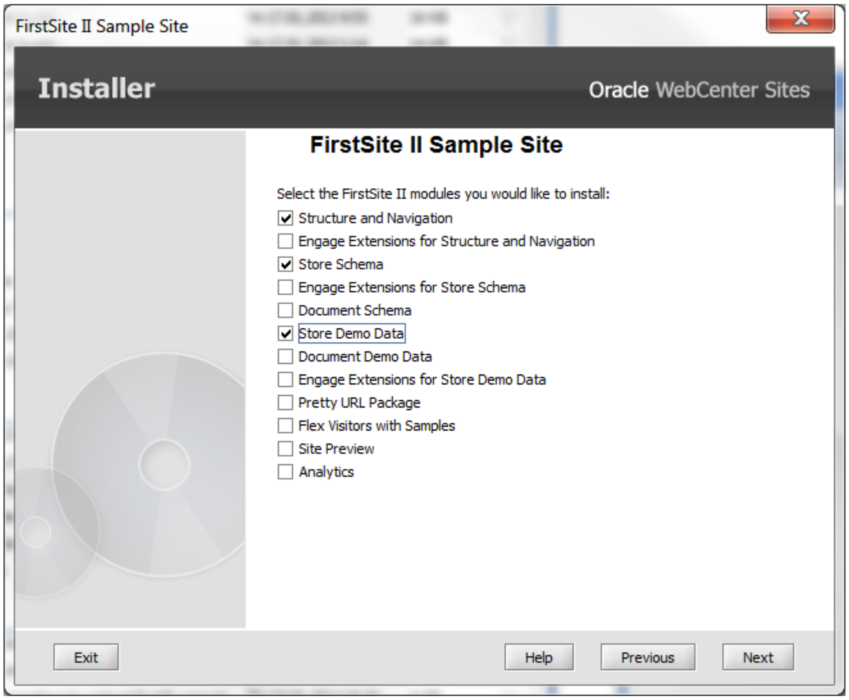
Women's Court Gazelle Tennis Dress
Movement, chic style and comfort combine to make you a winner in this

Savvy Shot Tennis Hat, Blue

The sample gadgets must be installed on the FirstSite II sample site. The site must be enabled to store demo data. If your management WebCenter Sites system does not have FirstSite II, do the following:

Prepare a WebCenter Sites system running FirstSite II with Store Demo Data enabled as shown in Figure 19–2.

Figure 19–2 FirstSite II Sample Site



FirstSite II Sample Site

Select the FirstSite II modules you would like to install:

☑ Structure and Navigation
□ Engage Extensions for Structure and Navigation
☑ Store Schema
□ Engage Extensions for Store Schema
□ Document Schema
☑ Store Demo Data
□ Document Demo Data
□ Engage Extensions for Store Demo Data
□ Pretty URL Package
□ Flex Visitors with Samples
□ Site Preview
□ Analytics

After installing the sample gadgets on your newly prepared WebCenter Sites system (see Section 19.2, "Installing the Sample Gadgets"), you will publish FirstSite II to the WebCenter Sites management system.

19.2 Installing the Sample Gadgets

Installing sample gadgets installs the following components:

- `FW_CSGadget` asset type (used to create gadgets)
- Sample gadgets (assets of type `FW_CSGadget`)
- Templates for rendering the gadgets
- Assets that provide content for the sample gadgets. The assets are instances of FirstSite II asset types.

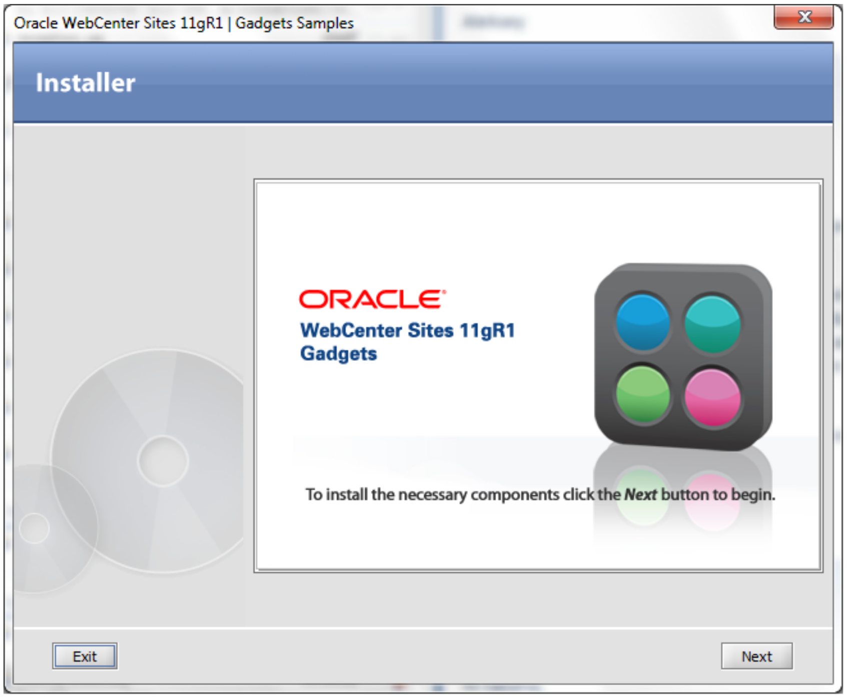
To install sample gadgets, complete the following steps:

1. Ensure that prerequisites in Section 19.1, "Prerequisites for Installing Sample Gadgets" are satisfied.
2. Extract the installation file named `WCS_GadgetServer_Samples.zip` to the host machine. Extracting this zip file creates a subdirectory named `gadgets_samples`.

Note: Do not change the name of this subdirectory or any of its subdirectories, and be sure to retain the archived directory structure. Otherwise, the installer will fail.

3. Run the Gadgets application integration installer script:
   - Windows: `gsInstall.bat`
   - UNIX: `gsInstall.sh`
4. In the "Welcome" screen (Figure 19–3), click **Next**.

Figure 19–3 Gadget Server Sample Gadgets



Oracle WebCenter Sites 11gR1 | Gadgets Samples

Installer

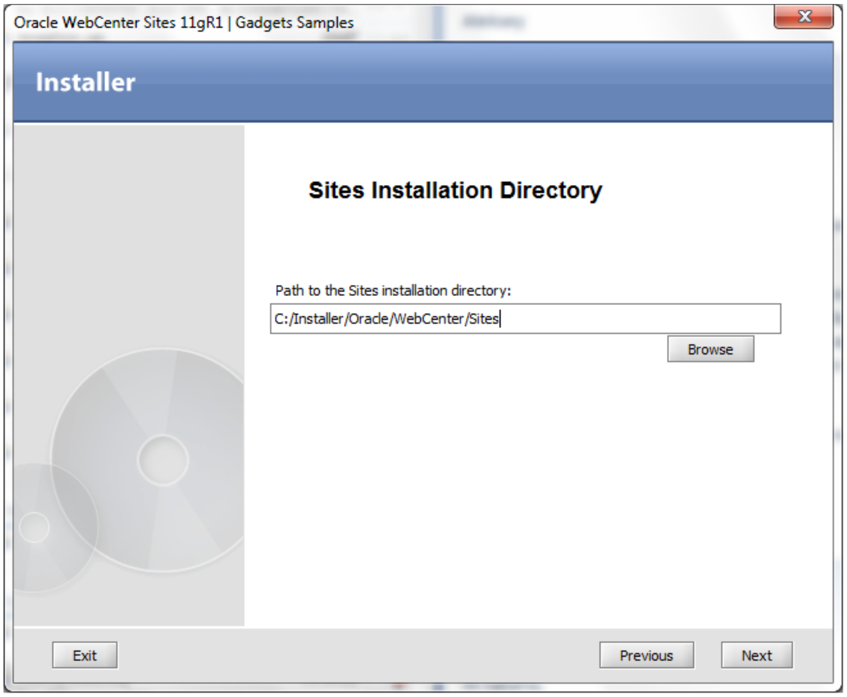
ORACLE
WebCenter Sites 11gR1
Gadgets

To install the necessary components click the *Next* button to begin.

5. In the "Sites Installation Directory" screen (Figure 19–4), enter the path to your production WebCenter Sites installation directory.

Note: You can also enter the path to the management WebCenter Sites installation directory. However, ensure that the production Community-Gadgets instance has the ability to access the sample gadgets installed on the management WebCenter Sites instance.

Figure 19–4 Sites Installation Directory



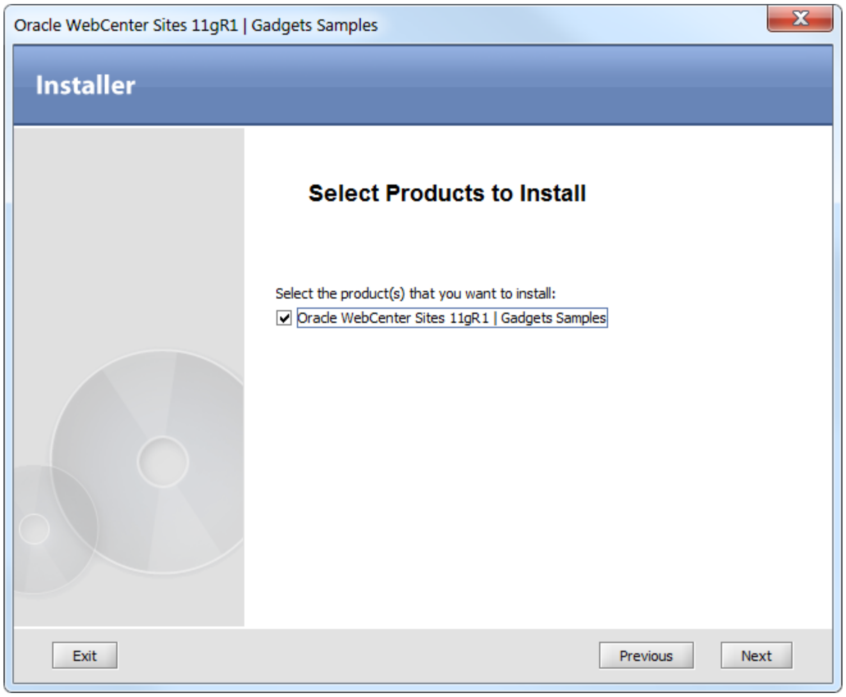
Sites Installation Directory

Path to the Sites installation directory:

C:/Installer/Oracle/WebCenter/Sites

6. In the "Select Products to Install" screen, select **Sample Gadgets** (Figure 19–5).

Figure 19–5 Products to Install



Select Products to Install

Select the product(s) that you want to install:

☑ Oracle WebCenter Sites 11gR1 | Gadgets Samples

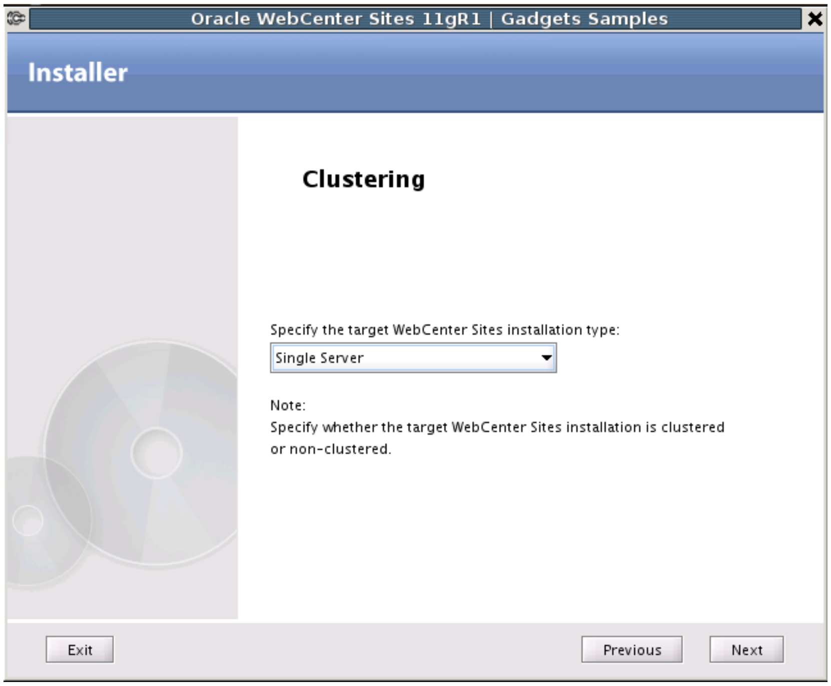
The following components will be installed:

- `FW_CSGadget` asset type (used to create gadgets)
- Sample gadgets (assets of type `FW_CSGadget`)

- Templates for rendering the gadgets
- Assets that provide content for the sample gadgets. The assets are instances of FirstSite II asset types.

7. In the "Clustering" screen (Figure 19–6), indicate whether you are installing the sample gadgets on a single server or a secondary cluster member. Click **Next**.

Figure 19–6 Clustering



Oracle WebCenter Sites 11gR1 | Gadgets Samples

Installer

Clustering

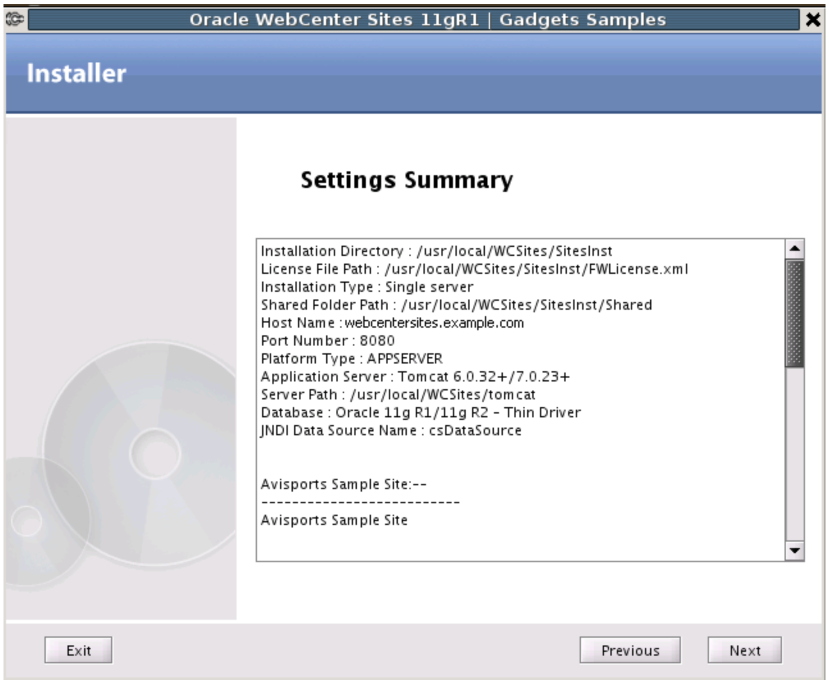
Specify the target WebCenter Sites installation type:

Single Server

Note:
Specify whether the target WebCenter Sites installation is clustered or non-clustered.

8. In the "Settings Summary" screen, click **Next** (Figure 19–7).

Figure 19–7 Settings Summary



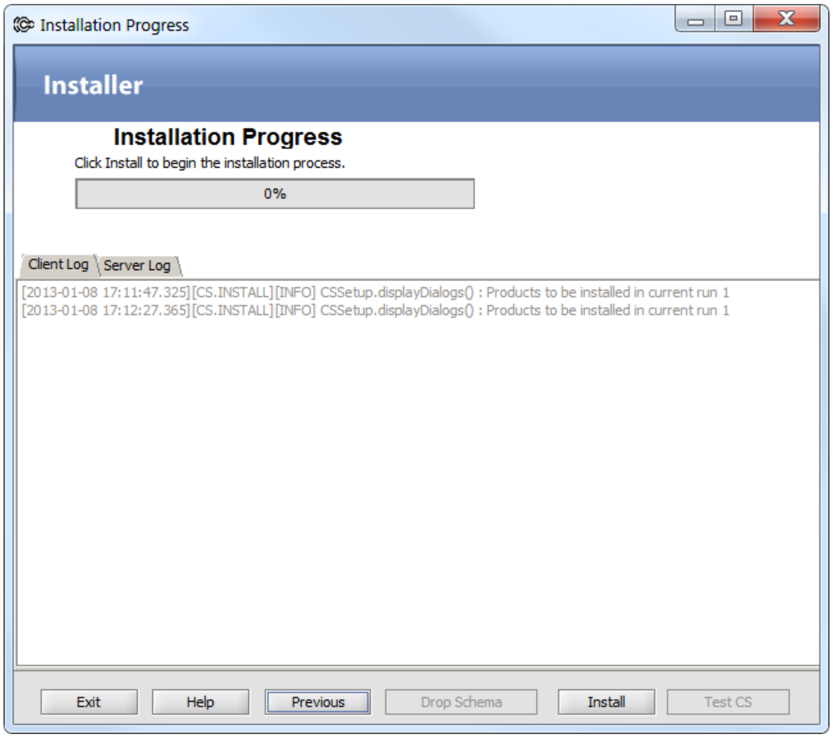
Installation Directory : /usr/local/WCSites/SitesInst
License File Path : /usr/local/WCSites/SitesInst/FWLicense.xml
Installation Type : Single server
Shared Folder Path : /usr/local/WCSites/SitesInst/Shared
Host Name : webcentersites.example.com
Port Number : 8080
Platform Type : APPSERVER
Application Server : Tomcat 6.0.32+/7.0.23+
Server Path : /usr/local/WCSites/tomcat
Database : Oracle 11g R1/11g R2 - Thin Driver
JNDI Data Source Name : csDataSource

Avisports Sample Site:--

Avisports Sample Site

9. In the "Installation Progress" screen, click **Install** to begin the installation (Figure 19–8).

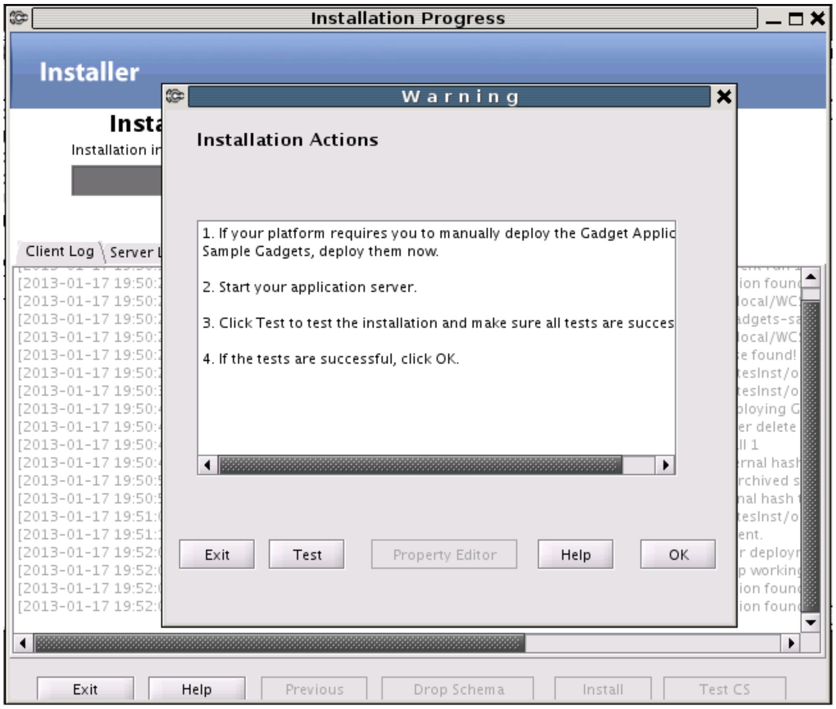
Figure 19–8 Installation Progress



```
[2013-01-08 17:11:47.325][CS.INSTALL][INFO] CSSetup.displayDialogs() : Products to be installed in current run 1
[2013-01-08 17:12:27.365][CS.INSTALL][INFO] CSSetup.displayDialogs() : Products to be installed in current run 1
```

10. When the "Installation Actions" dialog box opens, start your WebCenter Sites installation, and click **Test** to test your WebCenter Sites installation (Figure 19–9). Click **OK**.

Figure 19–9 Installation Actions



Installation Progress

Installer

Warning

Installation Actions

1. If your platform requires you to manually deploy the Gadget Applic
Sample Gadgets, deploy them now.

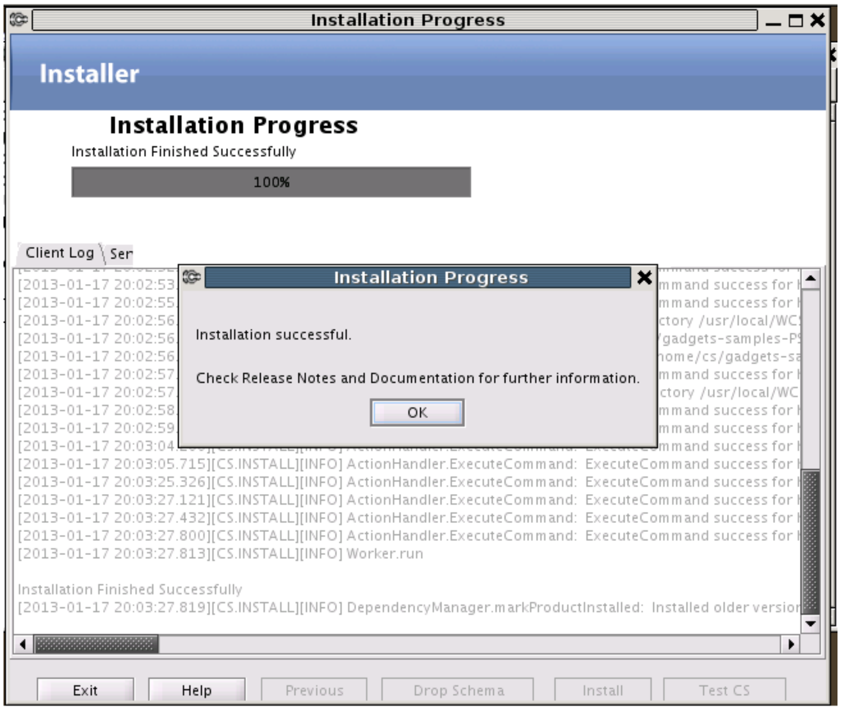
2. Start your application server.

3. Click Test to test the installation and make sure all tests are succes

4. If the tests are successful, click OK.

11. When the installation is complete (Figure 19–10), click **OK**.

Figure 19–10 Installation Successful



Installation Progress

Installer

Installation Progress

Installation Finished Successfully

100%

Installation successful.

Check Release Notes and Documentation for further information.

```
[2013-01-17 20:03:05.715][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for
[2013-01-17 20:03:25.326][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for
[2013-01-17 20:03:27.121][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for
[2013-01-17 20:03:27.432][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for
[2013-01-17 20:03:27.800][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for
[2013-01-17 20:03:27.813][CS.INSTALL][INFO] Worker.run

Installation Finished Successfully
[2013-01-17 20:03:27.819][CS.INSTALL][INFO] DependencyManager.markProductInstalled:  Installed older versio
```

12. Register the sample gadgets in either the Gadgets application's global gadget catalog or site gadget catalog. For instructions, see "Registering Gadgets" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

> **Note:** Gadget registration requires you to enter the gadgets' descriptor URLs. To obtain the URLs, follow the instructions below.

To obtain the sample gadgets' descriptor URLs:

a. Log in to WebCenter Sites with the credentials of a general administrator (for example, `fwadmin/xceladmin`). Select the FirstSite II sample site and then select the icon for the **Contributor** interface

b. In the "Search" field, click the **down-arrow** to open the "Search Type" menu.

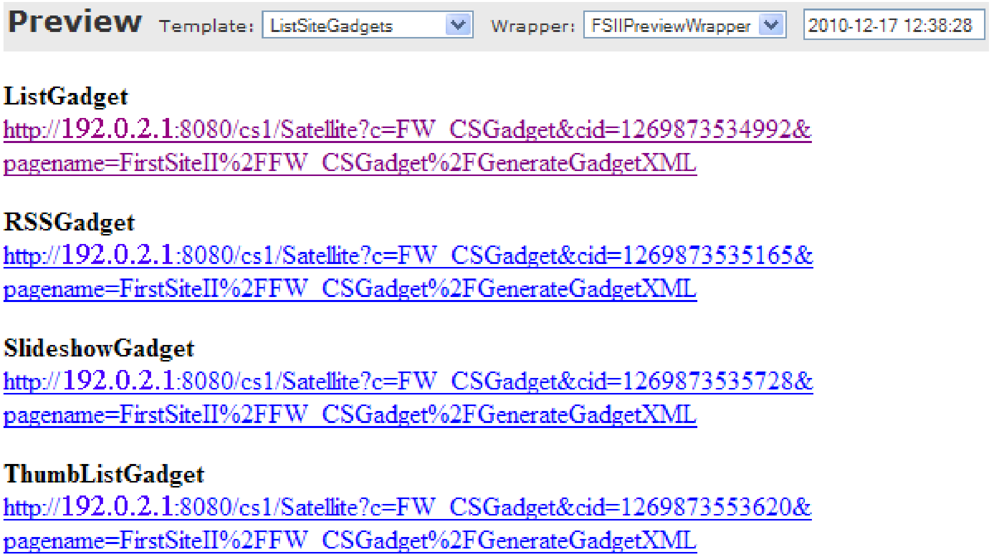
c. In the "Search Type" menu, select **Find CS-Based Gadget**.

d. In the "Search" field, click the **magnifying glass** button.

e. In the search results list, right-click any of the sample gadgets and select **Preview**.

f. In the "Select preview template" dialog box, select **ListSiteGadgets** to display the list of gadget descriptor URLs (Figure 19–11).

Figure 19–11 List of Gadget Descriptor URLs



Preview Template: ListSiteGadgets Wrapper: FSIIPreviewWrapper 2010-12-17 12:38:28

ListGadget
http://192.0.2.1:8080/cs1/Satellite?c=FW_CSGadget&cid=1269873534992&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

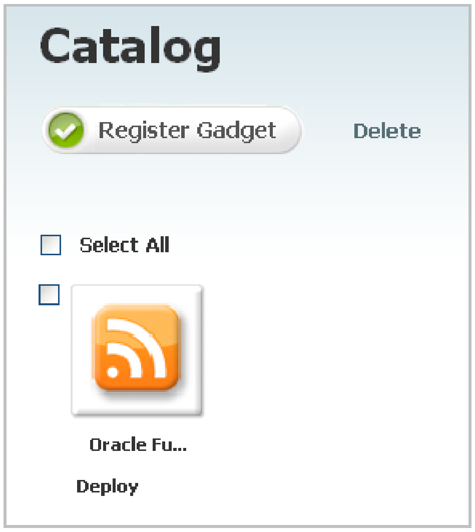
RSSGadget
http://192.0.2.1:8080/cs1/Satellite?c=FW_CSGadget&cid=1269873535165&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

SlideshowGadget
http://192.0.2.1:8080/cs1/Satellite?c=FW_CSGadget&cid=1269873535728&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

ThumbListGadget
http://192.0.2.1:8080/cs1/Satellite?c=FW_CSGadget&cid=1269873553620&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

g. Once you successfully register the sample gadgets, they are displayed as thumbnails on either the global gadget catalog or a site gadget catalog, depending on how you registered the gadgets. For example, see Figure 19–12.

Figure 19–12 Gadgets interface with a sample gadget thumbnail



For more information about sample gadgets and creating custom gadgets, see the part "Developing Gadgets" in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Part V

Installing the Community Blogs Module

Part V contains the following chapters:

20

Procedures for Installing the Community Blogs Module

This chapter provides an overview of the WebCenter Sites: Community Blogs module and instructions for installing the Community Blogs module on a content management site. It also provides information about the components that are added to the site once the Community Blogs module is installed.

This chapter contains the following sections:

20.1 Overview of the Community Blogs Module

Installing the Community Blogs module on a content management site supplies content providers with the necessary components to create pages dedicated to displaying blogs on a website.

The Community Blogs module components are:

- The Blog flex family
- A default SiteEntry asset
- Default blog templates
- Default CSElements
- Sample blog page assets

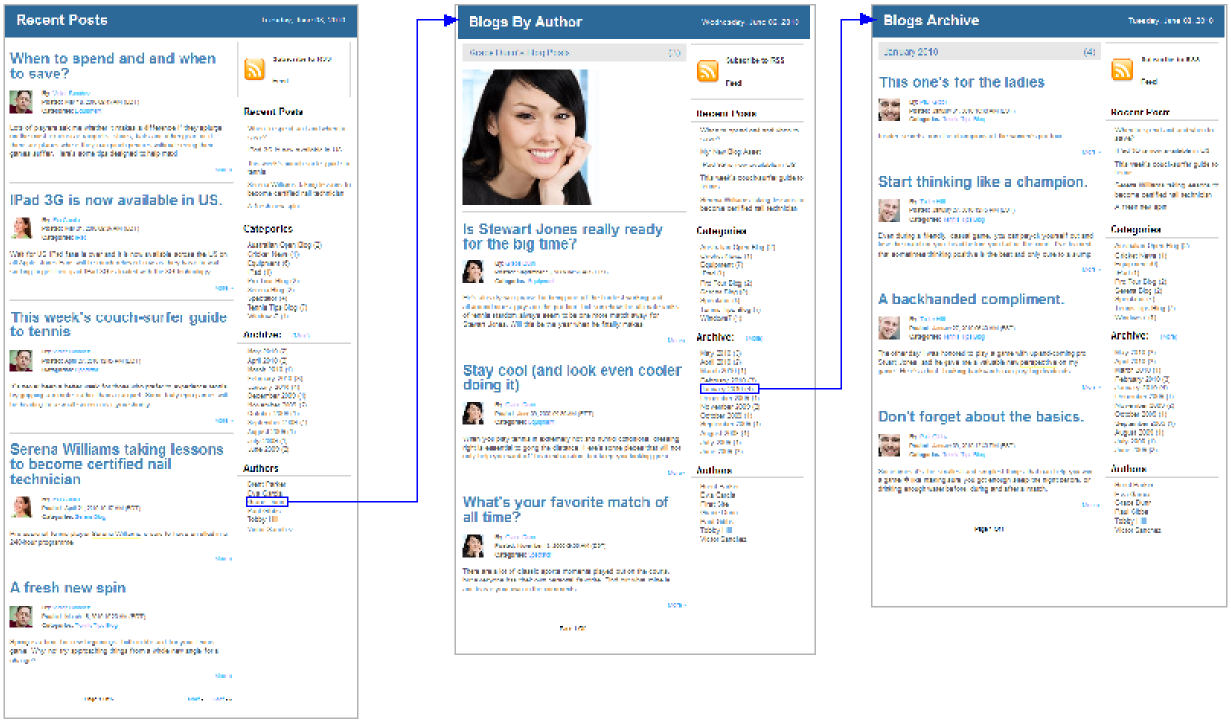
These components are pre-configured with default values and code which, out-of-the-box, render the sample blog Web pages shown in Figure 20–1.

Figure 20–1 Community Blogs Module's Sample Pages

The "Recent Posts" page is the sample blog home page. Clicking an author's name, in the "Authors" section of the right navigation panel, renders the "Blogs By Author" sample page.

The "Blogs By Author" sample page displays the blogs posted by a given author. Clicking a month, in the "Archive" section of the right navigation panel, renders the "Blogs Archive" page.

The "Blogs Archive" sample page displays the blogs posted for a given month.



Recent Posts

When to spend and and when to save?

IPad 3G is now available in US.

This week's couch-surfer guide to tennis

Serena Williams taking lessons to become certified nail technician

A fresh new spin

Blogs By Author

Is Stewart Jones really ready for the big time?

Stay cool (and look even cooler doing it)

What's your favorite match of all time?

Blogs Archive

This one's for the ladies

Start thinking like a champion.

A backhanded compliment.

Don't forget about the basics.

To add blog pages to your website, you can reuse and reconfigure the blog components to fit your requirements.

If you integrate the Community Blogs module with the WebCenter Sites: Community application, the Community application's comment fields are automatically inserted into the blog pages and displayed to site visitors. Commenting capability enables visitors to contribute feedback on the blogs, which in turn enables the content providers of your website to improve the content of the blogs they post.

20.2 Changes to WebCenter Sites During Community Blogs Module Installation

During the installation process, the blog installer adds the following components to your WebCenter Sites system:

- The **Blog** flex family was added to the CM site that you selected:
  - `FW_BlogAsset` (Blog Asset)
  - `FW_BlogParent` (Blog Category)
  - `FW_BlogAssetDef` (Blog Asset Definition)
  - `FW_BlogParentDef` (Blog Category Definition)
  - `FW_BlogAttribute` (Blog Attribute)
  - `FW_BlogFilter` (Blog Filter)

  - `FW_BlogAuthor` (Blog Author)
  - `FW_BlogAuthorParent` (Blog Author Category)
  - `FW_BlogAuthorDef` (Blog Author Definition)
  - `FW_BlogAuthorParentDef` (Blog Author Category Definition)
- Start menu items for the **Blog** flex family are automatically enabled on the site.

  > **Note:** A tree tab for the Community Blogs module must be manually created and enabled for the site. This tree tab will supply content providers with a way of reviewing the hierarchy of the blog assets and categories. For instructions about creating a tree tab, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

- The following blog related components and sample assets are added to your content management site:
  - Sample blog page assets
  - Default blog CSElements
  - Default blog templates
  - A default SiteEntry asset

  Out-of-the-box, you can publish the sample blog pages to a test website to view their default appearance (shown in Figure 20–1). You can also reuse and reconfigure the default blog components to fit your requirements. For more information about the components that compose the sample blog pages, see the chapter "Community Blogs: Sample Blog Pages" in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
- If you integrate the Community Blogs module with the Community application, the following properties are added to the `futuretense_xcel.ini` file:

Property	Description
`fwblogs.cos.enabled`	This property is set to `TRUE` if the Community Blogs module is integrated with the existing Community-Gadgets installation.
`cos.hostname`	This property points to the host name of your Community-Gadgets production system.
`cos.portnumber`	This property points to the port number of your Community-Gadgets production system.
`cos.protocol`	This property points to the protocol (`http` or `https`) of your Community-Gadgets production system.
`cos.contextroot`	This property points to the context root of your Community-Gadgets production system.

20.3 Prerequisites for Installing the Community Blogs Module

Before installing the Community Blogs module, ensure the following requirements are met:

- The Community Blogs module must be installed on a fully functional WebCenter Sites system running on either a content management or development system, and on the delivery system.

> **Note:** If you installed the Community Blogs module on the management WebCenter Sites system and then on the delivery WebCenter Sites system, ensure that the `futuretense_xcel.ini` file on the WebCenter Sites delivery system has the following parameters with the same values as on the management WebCenter Sites system:
>
> - `fwblogs.cos.enabled`
> - `cos.protocol`
> - `cos.hostname`
> - `cos.portnumber`
> - `cos.contextroot`

- If you wish to integrate the Community Blogs module with the Community-Gadgets application, then you must know the full URL (protocol, context root, host name, and port number) of the Community-Gadgets production system. For more information about the Community application's functionality, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

> **Note:** When integrated with the Community application, the Community Blogs Module uses the Oracle WebCenter Sites: Web Experience Management (WEM) Framework.

- Select or create a content management site on which to install the Community Blogs module. This site will be dedicated to containing the Community Blogs module components and sample assets. The purpose of this site is to help you review the sample assets, and to learn and experiment with the blog data model. **The same site must exist on the delivery system.**

20.4 Installing the Community Blogs Module on WebCenter Sites

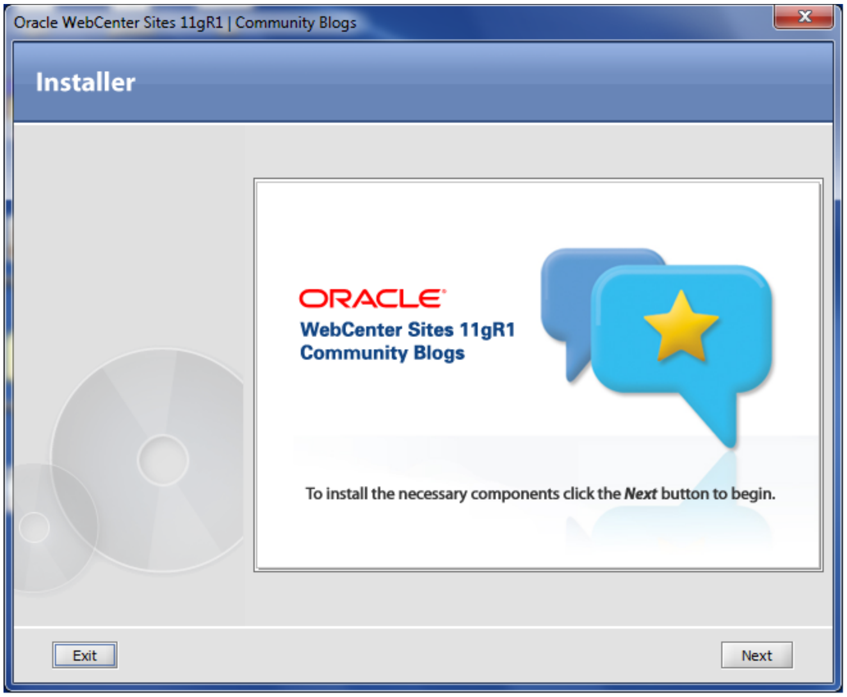
This section provides instructions for using the GUI installer to install the Community Blogs module on the content management site of your choice. You will run the installer on the content management or development system, and on the delivery system.

> **Note:** Ensure your WebCenter Sites installation is not running when you begin installing the Community Blogs module. If WebCenter Sites is running, the installation will fail.

To install the Community Blogs module on your WebCenter Sites system

1. Extract `WCS_Community_Blogs.zip` to a machine running WebCenter Sites.
2. Run the blog installer script:
   - Windows: `blogInstall.bat`
   - UNIX: `blogInstall.sh`
3. In the "Welcome" screen (Figure 20–2), click **Next**.

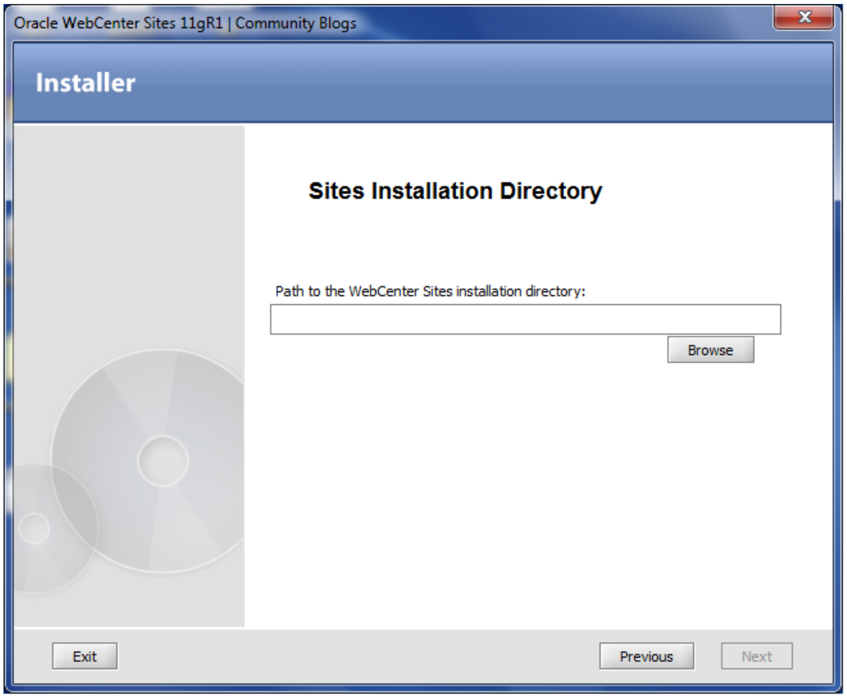
Figure 20–2 Welcome



To install the necessary components click the *Next* button to begin.

4. In the "Sites Installation Directory" screen, enter the path to your WebCenter Sites installation directory (Figure 20–3).

Figure 20–3 Sites Installation Directory

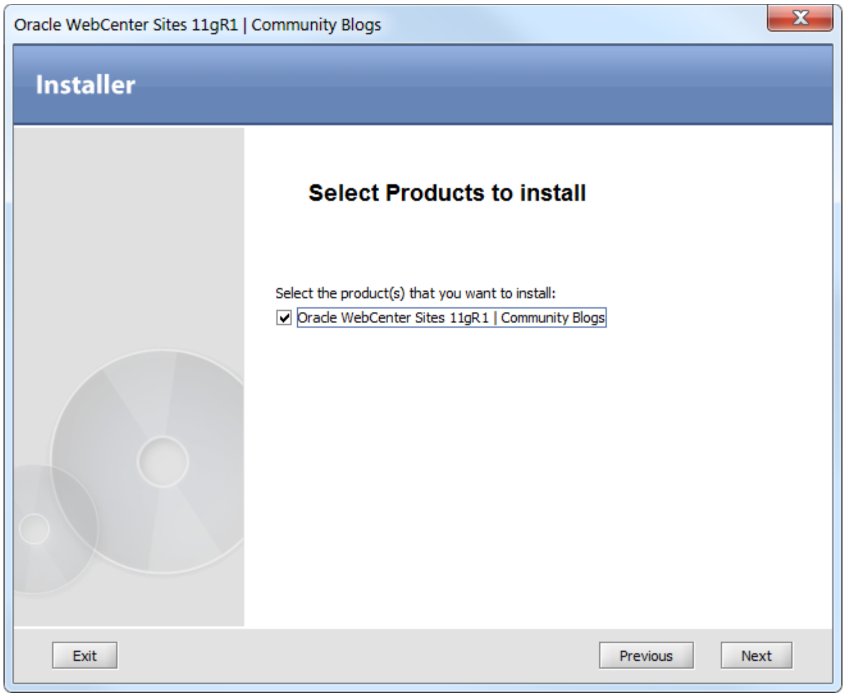


Sites Installation Directory

Path to the WebCenter Sites installation directory:

5. In the "Select Products to Install" screen, select the product(s) you wish to install (Figure 20–4).

Figure 20–4 Products to Install


Oracle WebCenter Sites 11gR1 | Community Blogs

Installer

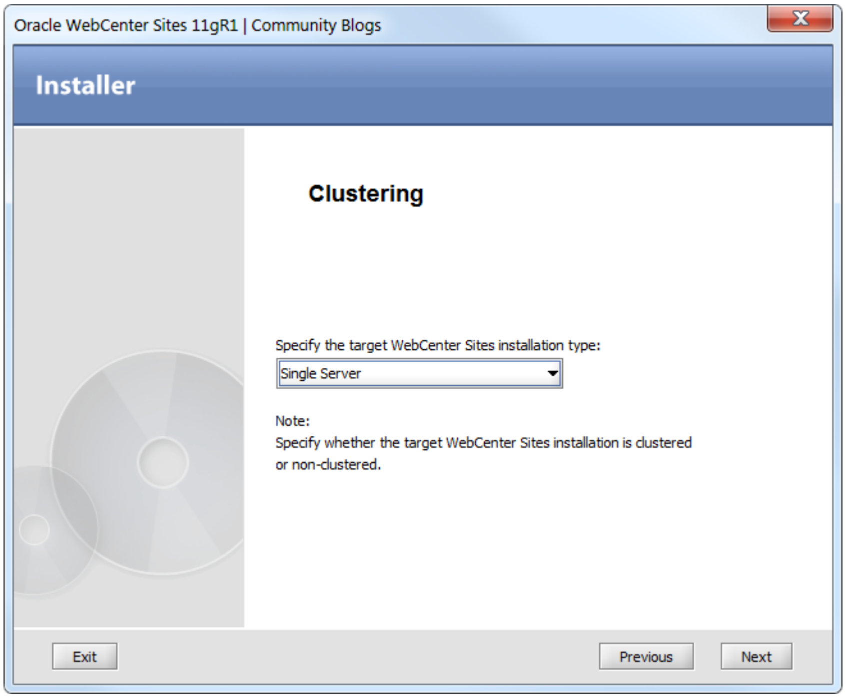
Select Products to install

Select the product(s) that you want to install:

☑ Oracle WebCenter Sites 11gR1 | Community Blogs

6. In the "Clustering" screen, specify whether you are installing the Community Blogs module on a single server or a cluster member (Figure 20–5).

Figure 20–5 Clustering


Oracle WebCenter Sites 11gR1 | Community Blogs

Installer

Clustering

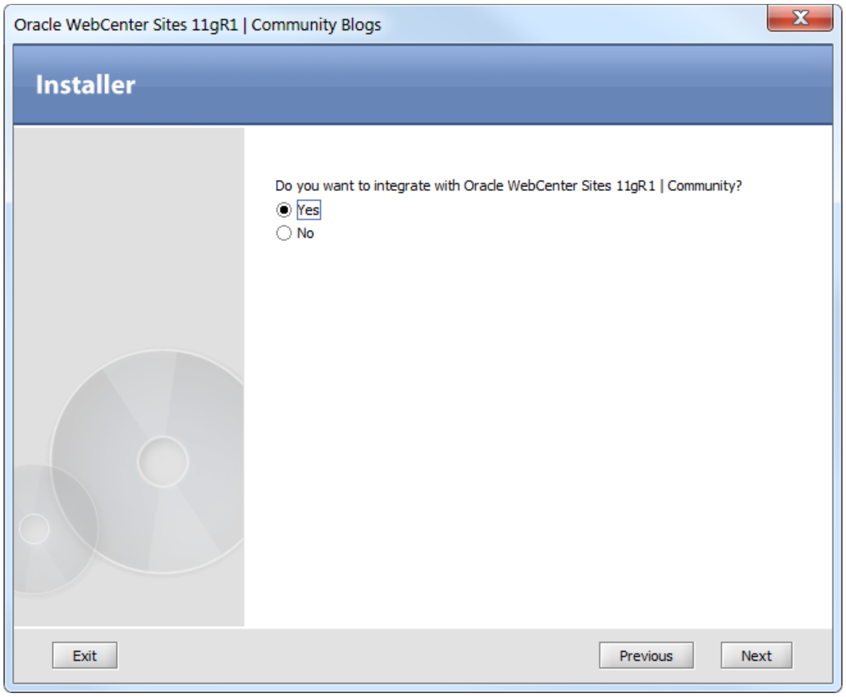
Specify the target WebCenter Sites installation type:

Single Server

Note:
Specify whether the target WebCenter Sites installation is clustered or non-clustered.

7. Select **Yes** if you wish to integrate the Community Blogs module with the Community application (Figure 20–6).

Figure 20–6 Integration of Community Blogs with Community Application



Oracle WebCenter Sites 11gR1 | Community Blogs

Installer

Do you want to integrate with Oracle WebCenter Sites 11gR1 | Community?

● Yes

○ No

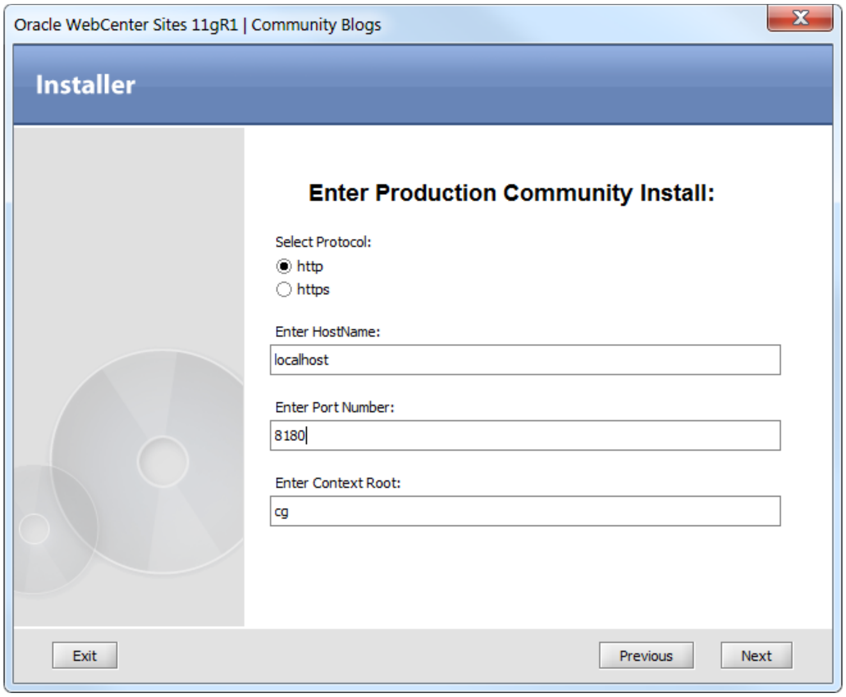
Note: Integrating the Community Blogs module with the Community application automatically adds the tags for the comments widget and comments link to the blog templates of your website.

---

8. If you chose to integrate the Community Blogs module with the Community application, then you must enter the following information about the Community application's production system (Figure 20–7):

- Host name or IP address of the machine running the Community application.
- Port number on which the Community application is running.

Figure 20–7 Production Information



Enter Production Community Install:

Select Protocol:

● http
○ https

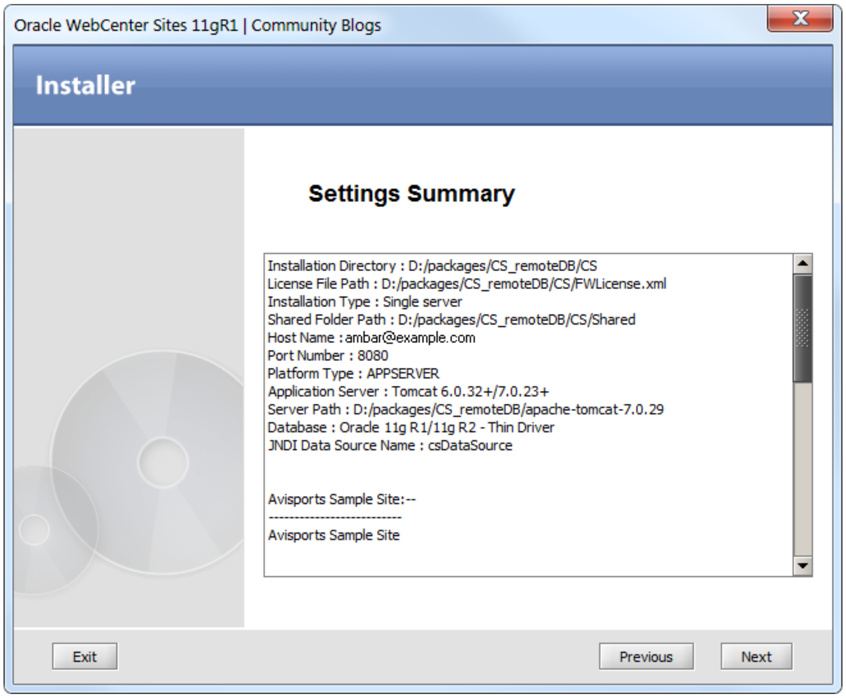
Enter HostName: localhost

Enter Port Number: 8180

Enter Context Root: cg

9. In the "Settings Summary" screen, review the environment on which you are installing the Community Blogs module (Figure 20–8). Click **Next**.

Figure 20–8 Settings Summary



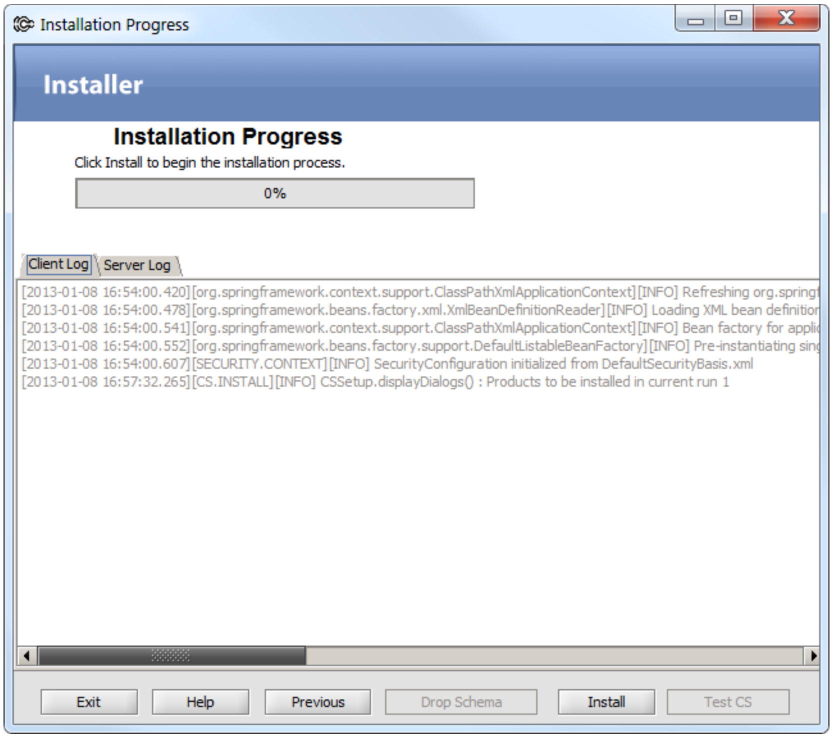
Settings Summary

```
Installation Directory : D:/packages/CS_remoteDB/CS
License File Path : D:/packages/CS_remoteDB/CS/FWLicense.xml
Installation Type : Single server
Shared Folder Path : D:/packages/CS_remoteDB/CS/Shared
Host Name : ambar@example.com
Port Number : 8080
Platform Type : APPSERVER
Application Server : Tomcat 6.0.32+/7.0.23+
Server Path : D:/packages/CS_remoteDB/apache-tomcat-7.0.29
Database : Oracle 11g R1/11g R2 - Thin Driver
JNDI Data Source Name : csDataSource


Avisports Sample Site:--
-------------------------
Avisports Sample Site
```

10. In the "Installation Progress" screen, click **Install** (Figure 20–9).

Figure 20–9 Installation Progress



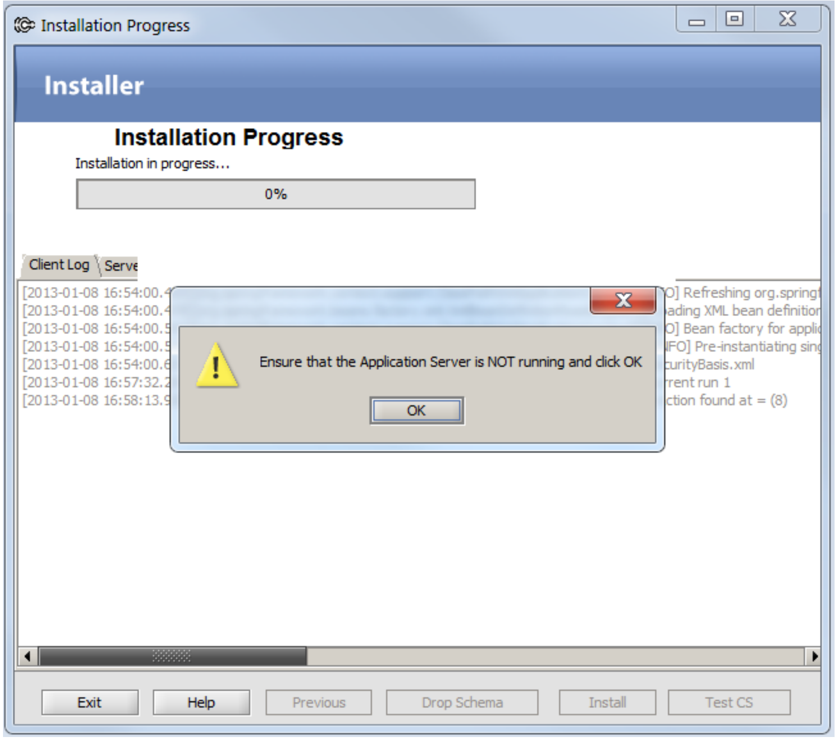
Installation Progress

Click Install to begin the installation process.

```
[2013-01-08 16:54:00.420][org.springframework.context.support.ClassPathXmlApplicationContext][INFO] Refreshing org.springf
[2013-01-08 16:54:00.478][org.springframework.beans.factory.xml.XmlBeanDefinitionReader][INFO] Loading XML bean definitior
[2013-01-08 16:54:00.541][org.springframework.context.support.ClassPathXmlApplicationContext][INFO] Bean factory for applic
[2013-01-08 16:54:00.552][org.springframework.beans.factory.support.DefaultListableBeanFactory][INFO] Pre-instantiating sing
[2013-01-08 16:54:00.607][SECURITY.CONTEXT][INFO] SecurityConfiguration initialized from DefaultSecurityBasis.xml
[2013-01-08 16:57:32.265][CS.INSTALL][INFO] CSSetup.displayDialogs() : Products to be installed in current run 1
```

11. Ensure that your application server is not running (Figure 20–10). Click **OK**.

Figure 20–10 Application Server Warning



Installation Progress

Installation in progress...

```
[2013-01-08 16:54:00.4                                                    O] Refreshing org.springf
[2013-01-08 16:54:00.4                                                    ading XML bean definitior
[2013-01-08 16:54:00.5                                                    O] Bean factory for applic
[2013-01-08 16:54:00.5                                                    IFO] Pre-instantiating sing
[2013-01-08 16:54:00.6                                                    curityBasis.xml
[2013-01-08 16:57:32.2                                                    rrent run 1
[2013-01-08 16:58:13.9                                                    ction found at = (8)
```

Ensure that the Application Server is NOT running and click OK

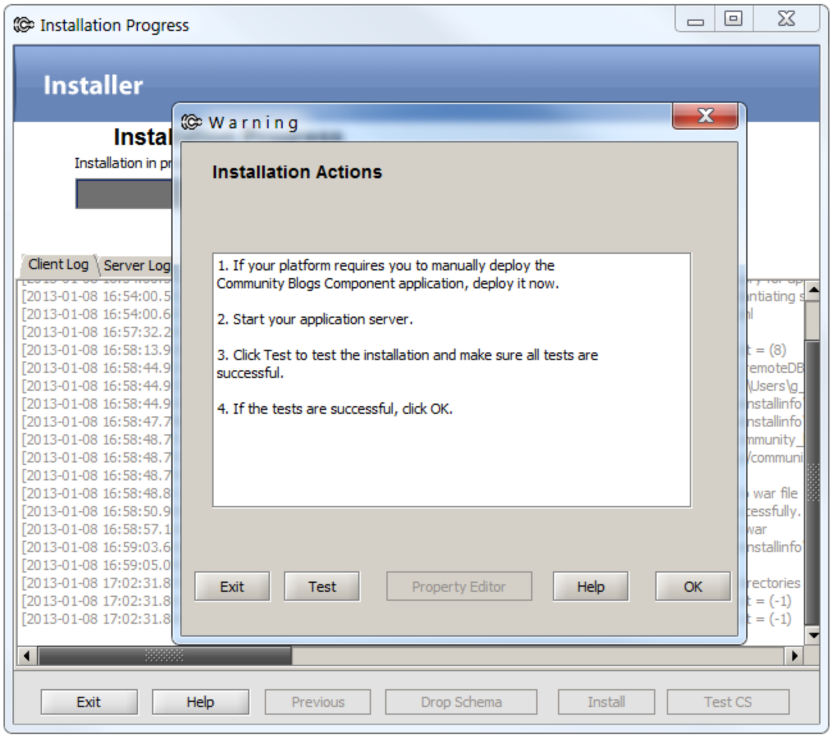
12. During the installation the "Installation Actions" dialog box opens (Figure 20–11).

a. Start your WebCenter Sites system.

b. Click **Test** to test the database connections between the Community Blogs module and WebCenter Sites.

c. Click **OK** to continue with the installation.

Figure 20–11 Installation Actions



1. If your platform requires you to manually deploy the Community Blogs Component application, deploy it now.

2. Start your application server.

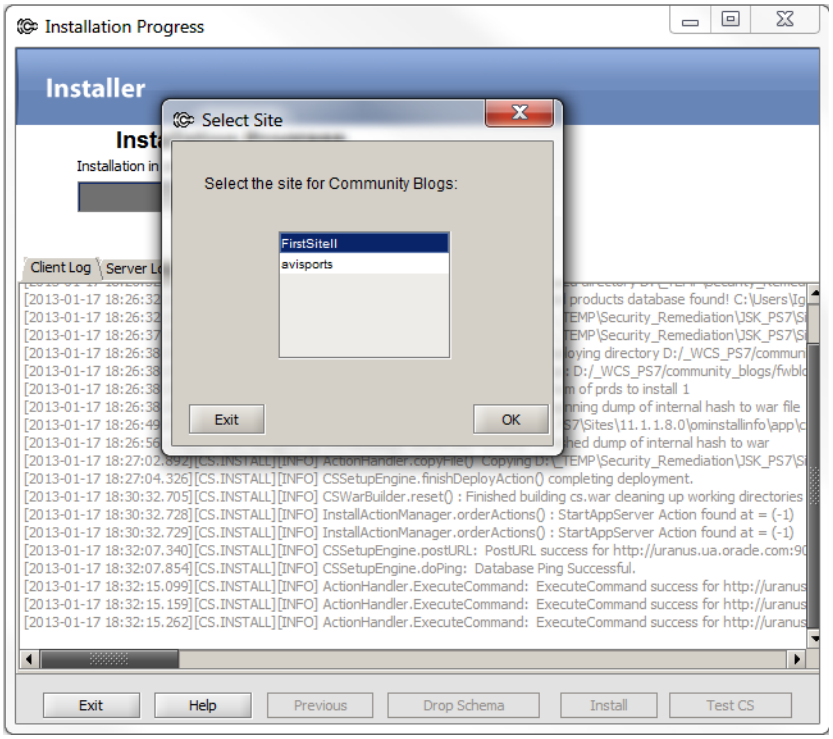
3. Click Test to test the installation and make sure all tests are successful.

4. If the tests are successful, click OK.

```
[2013-01-08 16:54:00.5
[2013-01-08 16:54:00.6
[2013-01-08 16:57:32.2
[2013-01-08 16:58:13.9
[2013-01-08 16:58:44.9
[2013-01-08 16:58:44.9
[2013-01-08 16:58:44.9
[2013-01-08 16:58:47.7
[2013-01-08 16:58:48.7
[2013-01-08 16:58:48.7
[2013-01-08 16:58:48.7
[2013-01-08 16:58:48.8
[2013-01-08 16:58:50.9
[2013-01-08 16:58:57.1
[2013-01-08 16:59:03.6
[2013-01-08 16:59:05.0
[2013-01-08 17:02:31.8
[2013-01-08 17:02:31.8
[2013-01-08 17:02:31.8
```

13. The "Select Site" screen opens (Figure 20–12), listing the content management sites on your WebCenter Sites system. Select the site on which you wish to install the Community Blogs module. Click **OK**.

Figure 20–12 Select Site



Select the site for Community Blogs:

FirstSiteII
avisports

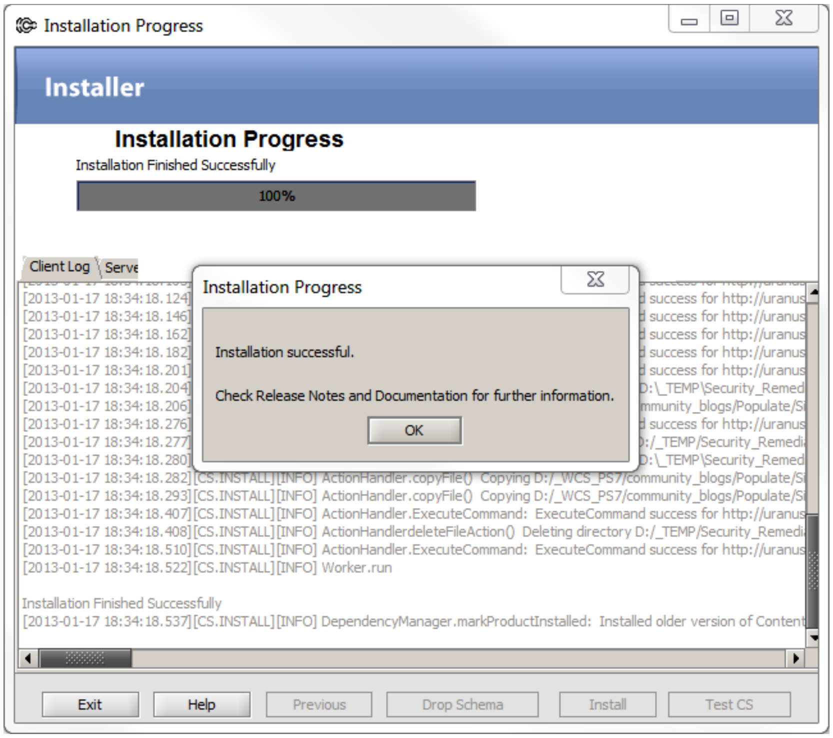
```
[2013-01-17 18:26:32                              products database found! C:\Users\Ig
[2013-01-17 18:26:32                              TEMP\Security_Remediation\JSK_PS7\Si
[2013-01-17 18:26:37                              TEMP\Security_Remediation\JSK_PS7\Si
[2013-01-17 18:26:38                              loying directory D:/_WCS_PS7/commun
[2013-01-17 18:26:38                              : D:/_WCS_PS7/community_blogs/fwblo
[2013-01-17 18:26:38                              m of prds to install 1
[2013-01-17 18:26:38                              nning dump of internal hash to war file
[2013-01-17 18:26:49                              S7\Sites\11.1.1.8.0\ominstallinfo\app\c
[2013-01-17 18:26:56                              hed dump of internal hash to war
[2013-01-17 18:27:02.892][CS.INSTALL][INFO] ActionHandler.copyFile() Copying D:\_TEMP\Security_Remediation\JSK_PS7\Si
[2013-01-17 18:27:04.326][CS.INSTALL][INFO] CSSetupEngine.finishDeployAction() completing deployment.
[2013-01-17 18:30:32.705][CS.INSTALL][INFO] CSWarBuilder.reset() : Finished building cs.war cleaning up working directories
[2013-01-17 18:30:32.728][CS.INSTALL][INFO] InstallActionManager.orderActions() : StartAppServer Action found at = (-1)
[2013-01-17 18:30:32.729][CS.INSTALL][INFO] InstallActionManager.orderActions() : StartAppServer Action found at = (-1)
[2013-01-17 18:32:07.340][CS.INSTALL][INFO] CSSetupEngine.postURL:  PostURL success for http://uranus.ua.oracle.com:90
[2013-01-17 18:32:07.854][CS.INSTALL][INFO] CSSetupEngine.doPing:  Database Ping Successful.
[2013-01-17 18:32:15.099][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for http://uranus
[2013-01-17 18:32:15.159][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for http://uranus
[2013-01-17 18:32:15.262][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for http://uranus
```

Note: If you run the Community Blogs installer more than once, the Community Blogs module will be automatically installed on the site that you selected the first time you ran the installer.

---

14. The installation is complete (Figure 20–13). Click **OK** to exit the installer. Stop, and then restart your WebCenter Sites system.

Figure 20–13 Installation Progress



Installation Progress

Installation Finished Successfully

100%

Installation Progress

Installation successful.

Check Release Notes and Documentation for further information.

OK

```
[2013-01-17 18:34:18.124]                                            d success for http://uranus
[2013-01-17 18:34:18.146]                                            d success for http://uranus
[2013-01-17 18:34:18.162]                                            d success for http://uranus
[2013-01-17 18:34:18.182]                                            d success for http://uranus
[2013-01-17 18:34:18.201]                                            d success for http://uranus
[2013-01-17 18:34:18.204]                                            D:\_TEMP\Security_Remed
[2013-01-17 18:34:18.206]                                            mmunity_blogs/Populate/Si
[2013-01-17 18:34:18.276]                                            d success for http://uranus
[2013-01-17 18:34:18.277]                                            D:/_TEMP/Security_Remedi
[2013-01-17 18:34:18.280]                                            D:\_TEMP\Security_Remed
[2013-01-17 18:34:18.282][CS.INSTALL][INFO] ActionHandler.copyFile()  Copying D:/_WCS_PS7/community_blogs/Populate/Si
[2013-01-17 18:34:18.293][CS.INSTALL][INFO] ActionHandler.copyFile()  Copying D:/_WCS_PS7/community_blogs/Populate/Si
[2013-01-17 18:34:18.407][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for http://uranus
[2013-01-17 18:34:18.408][CS.INSTALL][INFO] ActionHandlerdeleteFileAction()  Deleting directory D:/_TEMP/Security_Remedi
[2013-01-17 18:34:18.510][CS.INSTALL][INFO] ActionHandler.ExecuteCommand:  ExecuteCommand success for http://uranus
[2013-01-17 18:34:18.522][CS.INSTALL][INFO] Worker.run

Installation Finished Successfully
[2013-01-17 18:34:18.537][CS.INSTALL][INFO] DependencyManager.markProductInstalled:  Installed older version of Content
```

15. Verify that the installation was successful. See Section 20.5, "Verifying the Installation." If you need information about the blog components and sample assets that are added to your WebCenter Sites system, see Section 20.2, "Changes to WebCenter Sites During Community Blogs Module Installation."

20.5 Verifying the Installation

The installer added the **Blog** flex family to your content management site. This flex family provides the framework for creating blog web pages.

> **Note:** For WebSphere application server, redeploy the `ContentServer.ear` file after the Blogs application is installed.

To verify that the Blog flex family was added to the CM site

1. Log in to WebCenter Sites with the credentials of a general administrator (`fwadmin/xceladmin`, by default). Select the site on which you installed the Community Blogs module and then select the WebCenter Sites **Admin** icon.

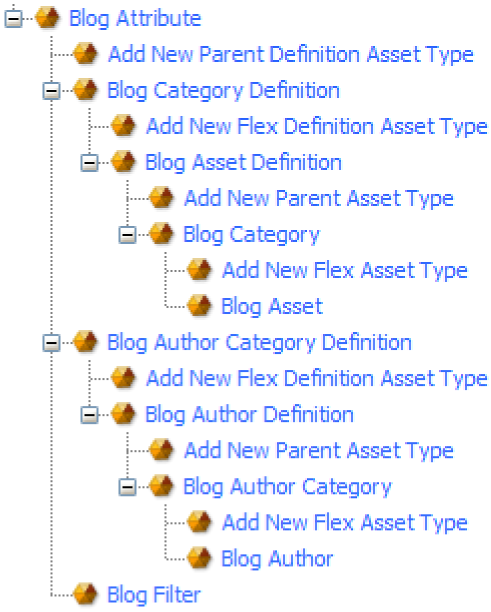
2. Navigate to the WebCenter Sites tree, and select the **Admin** tab.

   a. Expand the **Flex Family Maker** node.

      You should see the **Blog Attribute** asset type listed.

   b. Expand all of the nodes for the **Blog** flex family. The blog hierarchy looks as shown in Figure 20–14.

Figure 20–14 Blog Hierarchy



- Blog Attribute
  - Add New Parent Definition Asset Type
  - Blog Category Definition
    - Add New Flex Definition Asset Type
    - Blog Asset Definition
      - Add New Parent Asset Type
      - Blog Category
        - Add New Flex Asset Type
        - Blog Asset
  - Blog Author Category Definition
    - Add New Flex Definition Asset Type
    - Blog Author Definition
      - Add New Parent Asset Type
      - Blog Author Category
        - Add New Flex Asset Type
        - Blog Author
  - Blog Filter

If the **Blog** flex family hierarchy is rendered as shown above, then you have successfully installed the Community Blogs module. For more information about the blog components that are added to your WebCenter Sites installation, see Section 20.2, "Changes to WebCenter Sites During Community Blogs Module Installation."

Part VI

Installing Oracle WebCenter Sites: Site Capture

Part VI contains the following chapters:

21

Overview of WebCenter Sites: Site Capture

This chapter presents an overview of the Site Capture application and the installation process.

This chapter contains the following topics:

- Section 21.1, "About WebCenter Sites: Site Capture"
- Section 21.2, "Site Capture Installation Summary"
- Section 21.3, "Before You Begin"
- Section 21.4, "Next Step"

21.1 About WebCenter Sites: Site Capture

Oracle WebCenter Sites: Site Capture is a web application that integrates with Oracle WebCenter Sites through the Oracle WebCenter Sites: Web Experience Management (WEM) Framework to capture dynamically published websites for evaluation, compliance purposes, high availability requirements, and other types of scenarios.

Crawls can be initiated manually from the Site Capture interface, or they can be triggered by the completion of a WebCenter Sites RealTime publishing session. In each scenario, the crawler captures the site in one of the following modes, depending on the user's selections:

- Static mode: The site is stored as files that are ready to be served. Only the latest capture is kept.
- Archive mode: The site is stored in a zip file. A pointer in the Site Capture database enables archive management from the Site Capture interface.

21.2 Site Capture Installation Summary

This guide contains procedures for installing and configuring Site Capture to support:

- Static and archive capture initiated manually, from the Site Capture interface.
- Static and archive capture triggered by the completion of a WebCenter Sites RealTime publishing session. Setting up publishing-triggered site capture is an option.
- Administrative users and developers. The Site Capture application is designed for general administrators of the WebCenter Sites system on which the Site Capture application will be running. Developers will write advanced crawler configuration code – for example, code that triggers Site Capture to execute a post-crawl command such as copying statically captured sites to a web server's doc base.

21.3 Before You Begin

Users of this guide must have experience installing and configuring enterprise-level software, such as application servers and databases. Also required is a general administrator's experience with WebCenter Sites and the WEM Admin interface.

- To complete the procedures in this guide, you must be a WebCenter Sites general administrator who belongs to the `RESTAdmin` security group.
- Download the *Oracle WebCenter Sites 11g Release 1 (11.1.1.x) Certification Matrix* for information about supported operating systems, application servers, databases, and browsers. Also, download the Oracle WebCenter Sites release notes for information about Site Capture.
- Read this guide to acquire an understanding of the Site Capture installation process. The basic steps are configuring the Site Capture application server, running the installer to create the Site Capture `war` file, deploying Site Capture, completing post-installation and verification steps, and if necessary, setting up publishing-triggered site capture.
- On all systems, set the JAVA_HOME variable to point to a valid installation of certified version of JDK which is certified for use with WebCenter Sites as noted in the certification matrix.
- Prepare the Site Capture installation components:
  - Ensure you have a dedicated and fully functional Oracle WebCenter Sites installation running in development or content management mode. Site Capture must communicate with this WebCenter Sites system in order to run.
  - Ensure you have a dedicated application server on which to install the Site Capture application server and the application itself. During the installation process, you will configure Site Capture to communicate with the WebCenter Sites system (described above), which runs on its own host machine.
  - Decide whether your Site Capture application will be running as a single application or in clustered mode. For a diagram of a single-server installation, and clustered installation, see Figure 21–1.
    - Install or reuse the following components:

      Install a supported Site Capture application server on the dedicated application server. For clustered installations, install the application server for each Site Capture instance.

      For a clustered installation, install a load balancer on a host machine of your choice. **The Site Capture installation directory must be a shared directory, accessible to all cluster members**.

      To store archived sites, you can either reuse WebCenter Sites' database or install a dedicated, supported Site Capture database on a host machine of your choice.
- Decide whether you will be running publishing-triggered site capture. If so, you will need a WebCenter Sites source and target system:
  - The target system provides the REST API and WEM SSO API, which enable the target system to communicate with the Site Capture application at the end of the publishing session in order to start the required crawlers. The Site Capture application will then send crawler invocation status to the target WebCenter Sites system, which will, in turn, send the same information to the

source WebCenter Sites system. Both WebCenter Sites systems record the status information in their own log files (`futuretense.txt`, by default).

  – You will integrate the source and target WebCenter Sites systems with your Site Capture installation. For some of the possible configurations, see Figure 21–2, "Single-Server Installation Enabled for Publishing-Triggered Site Capture" and Figure 21–3, "Clustered Installation Enabled for Publishing-Triggered Site Capture".

- Decide whether to install sample crawlers (**recommended**). For more information about the sample crawlers, see the note in step 1 in Section 23.1, "Installation Steps."
- You have the option to install Site Capture silently or graphically. The silent installer provides help and sample values for every piece of information that needs to be set.

Figure 21–1 Single-Server and Cluster Installation

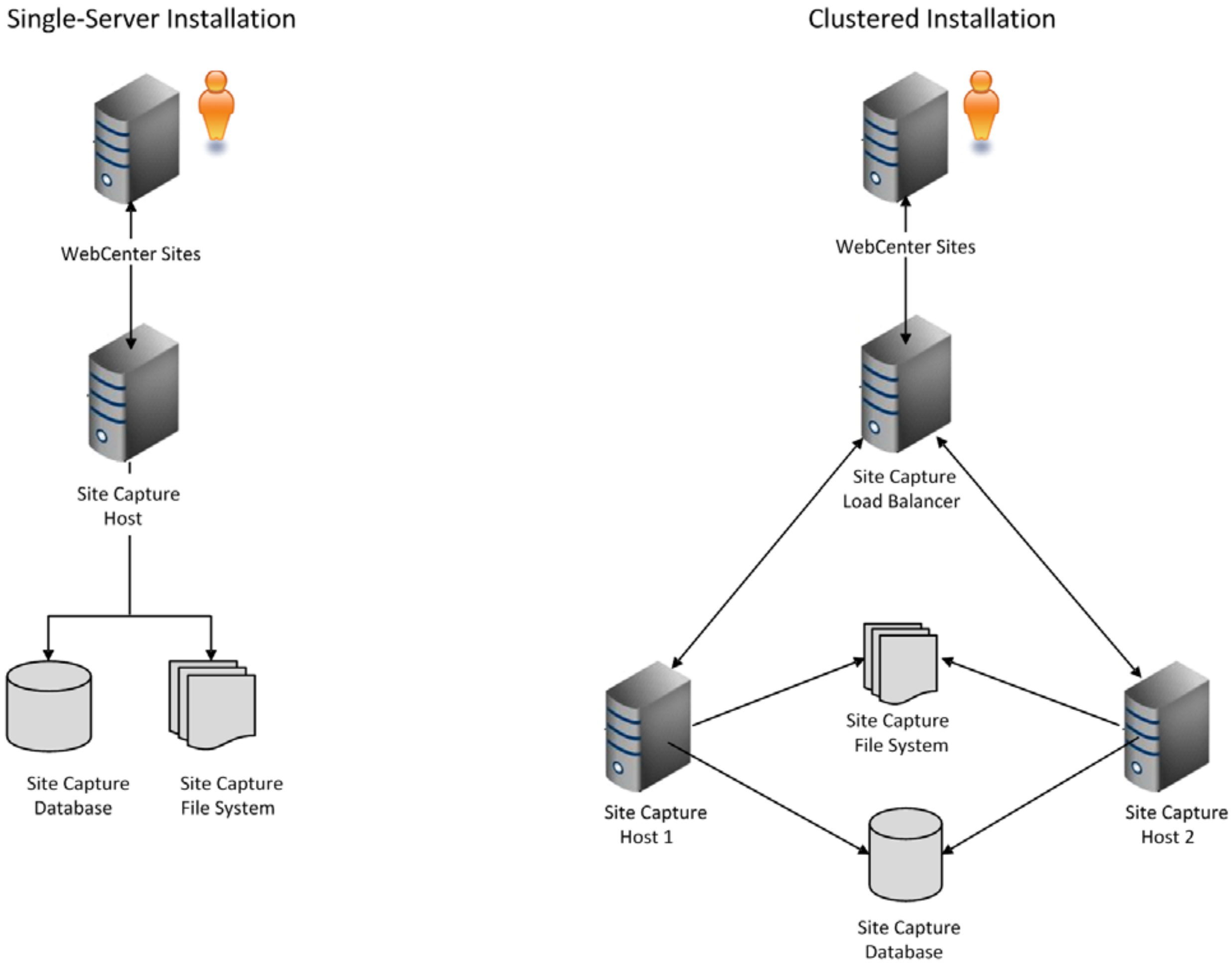
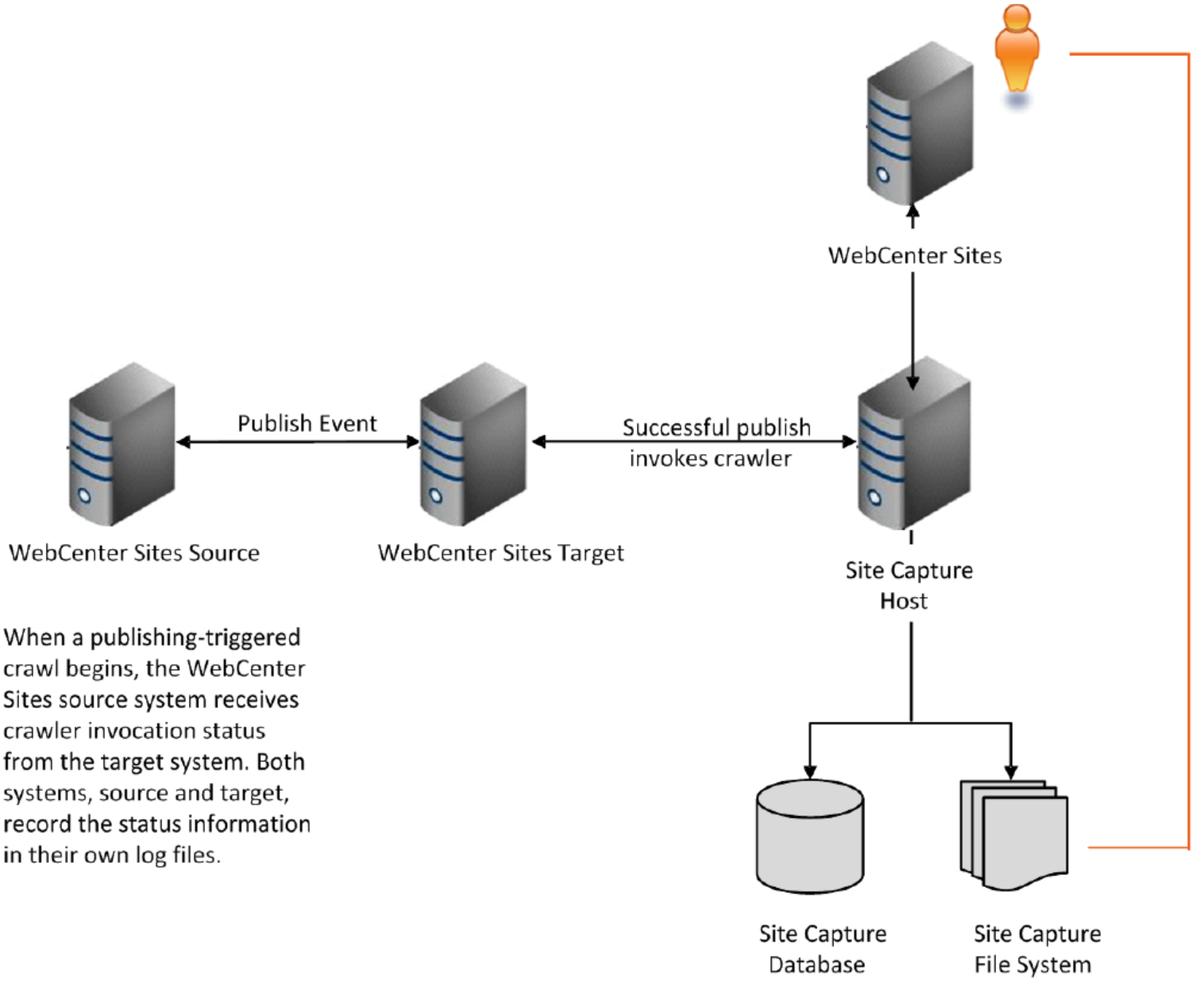
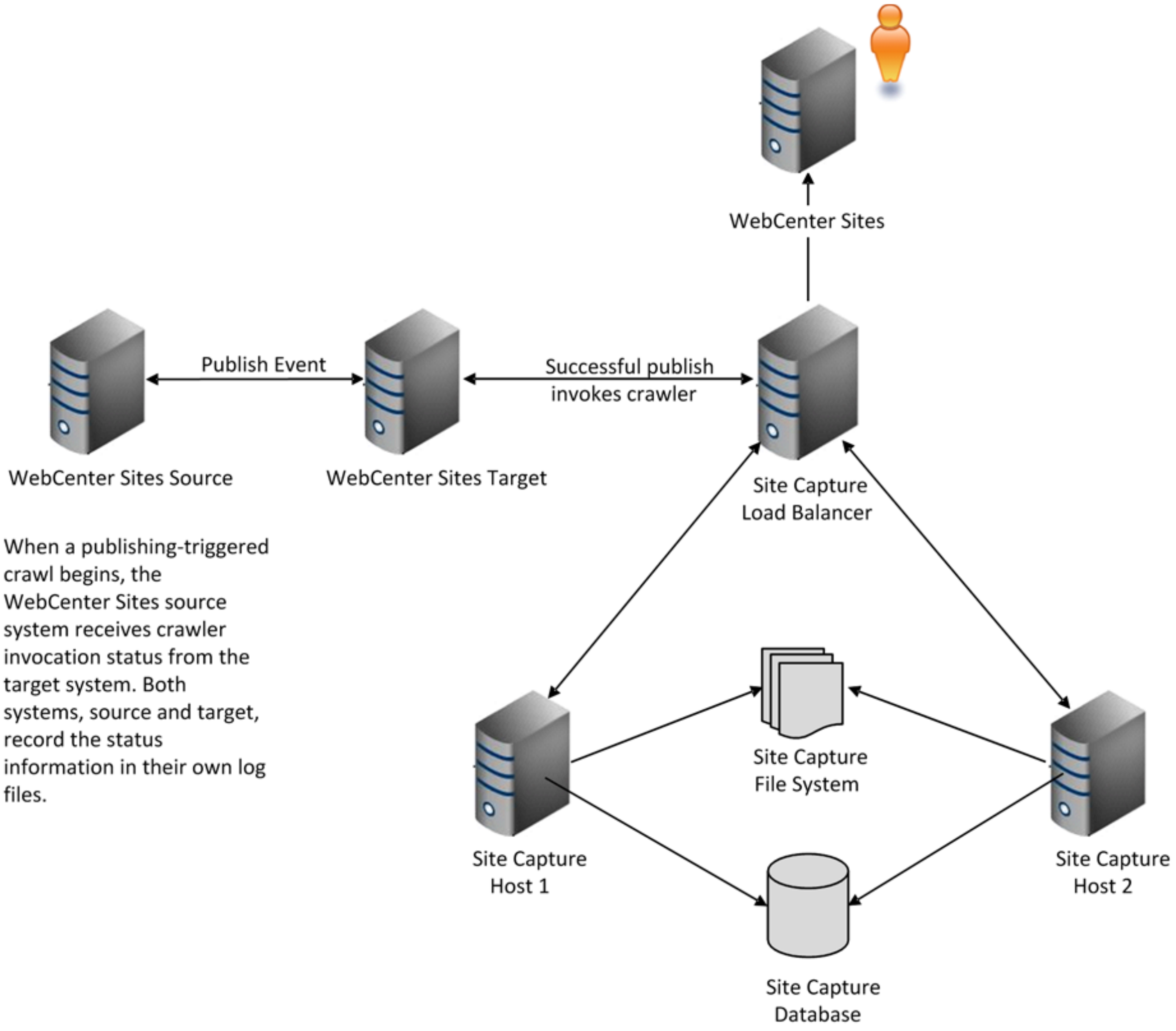


Figure 21–2 Single-Server Installation Enabled for Publishing-Triggered Site Capture



When a publishing-triggered crawl begins, the WebCenter Sites source system receives crawler invocation status from the target system. Both systems, source and target, record the status information in their own log files.

Figure 21–3 Clustered Installation Enabled for Publishing-Triggered Site Capture


WebCenter Sites
Publish Event
Successful publish invokes crawler
WebCenter Sites Source
WebCenter Sites Target
Site Capture Load Balancer
When a publishing-triggered crawl begins, the WebCenter Sites source system receives crawler invocation status from the target system. Both systems, source and target, record the status information in their own log files.
Site Capture File System
Site Capture Host 1
Site Capture Host 2
Site Capture Database

21.4 Next Step

Continue to Chapter 22, "Configuring the Site Capture Application Server."

22

Configuring the Site Capture Application Server

Site Capture can run as a single application or in a clustered installation.

This chapter contains the following topics:

- Section 22.1, "Configuration Options"
- Section 22.2, "Configuring Tomcat Application Server"
- Section 22.3, "Configuring WebLogic Application Server"
- Section 22.4, "Configuring WebSphere Application Server"

22.1 Configuration Options

In this chapter, you will configure an application server to support Site Capture running as a single instance or in cluster mode.

> **Note :** Before proceeding with the configuration steps, note that the Site Capture application must run on context root / to support previewing of archives. (Site Capture serves pages from within archives, which can have any context root.)

Complete one of the following procedures:

- Section 22.2, "Configuring Tomcat Application Server"
- Section 22.3, "Configuring WebLogic Application Server"
- Section 22.4, "Configuring WebSphere Application Server"

22.2 Configuring Tomcat Application Server

If you are creating a clustered installation, complete the following steps on all cluster members.

To configure Tomcat

1. Open `server.xml` in `<apache-tomcat-6.0.29 directory>\conf` and do the following:

   a. Replace port `8080` with the Site Capture port (on which the application server is listening):

Note: For clustered installations, specify a unique port number for each cluster member.

```
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" />
```

b. Configure the data source for Site Capture.

Note: For clustered installations, configure the same data source on each cluster member.

Refer to Table 22–1 for information about configuring the driver class name and URL for your selected driver.

Below is a sample DB2 configuration:

```
<Context path="" docBase="ROOT" debug="5" reloadable="true"
crossContext="true">
   <Resource name="scDataSource"
       auth="Container"
       type="javax.sql.DataSource"
       url="jdbc:db2://<hostname>:<port>/CLUSTDB"
       driverClassName="com.ibm.db2.jcc.DB2Driver"
       username="xxxxx"
       password="xxxxxxx"
       maxActive="20"
       maxIdle="10"
       maxWait="-1" />
</Context>
```

Table 22–1 Database Driver Parameters

Database Driver	Parameter	Value
SQL Server	DriverClass	`net.sourceforge.jtds.jdbcx.JtdsDataSource`
SQL Server	Required `.jar` files	`jtds-1.2.jar`
SQL Server	URL	`jdbc:jtds:sqlserver://<server>:<dbport>/<dbname>` **For example:** `jdbc:jtds:sqlserver://192.0.2.1:1433/SC70`
DB2	DriverClass	`com.ibm.db2.jcc.DB2Driver`
DB2	Required `.jar` files	`db2jcc.jar, db2cc_license_cu.jar`
DB2	URL	`jdbc:db2://<hostname>:<dbport>/<dbname>` **For example:** `jdbc:db2://192.0.2.1:5000/SC70`
Oracle	DriverClass	`oracle.jdbc.driver.OracleDriver`
Oracle	Required `.jar` files	`Ojdbc6.jar`

Table 22–1 (Cont.) Database Driver Parameters

Database Driver	Parameter	Value
Oracle	URL	`jdbc:oracle:thin:@//<hostname>:1521/<dbname>` **For example:** `jdbc:oracle:thin:@//godzilla.fatwire.com:1521/SC70`

2. Copy the required database driver jar file to `<apache-tomcat-6.0.29 directory>\lib`.
3. Once you have completed configuring the application server(s), continue to Chapter 23, "Procedures for Installing Site Capture."

22.3 Configuring WebLogic Application Server

If you are creating a clustered installation, complete the following steps on all cluster members.

To configure WebLogic

1. Create a domain in WebLogic and configure the administrator user name and password for the domain.
2. Start the WebLogic Admin Server from the command prompt:
   - Windows: `startweblogic.cmd`
   - UNIX: `./startweblogic.sh`
3. Log in to the console.
   a. Create a managed server `<managed_server_name>` in the domain you just created.

      Note: For clustered installations, specify a unique port number for each cluster member.

   b. Create a data source and map it to the managed server `<managed_ server_ name>`.

      Note: For clustered installations, configure the same data source on each cluster member.

4. Once you have completed configuring the application server(s), continue to Chapter 23, "Procedures for Installing Site Capture."

22.4 Configuring WebSphere Application Server

If you are creating a clustered installation, complete the following steps on all cluster members.

To configure WebSphere

1. Create a WebSphere application server instance.

> **Note:** For clustered installations, specify a unique port number for each cluster member.

2. Create a data source for the newly created server instance.

   > **Note:** For clustered installations, configure the same data source on each cluster member.

   a. Copy the required database driver into `<websphere_install_dir>/websphere/AppServer/universalDriver/lib`. (refer to Table 22–1 for the required driver).

   b. Log in to the WebSphere console and do the following:

      Add the user name and password of the database by creating an alias (go to **Security**, select **Global security**, then **Java Authentication and Authorization Service**, and then select **J2C authentication data**).

      Create a JDBC provider and data source (go to **Resources**, and then select **JDBC**).

3. Once you have completed configuring the application server(s), continue to Chapter 23, "Procedures for Installing Site Capture."

23 Procedures for Installing Site Capture

Site Capture can be installed graphically or silently. You will first create the war file using the installer, then manually deploy the war file and test the installation.

This chapter contains the following topics:

23.1 Installation Steps

To install Site Capture, you will complete the following basic steps:

1. Run the silent or GUI installer to create the Site Capture war file (ROOT.war).

> **Note:** Keep in mind the following about clustered installations and installing the sample crawlers:
>
> - If you are creating a clustered installation:
>
>   Run the installer once. You will deploy the ROOT.war file (and ROOT folder) on all cluster members, as shown in this chapter.
>
>   The Site Capture installation directory must be a shared directory, accessible to all other cluster members.
>
> - We recommend installing the sample crawlers. Procedures for quickly testing and using the Site Capture installation are based on the Sample crawler, described below.
>
>   **Sample** is a basic crawler that can be easily configured to capture any dynamic site. The configuration step amounts to setting the crawler's start URI in the crawler's configuration file.
>
>   **FirstSiteII** is a crawler with advanced configuration code for capturing the WebCenter Sites FirstSiteII sample website. The code demonstrates the implementation of various methods and interfaces that are used to control the crawler's site capture process, in this example, downloading a dynamic site (FirstSiteII) as a static site.
>
>   Crawlers can be easily deleted should you no longer need them.

2. If WebCenter Sites is running with Oracle Access Manager (OAM), integrate OAM with Site Capture.
3. Deploy the Site Capture application.

Detailed steps are provided in the rest of this section.

23.1.1 Running the Silent Installer

> **Note:** If you wish to run the GUI installer instead of the silent installer, skip to Section 23.1.2, "Running the GUI Installer."

1. Unzip the `sitecapture.zip` file into the desired location on the server.
2. Configure the `omii.ini` file (located at the root level in the extracted folder) according to instructions provided in the file, and save the file.

   The host name and port number of both single and clustered installation must be accessible externally.

   > **Reminder:** We recommend installing the sample crawlers. For information about the crawlers, see the note in step 1 in Section 23.1, "Installation Steps."

3. Open the `Install.ini` file (located at the root level in the extracted folder), add the line `loadfile=omii.ini` to the first section, and save the file.
4. Open a command prompt and run one of the following commands:
   - Windows: `scInstall.bat -silent`
   - Linux: `scInstall.sh -silent`

   You will see the following message in the command window after installation is successful – `"Installation Finished Successfully"`
5. Wait until the installer process exits normally and shows the command prompt again. The `ROOT` folder and the `ROOT.war` file will be created in the `webapps` folder of the installation directory specified in the `omii.ini` file.
6. Do one of the following:
   - If WebCenter Sites is running with Oracle Access Manager (OAM), integrate the Site Capture application with OAM. For instructions, see the section "Adjusting the root-context.xml File" in *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.
   - If WebCenter Sites is running with CAS, continue to Section 23.1.3, "Deploying Site Capture."

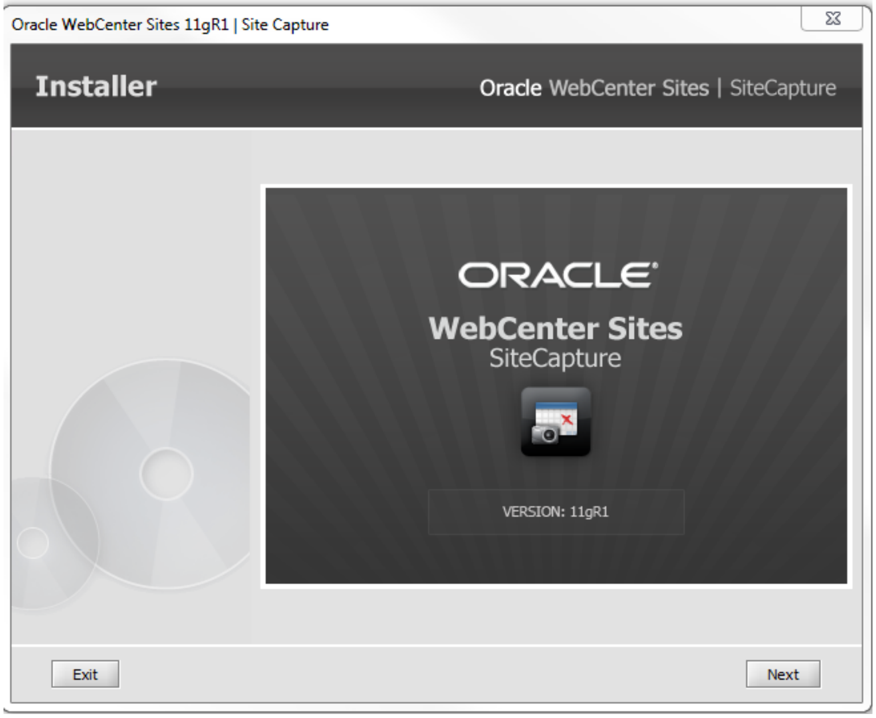
23.1.2 Running the GUI Installer

> **Note:** If you wish to run the silent installer instead of the GUI installer, go to Section 23.1.1, "Running the Silent Installer."

1. Download `sitecapture.zip` to a directory on your Site Capture server and extract it into a temporary directory. The extracted folder contains the `scInstall.bat` and `scInstall.sh` files, which are needed to run the Site Capture installer.
2. Execute the following installer script:
   - On Windows: `scInstall.bat`
   - On UNIX: `scInstall.sh`

   The installer "Welcome" screen is displayed (Figure 23–1).

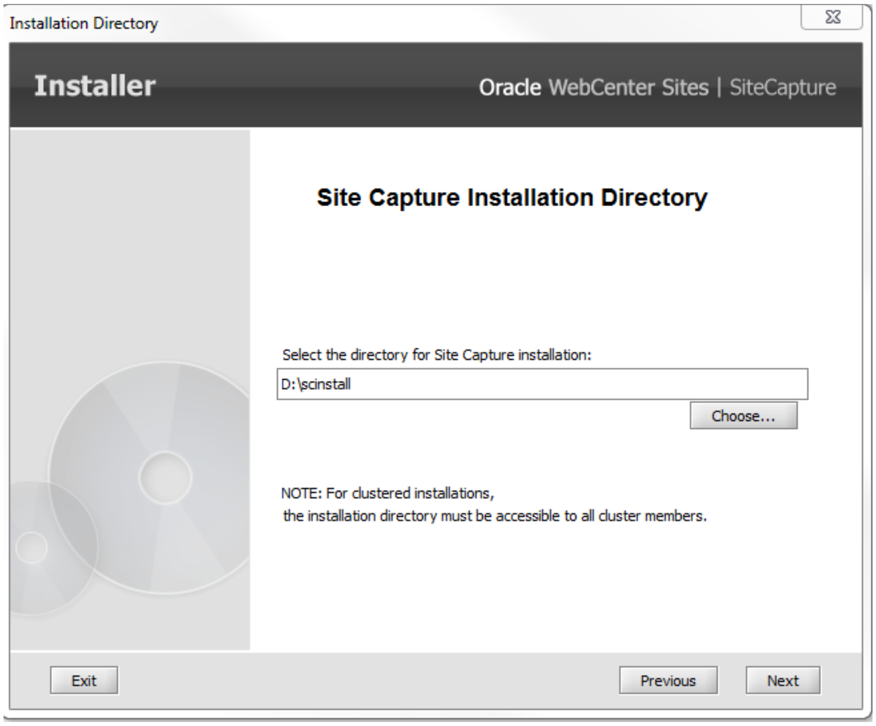
Figure 23–1 Welcome



Oracle WebCenter Sites 11gR1 | Site Capture
Installer
Oracle WebCenter Sites | SiteCapture
ORACLE
WebCenter Sites
SiteCapture
VERSION: 11gR1
Exit
Next

3. Specify the path to the directory where Site Capture will be installed (Figure 23–2) and click **Next**.

Figure 23–2 Site Capture Installation Directory



Installation Directory

Site Capture Installation Directory

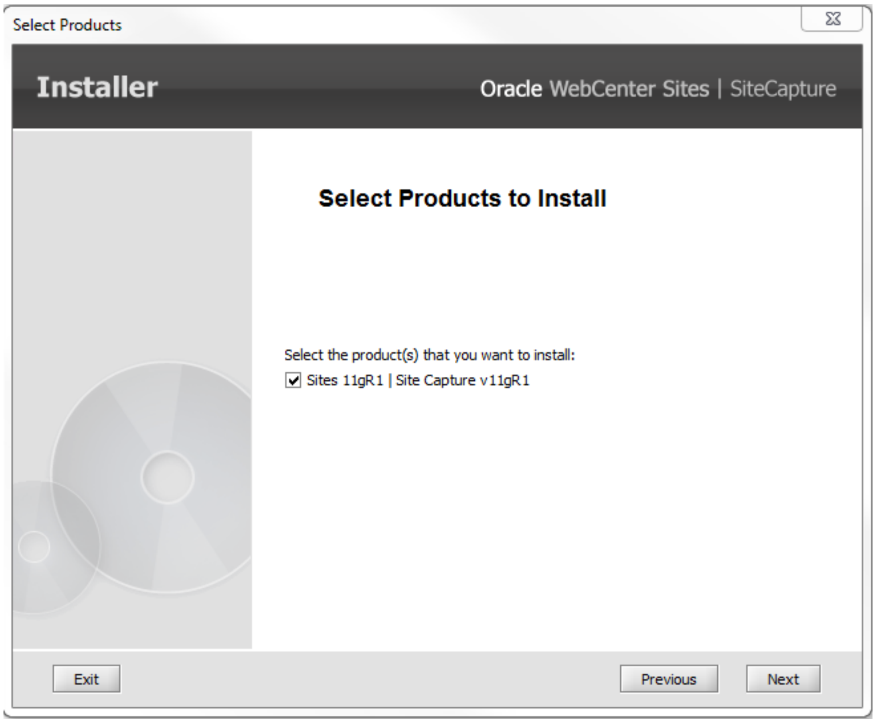
Select the directory for Site Capture installation:

D:\scinstall

NOTE: For clustered installations,
the installation directory must be accessible to all cluster members.

4. Select the product (Site Capture) to be installed (Figure 23–3) and click **Next**.

Figure 23–3 Select Products to Install



Select Products

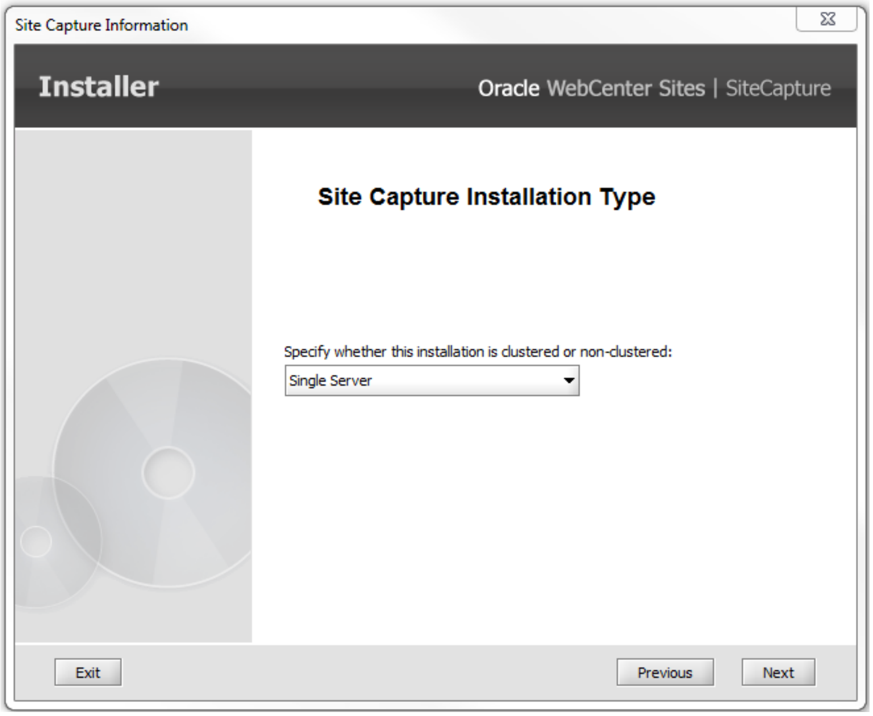
Select Products to Install

Select the product(s) that you want to install:

☑ Sites 11gR1 | Site Capture v11gR1

5. Select either **Single Server** or **Cluster** (Figure 23–4), and click **Next**.

Figure 23–4 Site Capture Installation Type



Site Capture Installation Type

Specify whether this installation is clustered or non-clustered: Single Server

6. Enter information about the Site Capture host machine (Figure 23–5). The host name and port number of both single and clustered installation must be accessible externally.

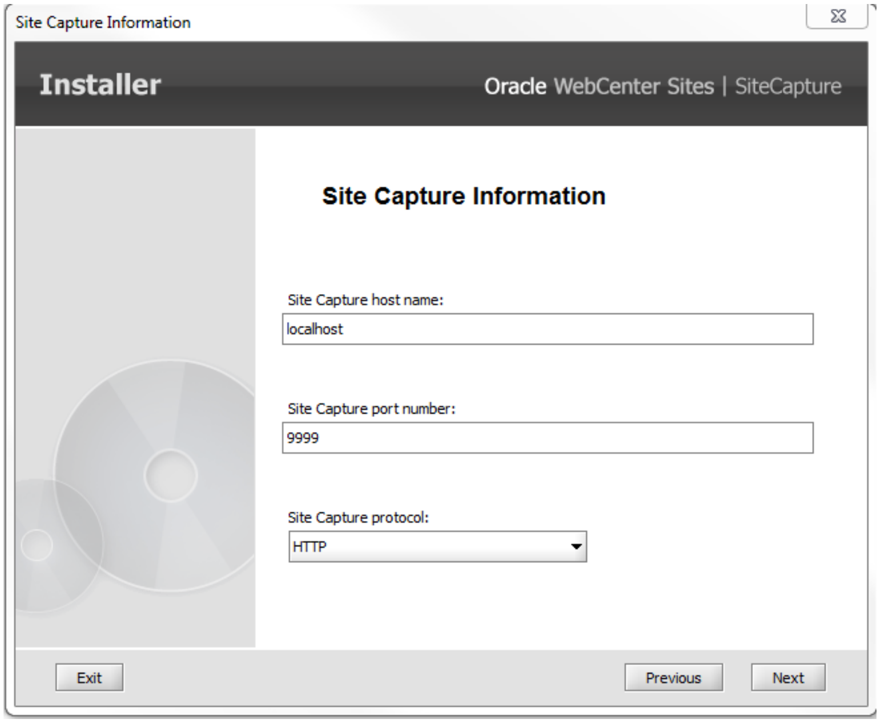
   – Single-server installation:

     – Enter the host name (or IP address) of the machine running the Site Capture application server.

     – Enter the port number of the Site Capture application server (the same number that you specified in the application server configuration steps).

     – Specify the protocol to be `HTTP` or `HTTPS`.

Figure 23–5 Site Capture Information: Single Server



Site Capture Information

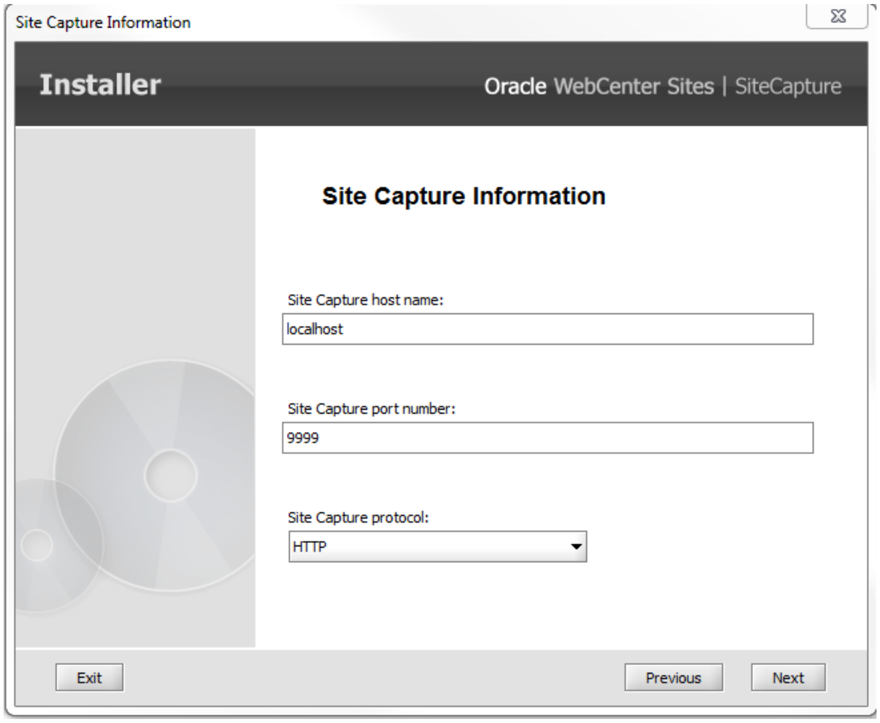
Site Capture host name: localhost

Site Capture port number: 9999

Site Capture protocol: HTTP

- Clustered installation (Figure 23–6):
  - Enter the host name (or IP address) of the machine running the load balancer.
  - Enter the port number of the load balancer.
  - Specify the protocol to be `HTTP` or `HTTPS`.

Figure 23–6 Site Capture Information: Clustered Installation



Site Capture Information

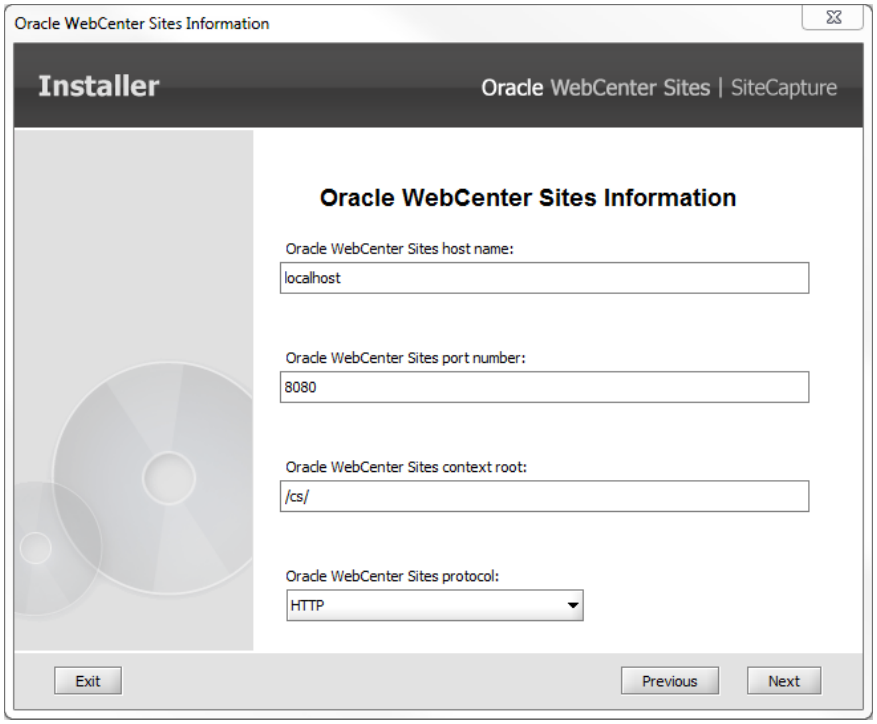
Site Capture host name: localhost

Site Capture port number: 9999

Site Capture protocol: HTTP

7. Enter information about the WebCenter Sites system on which Site Capture will run as an application (Figure 23–7):
   - WebCenter Sites host name or IP address
   - Port on which WebCenter Sites is listening
   - Protocol of the WebCenter Sites application server (`HTTP` or `HTTPS`)
   - Context root of the WebCenter Sites application

Figure 23–7 Oracle WebCenter Sites Information



Oracle WebCenter Sites Information

Oracle WebCenter Sites Information

Oracle WebCenter Sites host name: localhost

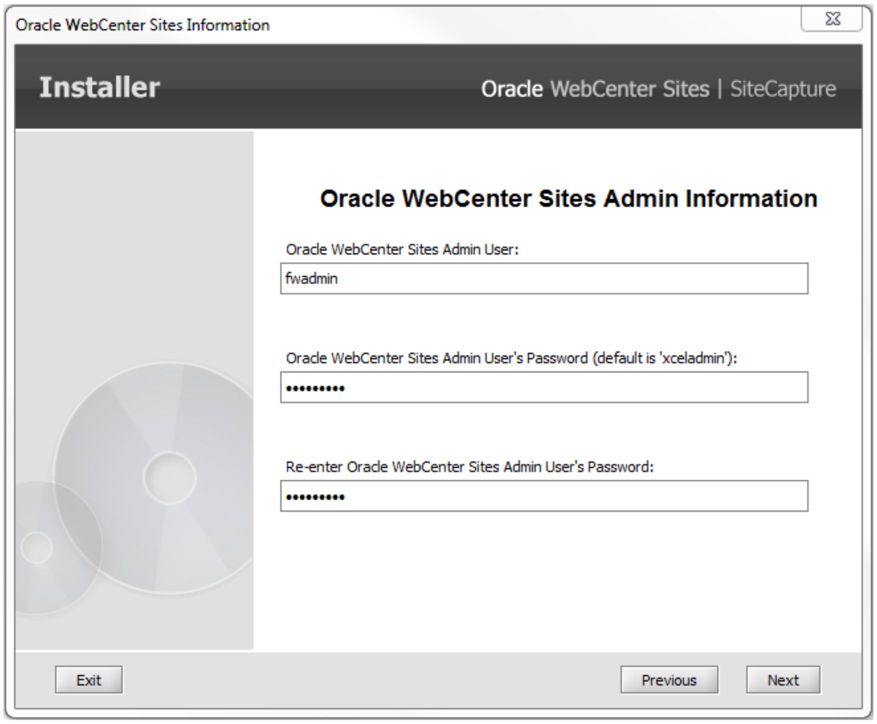
Oracle WebCenter Sites port number: 8080

Oracle WebCenter Sites context root: /cs/

Oracle WebCenter Sites protocol: HTTP

8. Enter the WebCenter Sites administrator's credentials (Figure 23–8):
   - Current user name of the WebCenter Sites general administrator
   - Current password

Figure 23–8 Oracle WebCenter Sites Admin Information



Oracle WebCenter Sites Information

Oracle WebCenter Sites Admin Information

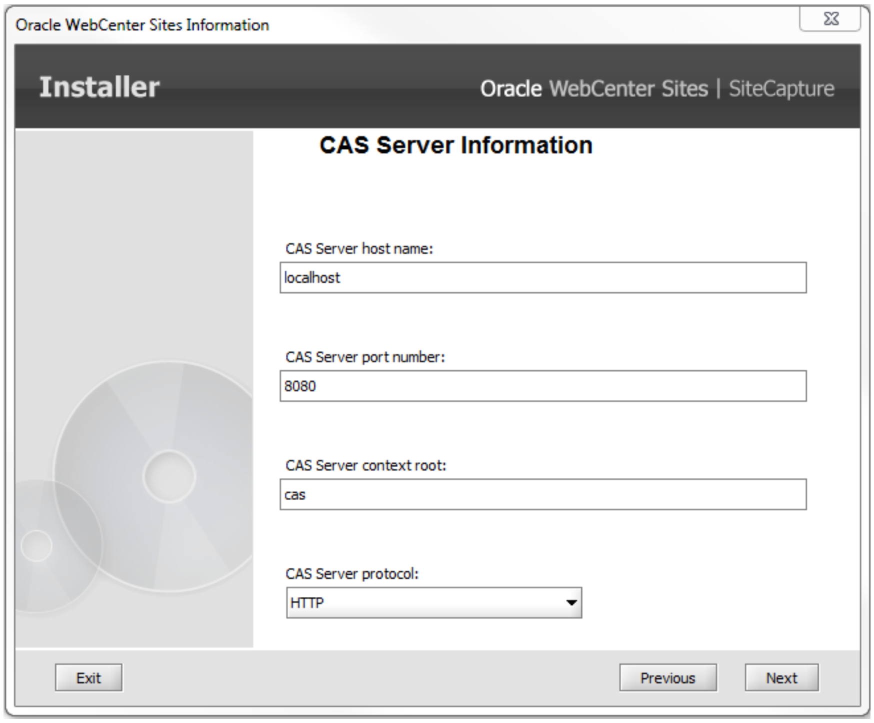
Oracle WebCenter Sites Admin User: fwadmin

Oracle WebCenter Sites Admin User's Password (default is 'xceladmin'): •••••••••

Re-enter Oracle WebCenter Sites Admin User's Password: •••••••••

9. Enter information about the CAS application (Figure 23–9):
   - CAS host name (or IP address)
   - Port number
   - Protocol of the CAS application server (`HTTP` or `HTTPS`)
   - CAS context root

Figure 23–9 CAS Server Information



Oracle WebCenter Sites Information

CAS Server Information

CAS Server host name: localhost

CAS Server port number: 8080

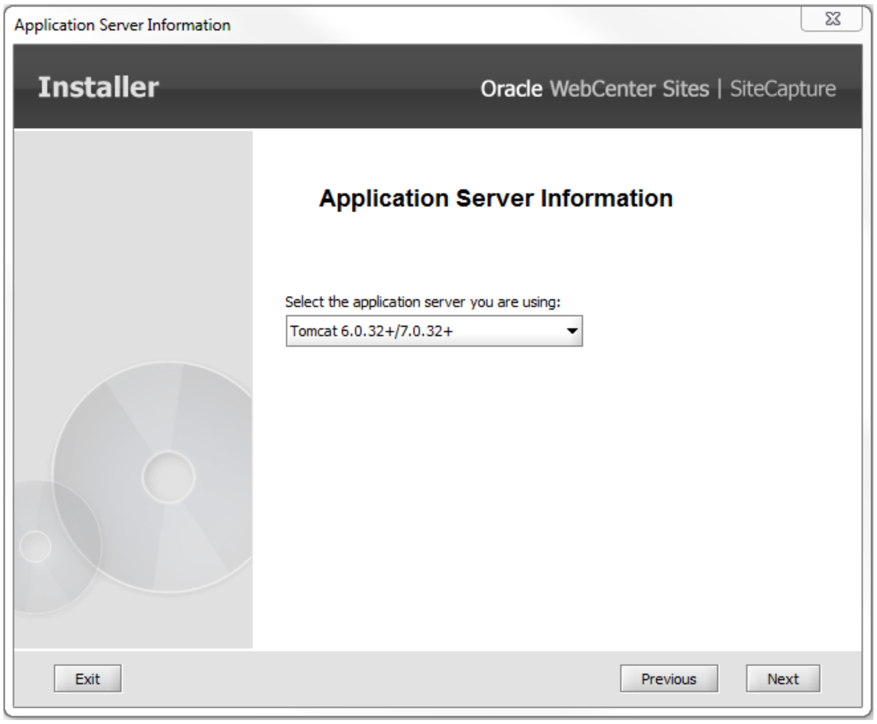
CAS Server context root: cas

CAS Server protocol: HTTP

10. Select the application server on which Site Capture will be installed (Figure 23–10).

> **Note:** The installer will create a `ROOT.war` file, which you will deploy on the application server.

Figure 23–10 Application Server Information



Application Server Information

Installer

Oracle WebCenter Sites | SiteCapture

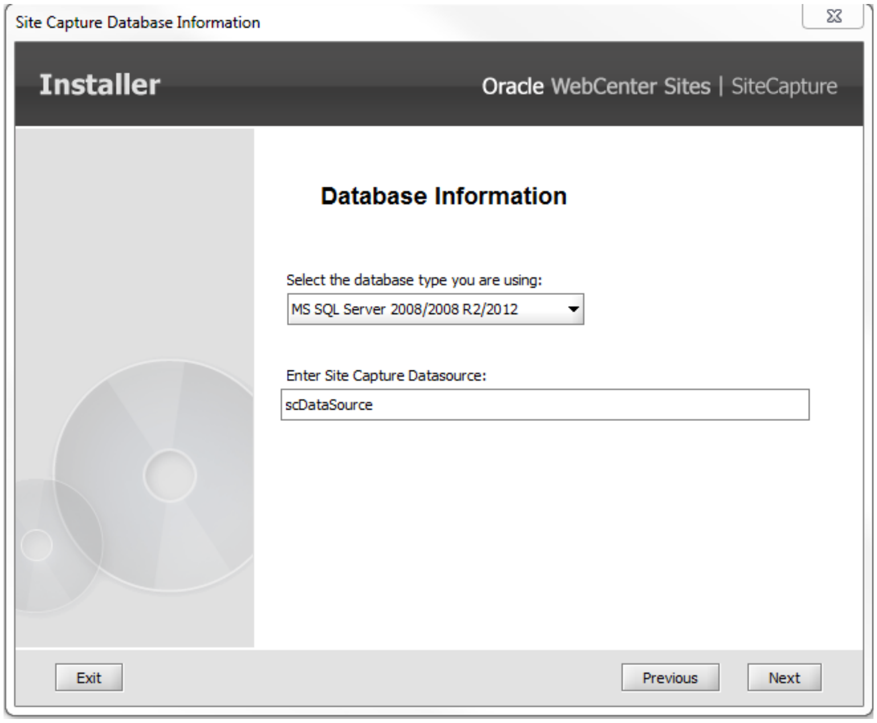
Application Server Information

Select the application server you are using:

Tomcat 6.0.32+/7.0.32+

11. Enter database information (Figure 23–11):
    - Select the database to which Site Capture will connect (select either Site Capture's database or WebCenter Sites' database).
    - Specify the name of Site Capture's data source exactly as it is configured in the application server.

Figure 23–11 Database Information


Site Capture Database Information
Installer
Oracle WebCenter Sites | SiteCapture
Exit
Previous
Next

Database Information

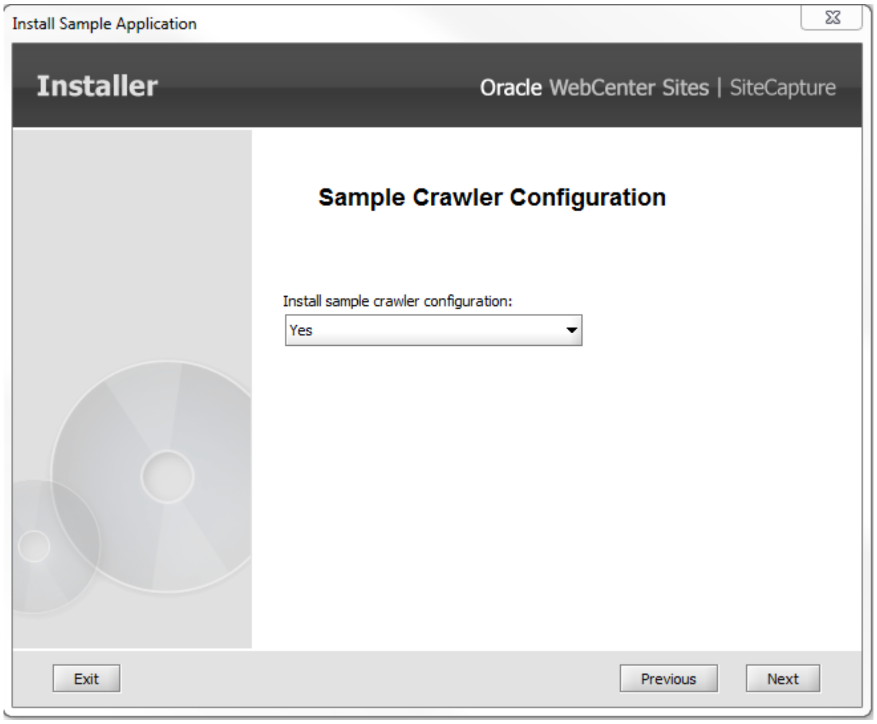
Select the database type you are using: MS SQL Server 2008/2008 R2/2012

Enter Site Capture Datasource: scDataSource

12. Select whether to install the sample crawlers (Figure 23–12):

> **Note:** We recommend installing the sample crawlers. For information about the crawlers, see the note in step 1 in Section 23.1, "Installation Steps."

Figure 23–12 Sample Crawler Configuration

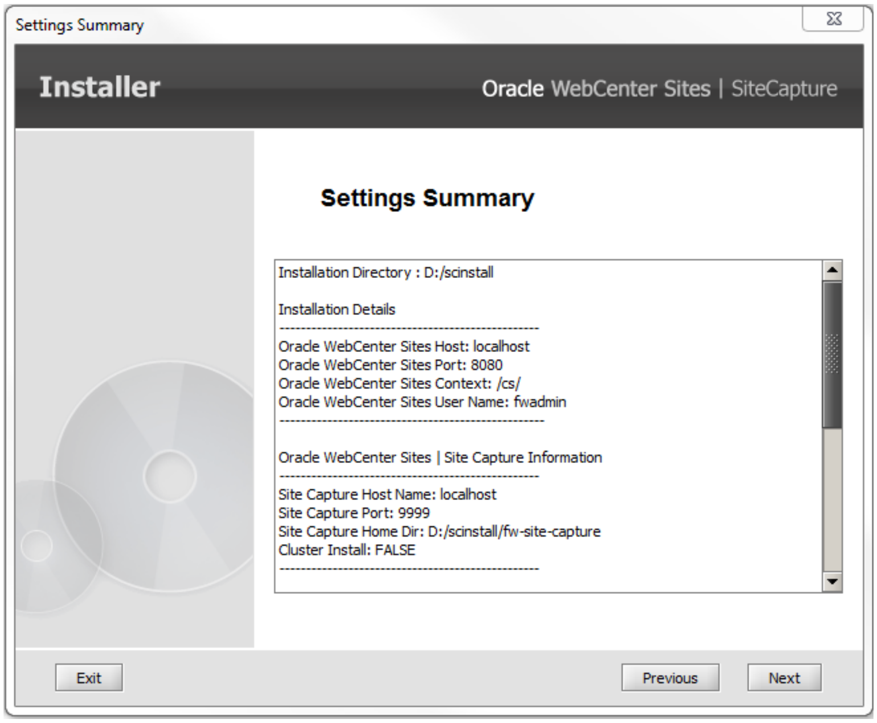

Install Sample Application
Installer
Oracle WebCenter Sites | SiteCapture
Exit
Previous
Next

Sample Crawler Configuration

Install sample crawler configuration: Yes

13. Review the settings summary to verify the data you have entered (Figure 23–13). To edit your settings, click **Previous** to return to the relevant screens.

Figure 23–13 Settings Summary



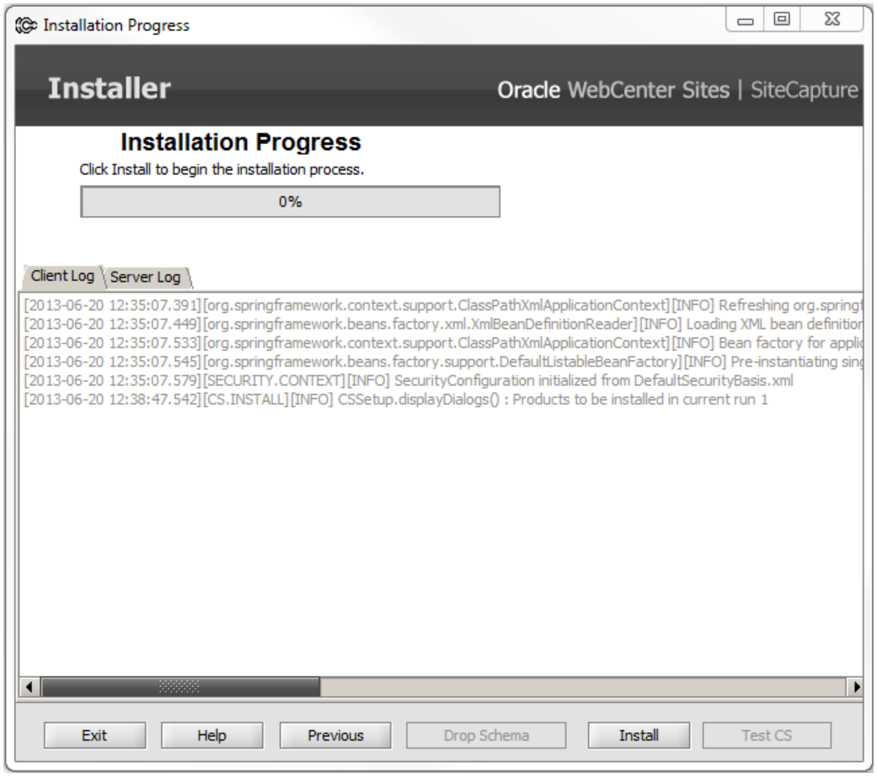
```
Installation Directory : D:/scinstall

Installation Details
------------------------------------------------
Oracle WebCenter Sites Host: localhost
Oracle WebCenter Sites Port: 8080
Oracle WebCenter Sites Context: /cs/
Oracle WebCenter Sites User Name: fwadmin
-------------------------------------------------

Oracle WebCenter Sites | Site Capture Information
------------------------------------------------
Site Capture Host Name: localhost
Site Capture Port: 9999
Site Capture Home Dir: D:/scinstall/fw-site-capture
Cluster Install: FALSE
------------------------------------------------
```

14. Once you have confirmed your settings (Figure 23–14), click **Install**.

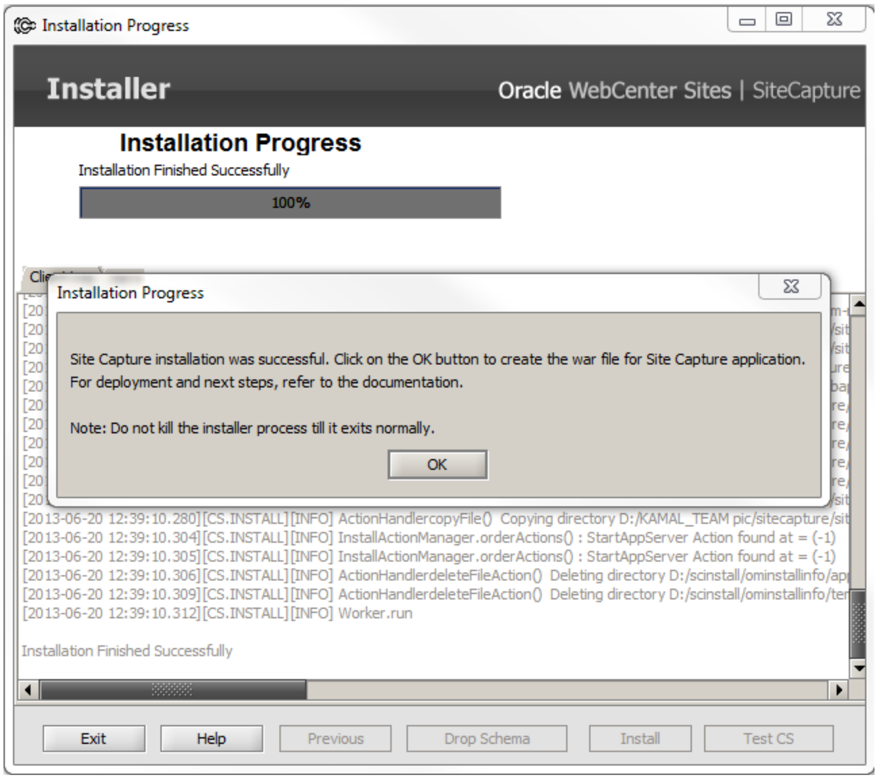
Figure 23–14 Installation Progress



```
[2013-06-20 12:35:07.391][org.springframework.context.support.ClassPathXmlApplicationContext][INFO] Refreshing org.springf
[2013-06-20 12:35:07.449][org.springframework.beans.factory.xml.XmlBeanDefinitionReader][INFO] Loading XML bean definitio
[2013-06-20 12:35:07.533][org.springframework.context.support.ClassPathXmlApplicationContext][INFO] Bean factory for applic
[2013-06-20 12:35:07.545][org.springframework.beans.factory.support.DefaultListableBeanFactory][INFO] Pre-instantiating sing
[2013-06-20 12:35:07.579][SECURITY.CONTEXT][INFO] SecurityConfiguration initialized from DefaultSecurityBasis.xml
[2013-06-20 12:38:47.542][CS.INSTALL][INFO] CSSetup.displayDialogs() : Products to be installed in current run 1
```

When the installation process successfully ends (Figure 23–15), the installer displays the following message: `"Site Capture war file was successfully created"`.

Figure 23–15 Site Capture Installation Successful



Installation Progress

Installation Finished Successfully

Installation Progress

Site Capture installation was successful. Click on the OK button to create the war file for Site Capture application.
For deployment and next steps, refer to the documentation.

Note: Do not kill the installer process till it exits normally.

```
[2013-06-20 12:39:10.280][CS.INSTALL][INFO] ActionHandlercopyFile()  Copying directory D:/KAMAL_TEAM pic/sitecapture/sit
[2013-06-20 12:39:10.304][CS.INSTALL][INFO] InstallActionManager.orderActions() : StartAppServer Action found at = (-1)
[2013-06-20 12:39:10.305][CS.INSTALL][INFO] InstallActionManager.orderActions() : StartAppServer Action found at = (-1)
[2013-06-20 12:39:10.306][CS.INSTALL][INFO] ActionHandlerdeleteFileAction()  Deleting directory D:/scinstall/ominstallinfo/ap
[2013-06-20 12:39:10.309][CS.INSTALL][INFO] ActionHandlerdeleteFileAction()  Deleting directory D:/scinstall/ominstallinfo/ter
[2013-06-20 12:39:10.312][CS.INSTALL][INFO] Worker.run

Installation Finished Successfully
```

The Site Capture file system was also created. The `war` file, called `ROOT.war`, is located in the `<SC_INSTALL_DIR>/fw-site-capture/webapps` folder. For information about the Site Capture file system, see the chapter, "Site Capture File System" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

15. Do one of the following:

- If WebCenter Sites is running with Oracle Access Manager (OAM), integrate the Site Capture application with OAM. For instructions, see the section "Adjusting the root-context.xml File" in *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.
- If WebCenter Sites is running with CAS, continue to Section 23.1.3, "Deploying Site Capture."

23.1.3 Deploying Site Capture

Note: Consider the following:

- For clustered installations, complete the deployment steps on each cluster member.
- The Site Capture application must have a context root of `/`.

This section provides instructions for deploying the Site Capture `ROOT.war` file. For instructions, see one of the following sections:

- Section 23.1.3.1, "Deploying on Tomcat Application Server"
- Section 23.1.3.2, "Deploying on WebLogic Application Server"

- Section 23.1.3.3, "Deploying on WebSphere Application Server"

23.1.3.1 Deploying on Tomcat Application Server

1. Copy the `ROOT` folder in `<SC_INSTALL_DIR>\fw-site-capture\webapps\` and overwrite the `ROOT` folder in the `<apache-tomcat-6.0.29 directory>\webapps\` directory.
2. To start Site Capture:
   a. Ensure that WebCenter Sites is running.
   b. Start the application server (run the `startup.bat` or `startup.sh` file in the `<apache-tomcat-6.0.29 directory>\bin` folder).
3. Continue to Section 23.2, "Post-Installation Steps."

23.1.3.2 Deploying on WebLogic Application Server

1. Copy the files named `antlr-2.7.7.jar` and `commons-lang-2.5.jar` in Site capture's `WEB-INF\lib` folder:

   `<SC_INSTALL_DIR>\fw-site_capture\webapps\ROOT\WEB-INF\lib`

   and paste them into the WebLogic installation path:

   `<Weblogic Home>\wlserver_10.3\common\lib.`

2. Set the pre-class path in the `setdomainEnv.cmd` or `setdomainEnv.sh` file (located in `<Weblogic Home>\user_projects\domains\<domain name>\bin`) for the Site Capture domain.

   Below is an example of where the changes need to be made:

   – On Windows (in `setdomainEnv.cmd`):

   ```
   set WL_HOME=E:\<WL HOME\Weblogic\wlserver_10.3
   set PRE_CLASSPATH=%WL_HOME%\common\lib\antlr-2.7.7.jar;%WL_HOME
   %\common\lib\commons-lang-2.5.jar(in windows machine)
   for %%i in ("%WL_HOME%") do set WL_HOME=%%~fsi
   ```

   – On Linux (in `setdomainEnv.sh`):

   ```
   WL_HOME="/root/Oracle/Middleware/wlserver_10.3"
   PRE_CLASSPATH=$WL_HOME/common/lib/antlr-2.7.7.jar:
   $WL_HOME/common/lib/commons-lang-2.5.jar
   export WL_HOME
   ```

3. Deploy the Site Capture application:
   a. Go to the deployment section in the console and select the path to the `ROOT` folder located in `<SC_INSTALL_DIR>\fw-site-capture\webapps`.
   b. Select the managed server as the deployment target.
4. To start Site Capture:
   a. Ensure that WebCenter Sites is running.
   b. Start the managed server `<managed_server_name>`. For example:

   ```
   ./startmanagedweblogic.sh <managed_server_name>
   ```

   ```
   http://<admin_server_hostname>:<admin_server_port>/
   ```

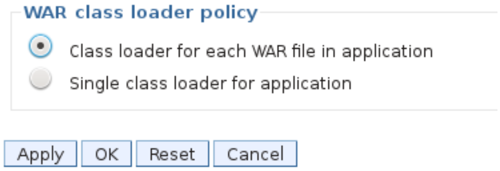
5. Continue to Post-Installation Steps.

23.1.3.3 Deploying on WebSphere Application Server

1. In the WebSphere Admin interface, do the following:

   a. Go to the path **Application > Application Types > WebSphere enterprise applications** and select the `ROOT.war` file located in `<SC_INSTALL_DIR>\fw-site-capture\webapps\`.

   b. From the WebSphere console, navigate to **Enterprise Applications > Site Capture Webapp > Class loader** (Class loading and update detection) to display the classloading options. Site Capture requires default configuration for War Policy. Ensure that the **Class loader for each WAR file in application** option is selected under the WAR class loader policy section.



WAR class loader policy

● Class loader for each WAR file in application

○ Single class loader for application

   c. Complete the deployment process.

2. To start Site Capture:

   a. Ensure that WebCenter Sites is running.

   b. Start the application server.

3. Continue to Section 23.2, "Post-Installation Steps."

23.2 Post-Installation Steps

When Site Capture is installed and the application server is started, Site Capture starts and automatically registers itself as an application enabled on AdminSite. In addition, Site Capture adds the necessary tables and records to its database, unless the database is Oracle.

To complete and verify your Site Capture installation, complete the steps in the following sections:

- Section 23.2.1, "If Site Capture Is Using an Oracle Database"
- Section 23.2.2, "Add the Site Capture URL in the customBeans.xml File"
- Section 23.2.3, "Verifying the Site Capture Application"
- Section 23.2.4, "Authorizing Users to Work with Site Capture"

23.2.1 If Site Capture Is Using an Oracle Database

When Site Capture uses the Oracle database, the `sql` script that creates the schema must be run manually. The script, named `crawler_oracle_db.sql`, is available in the `<SC_INSTALL_DIR>/Sql-Scripts` folder.

23.2.2 Add the Site Capture URL in the customBeans.xml File

1. In your installation, navigate to `<Sites_Installation_Directory>/bin/customBeans.xml`.

2. Open the `customBeans.xml` file.

3. In the `bean id` section, add the Site Capture URL as follows:

```
<bean id="RequestParamValidator"
class="com.fatwire.cas.web.RequestParamValidator">
     <property name="validUrls">
          <list>
             <value>http://<Sites_Host_Name>:<Sites_Port>/<Sites_Context_
root>/*</value>
             <value>http://<Site Capture Hostname>:<SiteCapture_port>/__
admin/*</value>
          </list>
     </property>
</bean>
```

4. Restart WebCenter Sites application servers and Site Capture servers.

23.2.3 Verifying the Site Capture Application

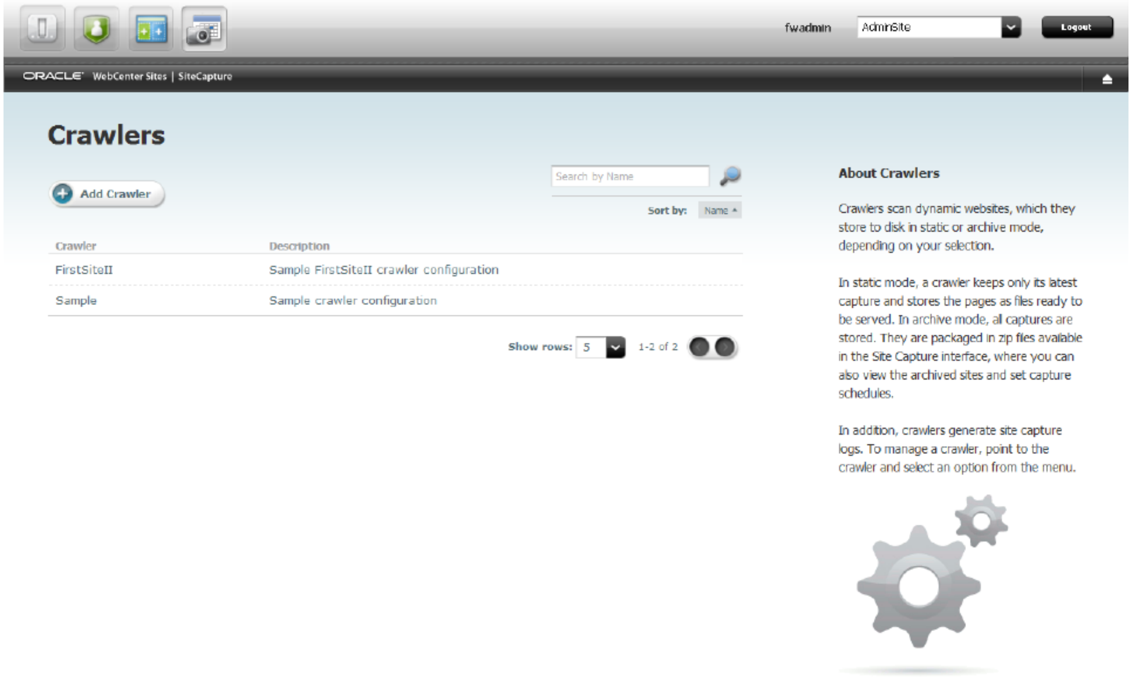
Ensure that the Site Capture application is available in the WEM Admin interface:

1. Log in to WebCenter Sites as a general administrator.
2. Navigate to AdminSite and select the WEM Admin application.
3. In the menu bar, click **Apps** and verify that **Site Capture** is listed on the "Apps" page.
4. Navigate to the Site Capture application, represented by the icon shown in Figure 23–16.

Figure 23–16 Site Capture Application Icon

The home page opens. If you chose to install the sample crawlers, they are listed on the home page as **Sample** and **FirstSiteII** (Figure 23–17).

Figure 23–17 FirstSiteII and Sample Crawlers



Crawlers

Crawler	Description
FirstSiteII	Sample FirstSiteII crawler configuration
Sample	Sample crawler configuration

About Crawlers

Crawlers scan dynamic websites, which they store to disk in static or archive mode, depending on your selection.

In static mode, a crawler keeps only its latest capture and stores the pages as files ready to be served. In archive mode, all captures are stored. They are packaged in zip files available in the Site Capture interface, where you can also view the archived sites and set capture schedules.

In addition, crawlers generate site capture logs. To manage a crawler, point to the crawler and select an option from the menu.

23.2.4 Authorizing Users to Work with Site Capture

The Site Capture application is designed for WebCenter Sites general administrators and developers with general administrator rights. All such users have access to AdminSite, where the Site Capture application is registered and running. (Access is granted via the `GeneralAdmin` role on AdminSite and the user's membership in the `RestAdmin` security group. For more information about user authorization and the WEM Admin interface, see the chapter "Creating and Authorizing Users" in the*Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

Site Capture users also require administrative access to its host machine, especially to retrieve statically captured sites and crawler logs.

23.3 Next Step

- If you wish to set up publishing-triggered site capture, see Chapter 24, "Enabling Publishing-Triggered Site Capture."
- To get started with Site Capture, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* for information about navigating the interface and writing crawler configuration code.

24

Enabling Publishing-Triggered Site Capture

You have the option to enable publishing-triggered site capture once the Site Capture application is installed.

This chapter contains the following topics:

- Section 24.1, "Integrating Site Capture with Oracle WebCenter Sites' Publishing Process"
- Section 24.2, "Next Step"

24.1 Integrating Site Capture with Oracle WebCenter Sites' Publishing Process

Site capture can be triggered by the completion of a RealTime publishing session. If you wish to enable publishing-triggered site capture, complete the steps in this section after you have installed the Site Capture application (Chapter 23, "Procedures for Installing Site Capture"). The steps show you how to integrate the Site Capture application with the WebCenter Sites publishing system to enable its communication with Site Capture. For some of the possible system configurations, see Figure 21–2, "Single-Server Installation Enabled for Publishing-Triggered Site Capture" and Figure 21–3, "Clustered Installation Enabled for Publishing-Triggered Site Capture".

To integrate Site Capture with WebCenter Sites' publishing process

1. On the WebCenter Sites source system:

   a. Deploy the `fw-crawler-publish-listener-1.0.jar` file to the `<cs_deploy>/WEB-INF/lib` folder.

   b. Unzip the `fw-crawler-publish-listener-1.0-elements.zip` file and import `FW_PublishingEventRegistry.html` using CatalogMover.

      This step creates a `RemoteElementInvokingPublishingEventListener` record in the `FW_PublishingEventRegistry` table in the WebCenter Sites database, which will allow publish events to call the `InvokeCrawler` element on the WebCenter Sites target system.

   c. Restart the WebCenter Sites source system.

2. On the WebCenter Sites target system:

   a. Using CatalogMover, import `ElementCatalog.html` and `SiteCatalog.html` from the unzipped `fw-crawler-publish-listener-1.0-elements.zip` file extracted in step 1b above (on the WebCenter Sites source system).

      This step imports the `InvokeCrawler.jsp`, which is used to start the crawler(s) in the Site Capture application.

> **Note:** The crawler(s) must be defined in the publishing destination definition for Site Capture and in the Site Capture application. For more information, see Section 24.2, "Next Step."

b. Copy the `crawler.properties` file (in the `/<cs_deploy>/WEB-INF/classes` folder) and configure the properties listed below:

   – `sc.url:` Do one of the following:

     For a single-server installation, specify the URL of the Site Capture application:

     ```
     sc.url=http://<sitecapturehost:sitecaptureport>/__admin
     ```

     For a clustered installation, specify the URL of the load balancer:

     ```
     sc.url=http://<loadbalancerhost:loadbalancer>/__admin
     ```

   – `cas.url=http://<cashost:casport>/cas`

     Specify the CAS application that is pointed to by the Site Capture application:

   – `cs.username=<RestAdmin User>`

     Specify the user name of the WebCenter Sites general administrator exactly as it was specified during the Site Capture installation process:

   – `cs.password=<Password>`

     Specify the above user's password exactly as it was specified during the Site Capture installation process:

c. Deploy the `fw-crawler-publish-listener-1.1.jar` file to the `<cs_deploy>/WEB-INF/lib` folder on the target WebCenter Sites system.

3. You have completed the integration process. Continue to Section 24.2, "Next Step" for a summary on setting up a publishing-triggered site capture operation.

24.2 Next Step

At this point, you have completed integrating Site Capture with the WebCenter Sites RealTime publishing process. However, for publishing-triggered site capture to work, the following conditions must also be satisfied:

- A RealTime publishing destination definition must be configured on the source system to name the crawler(s) that will be invoked to capture the newly published site. The definition must also specify the crawlers' capture mode.
- The crawler(s) named in the step above must exist in the Site Capture application. In addition, the `CrawlerConfigurator.groovy` file for each crawler must specify at least a valid starting URI and link extraction logic for the crawler.

Note: Once the above configuration steps are completed, you will publish the site from the source WebCenter Sites system to the target WebCenter Sites system. When publishing ends, site capture begins and proceeds as follows:

1. The source WebCenter Sites system calls the `InvokeCrawler` element on the target system.
2. The target WebCenter Sites system communicates with the Site Capture application and invokes the crawler(s).
3. The WebCenter Sites target system communicates crawler invocation status to the WebCenter Sites source system. Both the source and target systems record the status information in their own log files (`futuretense.txt`, by default).

At the same time, the invoked crawlers capture site resources either statically or in archive mode, depending on your settings in the publishing destination definition.

When setting up publishing-triggered site capture, you can configure as many publishing destination definitions and invoke as many crawlers as necessary. When you are ready to proceed with the configuration steps above, refer to the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* for instructions. In the same guide, you will find information about navigating the Site Capture interface, setting up a site capture operation, and coding a crawler's configuration file to control the site capture process.

Part VII

Installing Oracle WebCenter Sites: Analytics

Analytics plugs into WebCenter Sites to provide users with a comprehensive collection of site traffic information, delivered in the form of customizable reports. This part contains procedures for installing and configuring Oracle WebCenter Sites: Analytics to monitor traffic on websites powered by Oracle WebCenter Sites and to analyze visitors' interactions with content created in Oracle WebCenter Sites, including its Engage application.

> **Note:** The overview and procedures discussed in this part are for Oracle WebCenter Sites: Analytics version 11.1.1.6.0.

Part VII contains the following chapters:

25

Overview of Analytics Architecture

This chapter provides an overview of the components that make up the Analytics suite, and outlines the scenarios that you can choose to implement when installing Analytics.

This chapter contains the following sections:

- Section 25.1, "Components of an Analytics Installation"
- Section 25.2, "Installation Scenarios"
- Section 25.3, "Process Flow"
- Section 25.4, "Terms and Definitions"

25.1 Components of an Analytics Installation

Analytics is a modular system allowing for a high degree of scalability. An Analytics installation consists of the following components, which communicate with each other through JDBC for database access, connections for HTTP, RMI, and proprietary Socket protocols:

- Hadoop
- WebCenter Sites: Analytics
- Load Balancer

Hadoop

Hadoop provides distributed data storage (HDFS) and distributed data processing (`Map/Reduce`). The **Hadoop Distributed File System (HDFS**) stores input and output files of Hadoop programs in a distributed manner throughout the Hadoop cluster, thus providing high aggregated bandwidth.

WebCenter Sites: Analytics

- **Analytics data capture application (also called 'Analytics Sensor')** – web application that captures data on the activities of visitors as they browse your online site, and stores that data on the local file system.(For data capture to work, you must embed a special tag, `AddAnalyticsImgTag`, into the pages that you wish to monitor. The tag triggers the data capture process.)
- **Hadoop Distributed File System (HDFS) Agent** takes the raw data collected by the data capture server and copies it from the local file system to HDFS.
- **Hadoop Jobs (Scheduler)** runs jobs in a parallel and distributed fashion in order to efficiently compute statistics on the raw data that is stored in HDFS.

Hadoop implements a computational paradigm named `Map/Reduce`, which divides a large computation into smaller fragments of work, each of which may be executed or re-executed on any node in the cluster. `Map/Reduce` requires a combination of `jar` files and classes, all of which are collected into a single `jar` file that is usually referred to as a `job` file. To execute a job, you submit it to a `JobTracker`. Hadoop Jobs then responds with the following actions:

- Schedules and submits the jobs to `JobTracker`.
- Processes raw data captured by the data capture server into statistical data and then writes it to the Analytics database.

Hadoop provides a web interface to browse HDFS and to determine the status of the jobs.

- **Analytics database** – stores the aggregated and statistical data on the raw data captured by the data capture server.
- **Analytics reporting and administration web applications**
  - The reporting component provides the user interface, used to generate reports.
  - The administration component provides the administration interface, used to integrate Analytics with your WebCenter Sites system.

  Typically, the reporting and administration components reside on the same computer.

Load Balancer

Load balancer is used to link multiple data capture servers in order to increase performance. Load balancing is also recommended for failover.

A firewall is highly recommended, to protect your WebCenter Sites and Analytics systems from intrusion. The modular nature of Analytics gives you the option to install Analytics in several ways. Section 25.2, "Installation Scenarios" describes the more common approaches.

25.2 Installation Scenarios

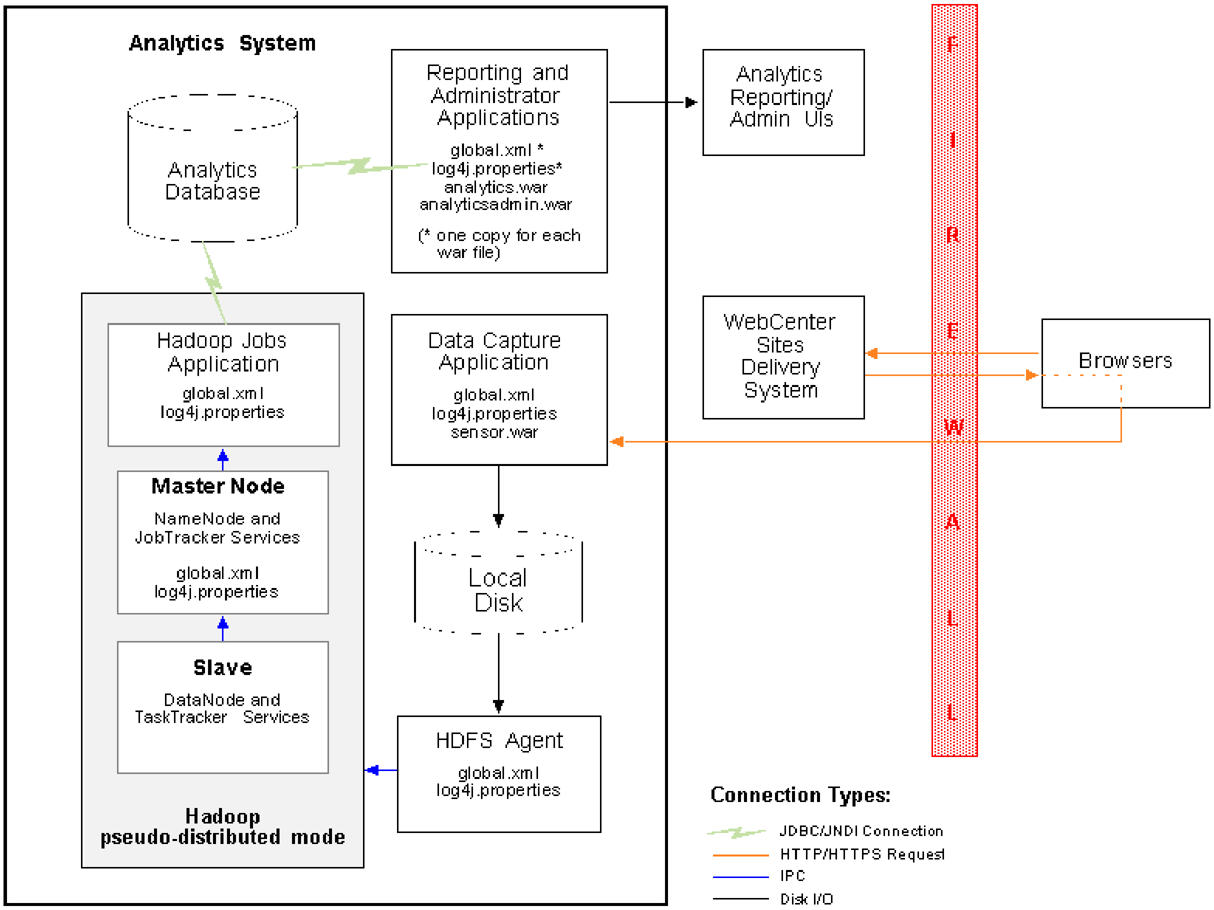
This section describes the different installation scenarios that you can choose to follow when implementing Analytics on your site. The scenarios are:

- Section 25.2.1, "Single-Server Installation: Analytics and Its Database on a Single Server"
- Section 25.2.2, "Dual-Server Installation: Analytics and Its Database on Separate Servers"
- Section 25.2.3, "Enterprise-Level Installation: Fully Distributed"

25.2.1 Single-Server Installation: Analytics and Its Database on a Single Server

In this scenario, all Analytics components reside on a single, dedicated computer. This scenario works best in situations when you need to test and experiment with Analytics. Figure 25–1 illustrates a single-server Analytics installation and indicates where configuration files reside and services run. Arrows represent data flow.

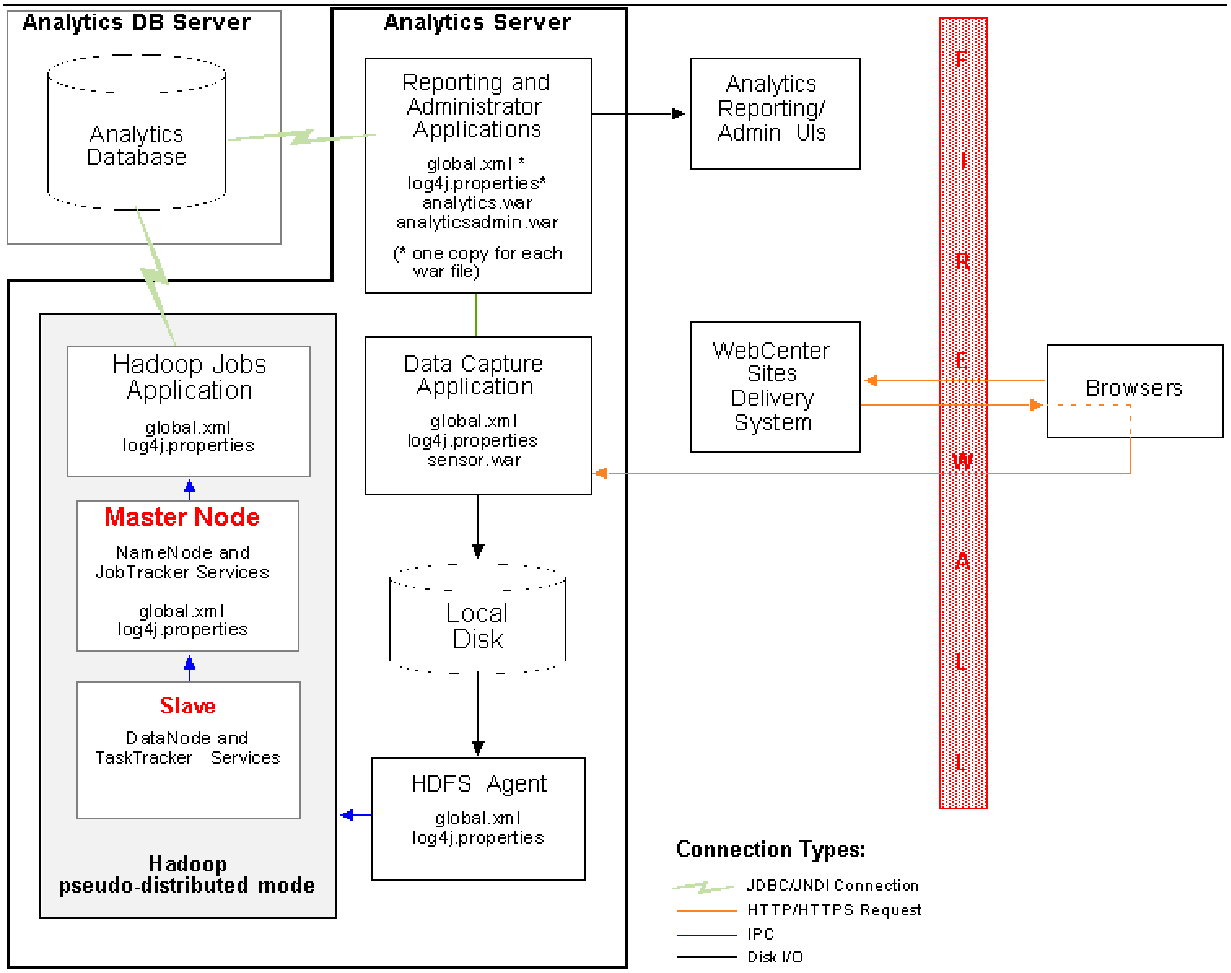
Figure 25–1 Single-Server Analytics Installation



25.2.2 Dual-Server Installation: Analytics and Its Database on Separate Servers

In this scenario, Analytics components **except for the Analytics database** are hosted on a single, dedicated server; the Analytics database is installed on its own server. This scenario works best in situations when you need to test and experiment with Analytics under increased performance conditions (isolating database transactions from Hadoop jobs minimizes their competition for resources). Figure 25–2 illustrates a dual-server Analytics installation and indicates where configuration files reside and services run. Arrows represent data flow.

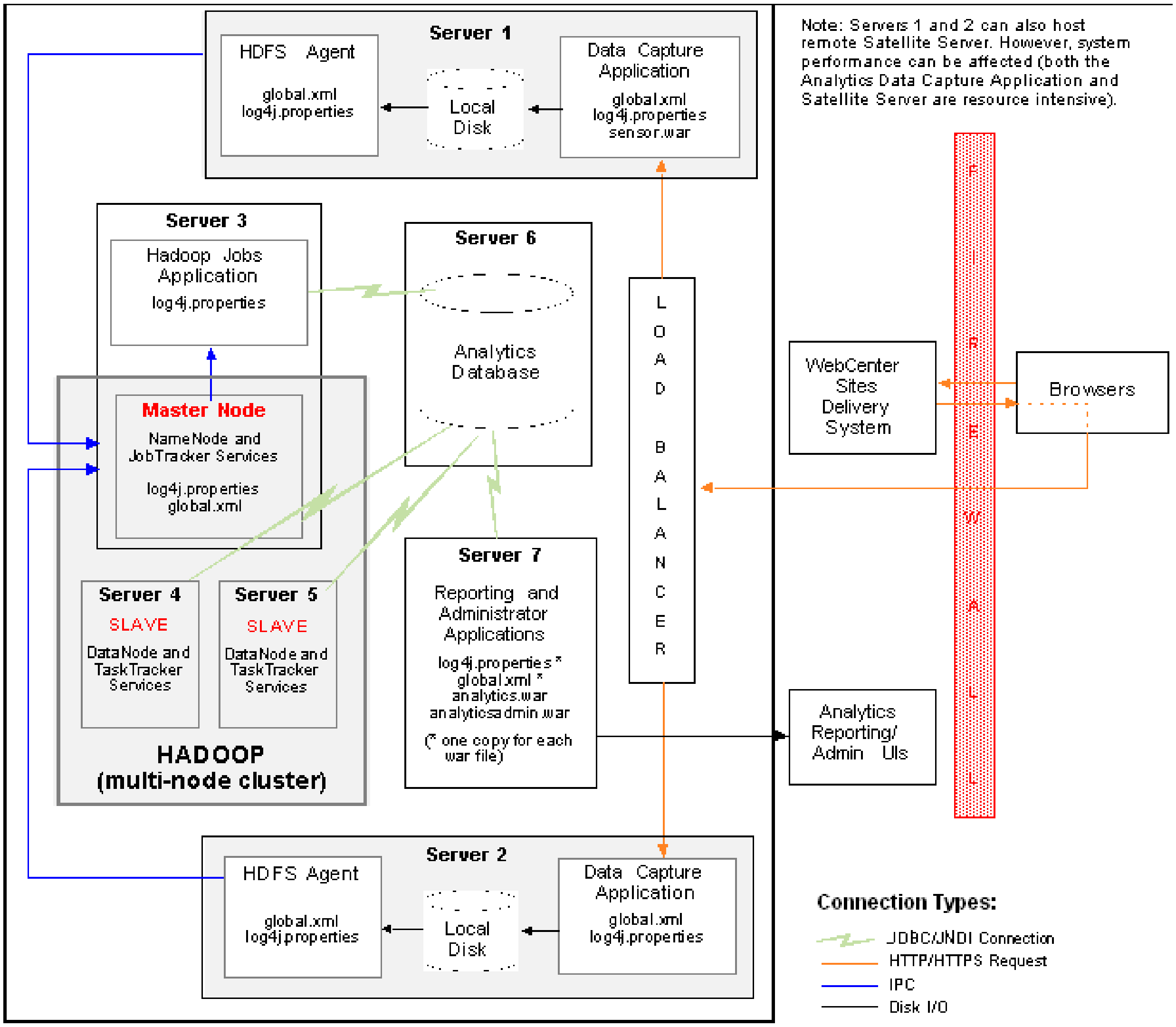
Figure 25–2 Dual-Server Analytics Installation



25.2.3 Enterprise-Level Installation: Fully Distributed

In this scenario, Analytics components run on separate computers. While more complex, this approach allows for scalability and provides better performance, as each component has dedicated processing power at its disposal. Figure 25–3 illustrates an enterprise-level installation and indicates where configuration files reside and services run. Arrows represent data flow. **For information about installing Analytics with remote Satellite Server, see the note in Figure 25–3.**

Figure 25–3 Enterprise-Level Analytics Installation



Note: Servers 1 and 2 can also host remote Satellite Server. However, system performance can be affected (both the Analytics Data Capture Application and Satellite Server are resource intensive).

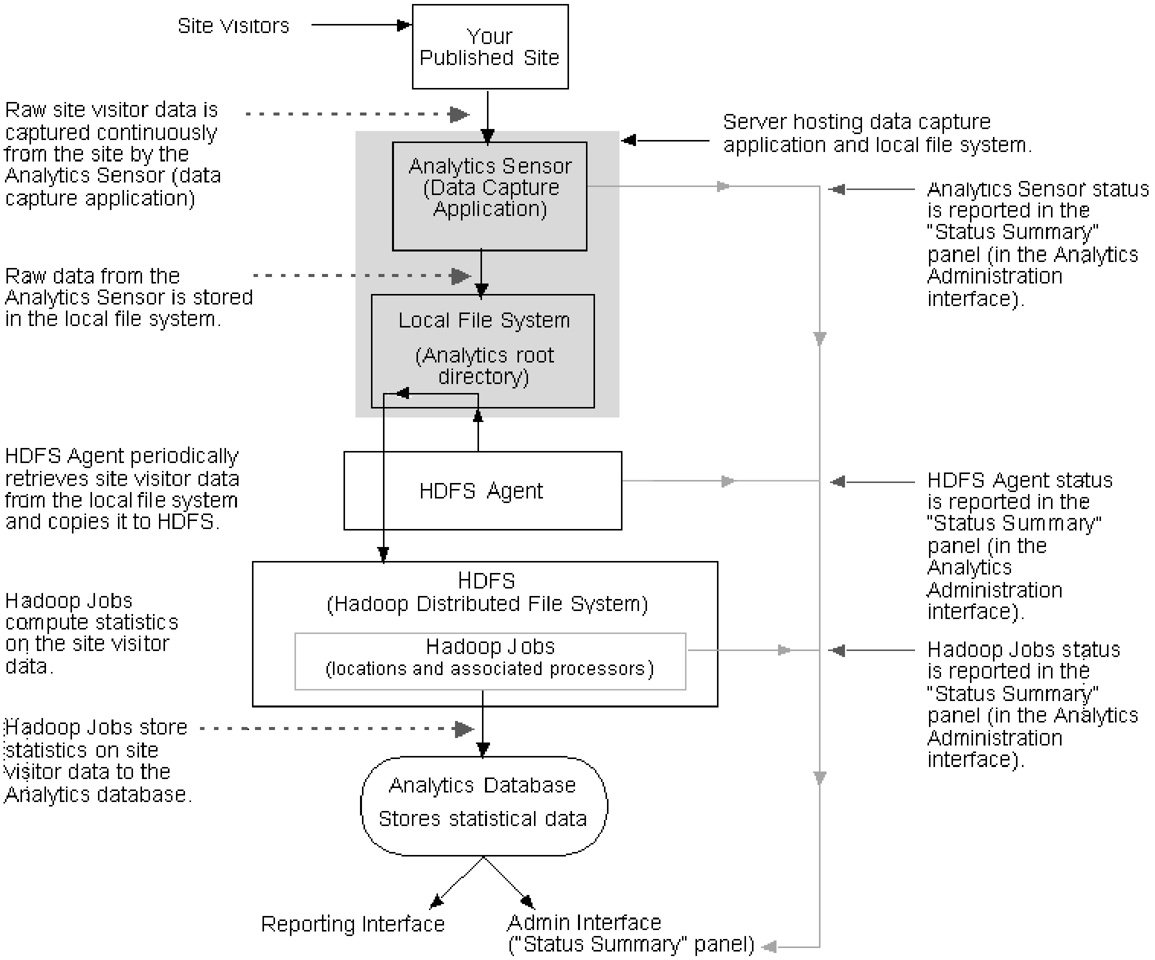
25.3 Process Flow

In a functional Analytics installation, raw site visitor data is continuously captured by the Analytics Sensor (data capture application), which then stores the data into the local file system. The raw data in the file system is called on periodically by the HDFS Agent. The HDFS Agent copies the raw data to the Hadoop Distributed File System (HDFS), where Hadoop jobs process the data (Figure 25–4).

Hadoop jobs consist of locations and Oracle-specific processors that read site visitor data in one location, statistically process that data, and write the results to another location for pickup by the next processor. When processing is complete, the results (statistics on the raw data) are injected into the Analytics database.

The status of Hadoop Jobs can be monitored from the "Status Summary" panel of the Analytics Administration interface. Detailed information about data processing and the "Status Summary" panel is available in the chapter "*Reference: Hadoop Jobs Processors and Locations* in the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide*.

Figure 25–4 Hadoop Jobs Process Flow


Site Visitors
Your Published Site
Raw site visitor data is captured continuously from the site by the Analytics Sensor (data capture application)
Server hosting data capture application and local file system.
Analytics Sensor (Data Capture Application)
Analytics Sensor status is reported in the "Status Summary" panel (in the Analytics Administration interface).
Raw data from the Analytics Sensor is stored in the local file system.
Local File System
(Analytics root directory)
HDFS Agent periodically retrieves site visitor data from the local file system and copies it to HDFS.
HDFS Agent
HDFS Agent status is reported in the "Status Summary" panel (in the Analytics Administration interface).
Hadoop Jobs compute statistics on the site visitor data.
HDFS
(Hadoop Distributed File System)
Hadoop Jobs
(locations and associated processors)
Hadoop Jobs status is reported in the "Status Summary" panel (in the Analytics Administration interface).
Hadoop Jobs store statistics on site visitor data to the Analytics database.
Analytics Database
Stores statistical data
Reporting Interface
Admin Interface
("Status Summary" panel)

25.4 Terms and Definitions

The terms listed below are used frequently throughout this guide. The glossary defines additional terms.

- The "Analytics Data Capture Application" is also referred to as the "Analytics Sensor," or simply "sensor."
- The term "site" in the context of installation/configuration procedures and in the interpretation of report statistics refers to the content management (CM) site that functions as the back end of your online site (or one of its sections).
- "FirstSite II" is the sample content management site, used throughout this guide to support examples of reports and to provide code snippets. FirstSite II is also the back end of the online sample site named "etravel."

26

Prerequisites for Installing Analytics

This chapter contains prerequisites for installing and configuring Analytics to run on the WebCenter Sites web application.

This chapter contains the following sections:

- Section 26.1, "Pre-Installation Checklist"
- Section 26.2, "Next Step"

26.1 Pre-Installation Checklist

To install Analytics, you will run a silent installer (a Java-based script). Before running the silent installer, verify the availability and configuration of all components that support Analytics.

- Section 26.1.1, "Required Experience"
- Section 26.1.2, "System Architecture"
- Section 26.1.3, "WebCenter Sites: Analytics Kit"
- Section 26.1.4, "Installing Hadoop"
- Section 26.1.5, "WebCenter Sites and Supporting Documentation"
- Section 26.1.6, "WebCenter Sites: Analytics Silent Installer"
- Section 26.1.7, "WebCenter Sites: Analytics Supporting Software"
- Section 26.1.8, "Environment Variables"
- Section 26.1.9, "Support for Charts"

26.1.1 Required Experience

To install Analytics, you must have experience installing and configuring enterprise-level software (such as application servers and databases), and setting system operating parameters.

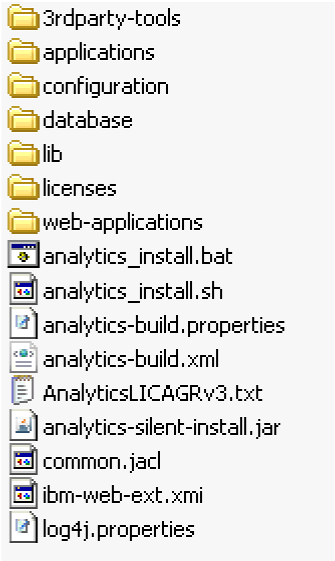
26.1.2 System Architecture

- Read Chapter 25, "Overview of Analytics Architecture" to familiarize yourself with the architecture of the Analytics product and the supported installation options.
- Read the release notes and the *Oracle WebCenter Sites Certification Matrix* to ensure that you are using certified versions of the third-party software that supports Analytics.

26.1.3 WebCenter Sites: Analytics Kit

Make sure you have a licensed Analytics Kit (`analytics2.5.zip`). The kit is organized as shown in Figure 26–1.

Figure 26–1 Analytics Kit's Directory Structure



3rdparty-tools
applications
configuration
database
lib
licenses
web-applications
analytics_install.bat
analytics_install.sh
analytics-build.properties
analytics-build.xml
AnalyticsLICAGRv3.txt
analytics-silent-install.jar
common.jacl
ibm-web-ext.xmi
log4j.properties

The kit contains the Analytics silent installer files, supporting third-party software, and the Analytics suite. The Analytics suite consists of the following applications:

- Analytics Data Capture web application (also called "sensor")
- Analytics Administrator web application
- Analytics Reporting web application (reporting engine and interface)
- Hadoop Distributed File System (HDFS) Agent
- Hadoop Jobs (scheduler)

26.1.4 Installing Hadoop

> **Note:** In the Analytics Kit, the `3rdparty-tools` folder contains Hadoop binaries. Use the Hadoop binaries to install Hadoop (and not the files that are available on the Hadoop web site).

1. In this section, you will install and configure Hadoop in one of the following modes: local, pseudo-distributed, or fully distributed (*recommended*), whichever is best suited to meet your development, scalability, and performance requirements. The modes are described as follows:

   – The local (standalone) mode is used for development and debugging. By default, Hadoop is configured to run in a non-distributed mode, as a single Java process.

   – The pseudo-distributed mode is used in single-server installations. In this mode, all the Hadoop services (for example, `NameNode`, `JobTracker`, `DataNode` and TaskTracker) run on a single node, and each service runs as a separate Java process.

- The fully distributed mode is used for enterprise-level installations. In this mode, Hadoop runs on multiple nodes in a parallel and distributed manner. A minimum of two nodes is required to set up Hadoop: One machine acts as the master node, while the remaining machines act as slave nodes. On the master node, the NameNode and JobTracker services will be running. On the slave nodes, the DataNode and TaskTracker services will be running.

2. For Hadoop installation instructions, refer to the *Hadoop Quick Start* site. The URL at the time of this writing is:

   http://hadoop.apache.org/docs/r0.18.3/quickstart.pdf

   If you install Hadoop in either pseudo- or fully distributed mode, you must configure a property file called hadoop-site.xml on all master and slave computers. Recommended property values and a sample file are available in this section.

   To configure hadoop-site.xml

   a. Configure the hadoop-site.xml file as shown in Table 26–1. Your configured file should look similar to the sample hadoop-site.xml file shown on page 26-5.

   b. If you are installing in fully distributed mode, copy the configured hadoop-site.xml to all master and slave computers.

Table 26–1 Properties in hadoop-site.xml

Property	Description	Sample Value
fs.default.name	Name of the default file system. A URI whose scheme and authority determine the FileSystem implementation. The URI's scheme determines the configuration property (fs.SCHEME.impl) that names the FileSystem implementation class. The URI's authority is used to determine the host, port (and so on) for a file system.	hdfs://<ipaddress>:<port1> where <ipaddress> is the IP address of the master node, and <port1> is the port on which NameNode will listen for incoming connections. For example: hdfs://192.0.2.1:9090
mapred.job.tracker	Host and port on which the MapReduce job tracker runs. If this property is set to local, then jobs are run in-process, as a single map and reduce task.	<ipaddress>:<port2> local For example: 192.0.2.1:7070 **Note:** In fully distributed mode, enter the IP address of the master node.
dfs.replication	Default block replication. The number of replications for any file that is created in HDFS. The value should be equal to the number of DataNodes in the cluster. The default is used if dfs.replication is not set.	<equal to the number of data nodes>

Table 26–1 (Cont.) Properties in `hadoop-site.xml`

Property	Description	Sample Value
`dfs.permissions`	Enables/disables permission checking in HDFS. ■ `true` enables permission checking in HDFS. ■ `false` disables permission checking, but leaves all other behavior unchanged. Switching from one value to the other does not change the mode, owner or group of files, or directories.	`true | false`

`hadoop.tmp.dir`	Hadoop file system location on the local file system.	`/work/hadoop/hadoop-0.18.2/tmp`
		`/hadoop-${user.name}`

`mapred.child.java.opts`	Java options for the `TaskTracker` child processes.
	The following parameter, if present, will be interpolated: `@taskid@` will be replaced by the current `TaskID`. Any other occurrences of `@` will be unchanged.
	For example:
	To enable verbose gc logging to a file named for the `taskid` in `/tmp` and to set the heap maximum to a gigabyte, pass a value of:
	`-Xmx1024m -verbose:gc`
	`-Xloggc:/tmp/@taskid@.gc`
	The configuration variable `mapred.child.ulimit` can be used to control the maximum virtual memory of child processes.
		`-Xmx1024m`

`mapred.tasktracker.` `expiry.interval`	Time interval, in milliseconds, after which a TaskTracker is declared 'lost' if it does not send heartbeats.	`60000000`
`mapred.task.timeout`	Number of milliseconds before a task is terminated if it neither reads an input, writes an output, nor updates its status string.	`60000000`
`mapred.map.tasks`	Default number of map tasks per job. Typically set to a prime number, several times greater than the number of available hosts. Ignored when `mapred.job.tracker` specifies the local IP address.	`11`
`mapred.reduce.tasks`	Default number of reduce tasks per job. Typically set to a prime number, close to the number of available hosts.Ignored when `mapred.job.tracker` specifies the local IP address.	`7`

`mapred.tasktracker`
`.map.tasks.maximum`
	Maximum number of `map` tasks that will be run simultaneously by a TaskTracker.
	Specify a number that
	exceeds the value of `mapred.map.tasks`.
		Integer that exceeds the value of `mapred.map.tasks`

`mapred.tasktracker`
`.reduce.tasks.maximum`
	Maximum number of `reduce` tasks that will be run simultaneously by a TaskTracker.
	Specify a number that
	exceeds the value of `mapred.reduce.tasks`.
		Integer that exceeds the value of `mapred.reduce.tasks`

Sample hadoop-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>fs.default.name</name>
  <value>hdfs://192.0.2.1:9090</value>
  <description>The name of the default file system.A URI whose
  scheme and authority determine the FileSystem implementation. The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>

 <property>
  <name>mapred.job.tracker</name>
  <value>192.0.2.1:7090</value>
  <description>The host and port that the MapReduce job tracker runs
  at.If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>

<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is
created.
  The default is used if replication is not specified in create time.
  </description>
</property>

<property>
  <name>dfs.permissions</name>
  <value>false</value>
  <description>
    If "true", enable permission checking in HDFS.
    If "false", permission checking is turned off,
    but all other behavior is unchanged.
    Switching from one parameter value to the other does not change the
    mode,owner or group of files or directories.
  </description>
</property>

<property>
  <name>hadoop.tmp.dir</name>
  <value/work/hadoop/hadoop-0.18.2/tmp/hadoop-${user.name}</value>
  <description>A base for other temporary directories.</description>
</property>

 <property>
  <name>mapred.child.java.opts</name>
  <value>-Xmx200m</value>
  <description>Java opts for the task tracker child processes.
  The following symbol, if present, will be interpolated: @taskid@ is
  replaced  by current TaskID. Any other occurrences of '@' will go
  unchanged.

   For example, to enable verbose gc logging to a file named for the
  taskid in  /tmp and to set the heap maximum to be a gigabyte, pass a
  'value' of: -Xmx1024m -verbose:gc -Xloggc:/tmp/@taskid@.gc
  The configuration variable mapred.child.ulimit can be used to control
  the maximum virtual memory of the child processes.
  </description>
</property>

<property>
  <name>mapred.tasktracker.expiry.interval</name>
  <value>600000</value>
  <description>Expert: The time-interval, in miliseconds, after which
  a tasktracker is declared 'lost' if it doesn't send heartbeats.
  </description>
</property>

<property>
  <name>mapred.task.timeout</name>
  <value>600000</value>
  <description>The number of milliseconds before a task will be
  terminated if it neither reads an input, writes an output, nor
  updates its status string.
  </description>
</property>

<property>
  <name>mapred.map.tasks</name>
  <value>2</value>
  <description>The default number of map tasks per job.Typically set
  to a prime several times greater than number of available hosts.
  Ignored when mapred.job.tracker is "local".
  </description>
</property>

<property>
  <name>mapred.reduce.tasks</name>
  <value>1</value>
  <description>The default number of reduce tasks per job.Typically
    set to a prime close to the number of available hosts.Ignored
     when mapred.job.tracker is "local".
  </description>
</property>

<property>
  <name>mapred.tasktracker.map.tasks.maximum</name>
  <value>2</value>
  <description>The maximum number of map tasks that will be run
  simultaneously by a task tracker.
  </description>
</property>

<property>
  <name>mapred.tasktracker.reduce.tasks.maximum</name>
  <value>2</value>
  <description>The maximum number of reduce tasks that will be run
  simultaneously by a task tracker.
  </description>
```

```
</property>

</configuration>
```

3. Once Hadoop is installed and configured, verify the Hadoop cluster:

   a. To determine whether your distributed file system is running across multiple machines, open the Hadoop HDFS interface on your master node:

      ```
      http://<hostname_MasterNode>:50070/
      ```

      The HDFS interface provides a summary of the cluster's status, including information about total/remaining capacity, active nodes, and dead nodes. Additionally, it allows you to browse the HDFS namespace and view the content of its files in the web browser. It also provides access to the local machine's Hadoop log files.

   b. View your MapReduce setup, using the MapReduce monitoring web app that comes with Hadoop and runs on your master node:

      ```
      http://<hostname_MasterNode>:50030/
      ```

26.1.5 WebCenter Sites and Supporting Documentation

- Ensure that you have a licensed version of the WebCenter Sites web application and it is powering a fully functional online site.
- Have WebCenter Sites documentation handy. Various steps in the installation process require you to create and configure third-party components, and integrate Analytics with WebCenter Sites. Download the following guides:
  - *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software* – contains instructions for creating and configuring the Oracle database for the WebCenter Sites environment.
  - *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* – contains instructions for creating and assigning roles during the integration process.
  - *Oracle Fusion Middleware WebCenter Sites Developer's Guide* – contains instructions for using CatalogMover in the integration process.

26.1.6 WebCenter Sites: Analytics Silent Installer

The Analytics silent installer is a Java-based script (developed on Ant) that installs Analytics. The silent installer is provided in the Analytics Kit.

- Ensure that the currently supported version of Ant (required by the silent installer) is running on each server where the silent installer, itself, will be running.
- Familiarize yourself with the installation scenarios that are covered in this guide and select the scenario that is appropriate for your operations. The scenarios are:
  - Single-server installation: Figure 25–1
  - Dual-server installation: Figure 25–2
  - Enterprise-level installation: Figure 25–3

Note: The silent installer script installs Analytics *locally* (on the computer where it is executed) and non-interactively. A silent installation involves all the steps from preparing the installation folders and setting up the database to deploying the web applications and utility programs.

26.1.7 WebCenter Sites: Analytics Supporting Software

- Section 26.1.7.1, "Databases"
- Section 26.1.7.2, "Application Servers"

26.1.7.1 Databases

- Install the Oracle database management system (DBMS) and the SQL Plus utility. Analytics schema will be installed on the Oracle database by SQL Plus.(If you need installation instructions, refer to the product vendor's documentation.)
- Create and configure an Oracle database as the Analytics database.

  If your WebCenter Sites installation runs on Oracle DBMS, you can use the same DBMS to create a database for Analytics, assuming the server has the capacity to support an additional database. Space requirements depend on the amount of site traffic data you expect to capture within a given time frame, the volume of statistics that will be computed on the captured data, and whether you plan to archive any of the raw data and statistics.

  The steps for creating and configuring an Oracle database are given below:

  1. Follow the procedures in the *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

     Note: Remember the following points:

     - When setting the Global name and SID, do not create names longer than 8 characters.
     - When creating the user, create the `analytics` user.

  2. Set the encoding to Unicode (`AL32UTF8`). Change the environment variable `nls_lang` to: `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`, using one of the following commands:

     * In Windows, enter the command

       ```
       set NLS_LANG=AMERICAN_AMERICA.AL32UTF8
       ```

     * In Linux, the command depends on the shell you are using:

       For Korn and Bourne shells:

       ```
       NLS_LANG=AMERICAN_AMERICA.AL32UTF8
       export NLS_LANG
       ```

       For C shell:

       ```
       setenv NLS_LANG AMERICAN_AMERICA.AL32UTF8
       ```

26.1.7.2 Application Servers

- Install a supported application server to host the Analytics web applications (i.e., data capture, administrator application, and reporting application). For the list of supported application servers, see the *Oracle WebCenter Sites Certification Matrix*.

> **Note:** A single-server installation requires a single application server.
>
> A multi-server installation requires up to three application servers, depending on its configuration (for example, three application servers if the data capture application, administrator application, and reporting application are installed on separate computers).

- Make sure that each application server provides a JDBC driver that works with the Analytics database. (Analytics does not ship with a JDBC driver.)

26.1.7.2.1 All Application Servers

- Configure each application server for UTF-8 character encoding.

> **Note:** The application server's encoding setting must match the value of the `encoding` parameter in `global.xml`. The value is `UTF-8`.

  - In Tomcat:

    Edit the file `$CATALINA_HOME/conf/server.xml` and set the `URIEncoding` attribute to UTF-8:

    ```
    <Connector port="8080" URIEncoding="UTF-8"/>
    ```

  - In WebSphere:

    Set the value of system property `default.client.encoding` on the JVM settings of the application server to UTF-8.

- For the application server, set the JVM parameter to:

  ```
  -Djava.awt.headless=true
  ```

- Enable DNS lookups on your application server. Your DNS server must perform DNS lookups in order for the "Hosts" report to display host names of the machines from which visitors access your site. For instructions, consult your application server's documentation.

> **Note:** If the application server is not configured to perform DNS lookups, the "Hosts" report will display IP addresses instead (just like the "IP Addresses" report).

26.1.7.2.2 JBoss Application Server Perform the following:

- Delete the common `jar` files from the `lib` folder used by JBoss (in order for the Analytics Administrator application to run).

26.1.7.2.3 WebLogic Application Server Perform the following:

- Add the `log4j jar` file to the `lib` folder for the WebLogic domain in order for the Analytics applications to create log files.

- Add the `antlr.jar` file to the `PRE_CLASSPATH` in the application server's startup command. For example:

  ```
  C:/bea/wlserver_10.3/samples/domains/wl_server/bin/setDomainEnv.cmd
  ```

26.1.7.2.4 If You are Using WebSphere

- Configure the web application class loader for "parentLast" class loading order.

26.1.8 Environment Variables

Perform the following:

- `Set JAVA_HOME` to the path of the currently supported JDK and the `PATH` variable to `$JAVA_HOME/bin`. These settings are required by Hadoop (`Hadoop-env.sh`), the HDFS Agent, and Hadoop-jobs (all of which, otherwise, will not run).

  > **Note:** On Windows, set `JAVA_HOME` to its canonical form:
  >
  > ```
  > C:\PROGRA~1\<path_to_jdk>
  > ```
  >
  > Otherwise, if the path contains spaces (for example, `C:\Program Files`), the path must be enclosed in double quotes (for example, `"C:\Program Files"`).

- On Solaris systems, add the following line to `hadoop-env.sh`:

  ```
  export PATH=$PATH:/usr/ucb
  ```

- Set `ANT_HOME` (required by the silent installer) to the correct path.

26.1.9 Support for Charts

Perform the following:

- The Swiff Chart Generator is used to render charts within Analytics reports. Install the Swiff Chart Generator either on the Analytics host (single-server installation), or on the reporting server (in multi-server installations. The reporting server hosts `analytics.war`.)

  Copies of the Swiff Chart Generator can be purchased at:

  ```
   http://www.globfx.com/
  ```

  Evaluation copies are available at:

  ```
  http://www.globfx.com/downloads/swfchartgen/
  ```

- Install Adobe Flash Player on the computers on which reports will be viewed. A free copy of Adobe Flash Player is available at:

  ```
  http://www.adobe.com/go/getflashplayer
  ```

  If you choose not to install Adobe Flash Player, you can still generate reports. However, the charts they might contain will be replaced by the **download plugin** link.

26.2 Next Step

Install Analytics, using the silent installer. For instructions, see Chapter 27, "Procedures for Installing Analytics."

27

Procedures for Installing Analytics

This chapter contains procedures for using a script to install Analytics on the WebCenter Sites web application.

This chapter contains the following sections:

27.1 Overview of the Analytics Silent Installer

The silent installer is a Java-based script, developed on Ant. The script installs Analytics *locally* (on the computer where it is executed) and non-interactively.

The more common installation scenarios are covered in this chapter. They are:

- Single-server: Installing Analytics and its database on a single server (Figure 25–1)
- Dual server: Installing Analytics and its database on separate servers (Figure 25–2)
- Enterprise-level: Analytics in fully distributed mode (Figure 25–3)

A silent installation involves all the steps from preparing the installation folders and setting up the database to deploying the web applications and utility programs. The remaining sections of this chapter, starting with Installation Steps guide you through the steps that you need to complete in order to run the silent installer. Below is a summary.

27.1.1 Installation Summary

Briefly, you will do the following to install Analytics, **after ensuring that prerequisites (in Chapter 26, "Prerequisites for Installing Analytics") are satisfied**:

1. Unzip Analytics:

   a. Unzip Analytics on the master node. (In a distributed installation, unzip Analytics on other relevant nodes, including the server that hosts SQL Plus.)

   > **Note:** Consider the following:
   >
   > - The silent installer is packaged with the Analytics product.
   > - The silent installer must be executed on the SQL Plus host in order to update the Oracle database with Analytics-specific schema.

b. Unzip Analytics on the WebCenter Sites host.

> **Note:** The silent installer must be executed on the WebCenter Sites host in order to initialize WebCenter Sites to Analytics. You will help to initialize by specifying the location of the WebCenter Sites `futuretense_xcel.ini` property file, so that it can be modified by the silent installer with Analytics-specific settings.
>
> Initializing WebCenter Sites enables the `AddAnalyticsImgTag` to capture data on site visitors. For more information about the `AddAnalyticsImgTag`, see the chapter "Enabling Data Capture for Different Types of Reports" in the *Oracle Fusion Middleware WebCenter Sites: Analytics Developer's Guide*.

2. Customize the `analytics-build.properties` file for Analytics. Customize the file on every server where the Analytics product was unzipped to declare information that the silent installer needs in order to correctly deploy the Analytics product in your environment, and to initialize Analytics to WebCenter Sites (as explained in the note above).
3. Prepare to run the installer by checking environment variables and classpaths to make sure they are properly set.
4. Install and deploy Analytics by running the silent installer on every server where the Analytics product is unzipped.
5. Configure WebCenter Sites to enable Analytics. In this step, you customize the `analytics-build.properties` file on the WebCenter Sites host.

27.1.2 Silent Installer Actions

When the silent installer starts running, it performs the following steps:

1. Prepares Analytics product folders to store all the installed components, ready for your use.
2. Prepares a separate subdirectory for the Analytics/Hadoop job scheduling system.
3. Sets up and prepares a separate subdirectory for the `hdfsagent` utility, required by the `sensor.war` web application.
4. Customizes various configuration files (shown in Chapter 28, "Tuning Analytics Configuration Parameters"), using the values that you specified in the `analytics-build.properties` file:
   - Renames `global.xml-dist` to `global.xml` and sets its properties to the values that you specified in `analytics-build.properties`.
   - Renames `log4j.properties-dist` to `log4j.properties` and sets its properties to the values that you specified in `analytics-build.properties`.
   - Customizes properties in `futuretense_xcel.ini` (one of the WebCenter Sites property files).
5. Updates your Oracle database with the Analytics database schemas by using SQL Plus and running the following scripts: `create_sys.sql`, `create_normal.sql`, and `region.sql.`
6. Unpacks the `reports.zip` archive and places the files in a subdirectory (on the local file system) referenced by the reporting engine.

7. Auto deploys the Analytics web applications (`sensor.war`, `analytics.war`, and `analyticsadmin.war`) to your designated application server.
8. The silent installer on the WebCenter Sites host updates the WebCenter Sites system, allowing it to determine that Analytics is installed. This update enables the **analytics** link, displayed in the upper left-hand corner of the WebCenter Sites Admin interface.

27.2 Installation Steps

> **Note:** Before starting the steps in this section, ensure that all prerequisites listed in Chapter 26, "Prerequisites for Installing Analytics" are satisfied.

Steps for installing Analytics are the following:

Step 1. Unzipping Analytics

Step 2. Customizing analytics-build.properties for Analytics

Step 3. Preparing to Run the Silent Installer

Step 4. Installing Analytics

Step 5. Configuring WebCenter Sites to Enable Analytics

27.2.1 Step 1. Unzipping Analytics

The Analytics product, `analytics2.5.zip`, contains the silent installer.

To unzip Analytics

1. Complete one of the following steps, depending on the type of installation you plan to create:

   – **Single-server installation** (Figure 25–1). In this scenario, you are installing Analytics on its own server (which hosts all Analytics supporting software). Unzip `analytics2.5.zip` on the server.

   – **Dual-server installation** (Figure 25–2). In this scenario, you are installing Analytics and its database on their own servers.

     – Unzip `analytics2.5.zip` on the Analytics server.

     – If SQL Plus is installed on the database server's host, unzip `analytics2.5.zip` on the database server's host. (To enable communication between SQL Plus and the Oracle database, the silent installer must be run on the server that hosts SQL Plus.)

   – **Enterprise-level installation** (Figure 25–3). In this scenario, you are installing Analytics in fully distributed mode. Unzip `analytics2.5.zip` on the master node and on each remaining server where you wish to install Analytics components.

     For example, if your load balancing scheme requires multiple Analytics Sensors (data capture applications) to be deployed on different data capture nodes, unzip `analytics2.5.zip` on all the data capture nodes.

Note: To enable SQL Plus to update the Oracle database with Analytics-specific schema, make sure that `analytics2.5.zip` is unzipped on the server that hosts SQL Plus.

1. Unzip `analytics2.5.zip` on the WebCenter Sites host.
2. Continue to Step 2. Customizing analytics-build.properties for Analytics.

27.2.2 Step 2. Customizing analytics-build.properties for Analytics

The `analytics-build.properties` property file contains all the environment-specific configuration data that is required by the Analytics silent installer. In this step, you will customize `analytics-build.properties` in order to provide installation specifications to the silent installer. (The `analytics-build.properties` file is divided into sections. Each section is specific to certain information necessary to tailor the system for your use.)

Complete the following steps on each server where you plan to install Analytics (or its components):

To customize analytics-build.properties for Analytics

1. Back up `analytics-build.properties` (located in the root of the Analytics Kit).
2. Open `analytics-build.properties` in a text editor of your choice, and set the properties as indicated in the following sections:
   - Section 27.2.2.1, "General Installation Properties"
   - Section 27.2.2.2, "Visitor Detection Properties"
   - Section 27.2.2.3, "System Configuration and Operation Defaults"
   - Section 27.2.2.4, "Web Server URL Properties"
   - Section 27.2.2.5, "Application Server Deployment Properties"
   - Section 27.2.2.6, "Database Connection Properties"
   - Section 27.2.2.7, "Hadoop Properties"

Note: **Windows Only.** When you specify the path to an installation directory, be sure to enclose the path in double quotes **if it contains spaces**.

27.2.2.1 General Installation Properties

This group of properties provides information of a generalized nature for the installation process. In this section, you will specify paths to Analytics directories, the location of third-party components, and the email addresses of administrators who can reset passwords and create accounts.

Table 27–1 General installation properties in `analytics-build.properties`

Property	Description
`Analytics.installation.path`	Absolute path to the final Analytics installation directory.

Table 27–1 (Cont.) General installation properties in ***analytics-build.properties***

Property	Description
swchart.instdir	Absolute path to the directory where the Swiff Chart product is installed. **Note:** The path must end with a slash (on UNIX) or backslash (on Windows).
forgotpassword.value	Email address for the Analytics administrator who is responsible for password recovery. Set the email address in the global.xml file.
noaccount.value	Email address of the Analytics administrator who is responsible for creating accounts. Set the email address in the global.xml file.
href.help.value	URL at which help relating to Analytics can be obtained: http://www.oracle.com/technetwork/middleware/webcenter/sites/overview/index.html

27.2.2.2 Visitor Detection Properties

Visitor detection is done by the Analytics Sensor (data capture application). By default, the analytics-build.properties file supports the Sessionfingerprint method of tracking visitors across all sites. The Sessionfingerprint method identifies each visitor by a combination of the IP address, screen resolution, and agent string.

Using Table 27–2, verify or set (as indicated) the visitor detection properties in analytics-build.properties. The properties specify how identifiers are generated by the system for inclusion in object impressions (which are captured and processed by the system). Identifiers are important to providing the correct grouping of object impressions for aggregation.

> **Note:** An **object impression** is a single invocation of the sensor servlet. For more information, see the section "Object Impressions" in *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide*.

Table 27–2 Visitor detection properties in ***analytics-build.properties***

Property	Description
sessionIdGenerator	Specifies the ID generator that is used to identify sessions. **Caution:** The default value is AppServerID. Do not modify this value. It is a reference to the object that generates the ID.
visitorIdGenerator	Specifies the ID generator that is used to identify a visitor to the site. **Note:** The default value is SessionfingerprintId. The default value generates an identifier that is a combination of IP address, screen size, and agent (browser type). If the default generator is insufficient, you have two other options by which to uniquely identify a visitor: self-organized detection and cookie method. If you wish to implement these options or refine the Sessionfingerprint configuration (to detect visitors on selected sites), you can do so after Analytics is installed, by modifying global.xml directly. Instructions are available in Chapter 29, "Configuring Visitor Detection."

27.2.2.3 System Configuration and Operation Defaults

Verify that the encoding property in analytics-build.properties is set to UTF-8 and the application server's setting is set to the same value. The silent installer will set the encoding parameter in global.xml to the value that you provide in analytics-build.properties.

Set the properties related to data processing and archiving (described in Table 27–3), as necessary:

Table 27–3 Data processing and archiving properties in analytics-build.properties

Property	Description
midnight.offset	Allows the system to derive relative midnight used for file rotation. Relative midnight and the session.rotate.delay determine when the daily cycle for capturing session data ends. (Information about session.rotate.delay can be found in Table 28–1.) Format: minutes **Default value:** 0
cs_enabled	Specifies whether buttons for navigating to the WebCenter Sites interface are enabled or disabled in the Analytics interface. **Default value:** true
archive.enabled	Specifies whether HDFS Agent archiving of raw data files is enabled. If archive.enabled is set to either true or false, the data.txt file will be deleted from the analytics root folder. To enable archiving, set this property to true. Once archiving is enabled, HDFS Agent will automatically create archives of raw Analytics data on a periodic basis by moving data.txt to the archiving folder. (The archive directory and start time are specified in the following properties: archive.output.dir and archive.start.time) **Default value:** false
archive.output.dir	Specifies the path to the directory for storing archived data files. Must be a valid URI. **Sample value:** ■ Windows: archive.output.dir=file://d:/archive ■ Linux: archive.output.dir=/analytics/archive **Format:** directory path
archive.start.time	Specifies the start time (HH:mm) for archiving raw data. The HDFS Agent will start the archiving task on a daily basis at the time specified in this property. For example, to start archiving at 4:00 PM every day, set: archive.start.time=16:00 **Format:**24-hour format, expressed as HH:mm, where HH ranges from 00-23 and mm ranges from 00-59. **Default value:** 06:00

Table 27–3 (Cont.) Data processing and archiving properties in

Property	Description
`purgejobs.enabled`	Determines when purge jobs will run. When this property is set to `true` the system will automatically schedule cleanup jobs to remove subfolders and files after they have been successfully processed. **Default value:** `false`
`sensor.requestqueue .maxsize`	Specifies `CRITICAL` condition for the Analytics Sensor. This property specifies a threshold value that triggers a `CRITICAL` (red) condition when the sensor cannot respond quickly enough to the amount of raw data that it needs to record. When the threshold is reached or exceeded, the **Analytics Sensor** component is displayed in red. The threshold value for this property is expressed as an object impression, i.e., a single invocation of the `sensor` servlet. (The **Analytics Sensor** component is represented in the **Overview** option of the **Components** tab in the panel of the Analytics Administration interface, shown in the figure Hadoop Jobs Process Flow in the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide*. **Default value:** `10000`
`sensor.requestqueue .warnsize`	Specifies `WARNING` condition for the Analytics Sensor. This property specifies a threshold value that triggers a `WARNING` (yellow) condition when the sensor cannot respond quickly enough to the amount of raw data that it needs to record. When the threshold is reached or exceeded, the **Analytics Sensor** component is displayed in yellow. The threshold value for this property is expressed as an object impression, i.e., a single invocation of the `sensor` servlet. (The **Analytics Sensor** component is represented in the **Overview** option of **Components tab** in the panel of the Analytics Administration interface, shown in the figure Hadoop Jobs Process Flow in the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide*. **Default value:** `3000`

27.2.2.4 Web Server URL Properties

In this section, you will specify the WebCenter Sites URL and the basic URL (`http://<address>:<port>`) where each Analytics web application will reside after the installation is complete. See Table 27–4.

Table 27–4 Web server URL properties in `analytics-build.properties`

Property	Description
`analytics.sensor.web.server`	URL of sensor web application.
`analytics.report.web.server`	URL of Analytics web application.
`analytics.admin.web.server`	URL of Analytics administration web application.

27.2.2.5 Application Server Deployment Properties

In this section, you will specify which application server the installer will use to deploy the Analytics web applications. You will also disable the unused application servers. See Table 27–5, Table 27–6, Table 27–7, and Table 27–8.

Note: Ensure the following:

- For the application server that will be used, set its `install` property to `true`. Set all other relevant properties.
- For each unused application server, do not delete the statement pertaining to the application server. Verify that the application server's `install` property is set to `false` (the default). Otherwise, the installer will try to deploy the Analytics web applications to that server.

Table 27–5 WebLogic deployment properties in `analytics-build.properties`

Property	Description
`install.weblogic`	Set to `true` to deploy to WebLogic.
`weblogic.userid`	WebLogic admin user id.
`weblogic.password`	Password for the WebLogic admin user specified above.
`weblogic.targets`	Server name, cluster name, or virtual host name.
`weblogic.admin.url`	URL to the admin function in WebLogic.
`weblogic.home.dir`	Path to the WebLogic home directory.

Table 27–6 JBoss deployment properties in `analytics-build.properties`

Property	Description
`install.jboss`	Set to `true` to deploy to JBoss.
`jboss.deploy.dir`	Path to the JBoss deployment directory.

Table 27–7 Tomcat deployment properties in `analytics-build.properties`

Property	Description
`install.tomcat`	Set to `true` to deploy to Tomcat.
`tomcat.home.dir`	Path to the Tomcat home directory.

Table 27–8 WebSphere deployment properties in `analytics-build.properties`

Property	Description
`install.websphere`	Set to `true` to deploy to WebSphere.
`websphere.userid`	WebSphere admin user id.
`websphere.password`	WebSphere admin password.
`websphere.home.dir`	WebSphere home directory.
`websphere.node.name`	WebSphere node name.
`websphere.base.command`	WebSphere `wsadmin` command.
`websphere.save.command`	WebSphere `save` command.
`websphere.engine.instdir.value`	Location of the Analytics `reportingengine` installation directory.

Table 27–8 (Cont.) WebSphere deployment properties in **`analytics-build.properties`**

Property	Description
`websphere.cell.name`	Namespace that has been defined to represent a single node (machine instance) or multiple nodes where a software component is distributed and run. When installing an Analytics application on the WebSphere application server, you must specify how the application is to be distributed, by specifying the name of the cell and node where the application will be installed and run.
`websphere.server.name`	Name of the server within the WebSphere installation where the application is deployed.

27.2.2.6 Database Connection Properties

In this section, you will specify information that the installer will use to access the Oracle database to store Analytics data. You will also specify JDBC/JNDI information to be placed in the Analytics configuration files. JDBC and JNDI data are mutually exclusive. Only one of them must have its `enabled` property set to `true`. See Table 27–9, Table 27–10, and Table 27–11.

Table 27–9 Database properties in **`analytics-build.properties`**

Property	Description
`install.database`	Set to `true` to run install schema queries.
`db.home.dir`	Database home directory.
`db.sys.user`	System user name.
`db.sys.password`	System user password.
`db.host`	Host address.
`db.port`	Host port number.
`db.sid`	Database SID.

Table 27–10 JDBC database writer properties in **`analytics-build.properties`**

Property	Description
`jdbc.enabled`	`True | false` Set to `true` to configure JDBC settings.
`jdbc.name.value`	Name of the database connection.
`jdbc.default.value`	There must be exactly one connection marked with `default="true"`
`jdbc.type.value`	Type of connection: `jdbc`
`jdbc.classname.value`	JDBC driver class.
`jdbc.url.value`	JDBC URL
`jdbc.user.value`	JDBC attribute. Database user name
`jdbc.password.value`	JDBC attribute; database password

Table 27–11 JNDI properties in ***`analytics-build.properties`***

Property	Description
`jndi.enabled`	`True` | `false` Set to `true` to configure JNDI settings.
`jndi.name.value`	Name of the database connection.
`jndi.default.value`	There must be exactly one connection marked with `default="true"`
`jndi.type.value`	Type of connection: `resource`
`jndi.resource.value`	JNDI attribute. JNDI name

27.2.2.7 Hadoop Properties

In this section, you will provide information about your Hadoop configuration (Table 27–12):

- The base path to the Hadoop installation directory
- Paths to raw data. One path specifies where, on the local file system, raw data will be recorded by the Analytics Sensor (data capture application). The other path specifies where, on the Hadoop distributed file system, the raw data will be written by the HDFS agent.

Table 27–12 Hadoop properties in ***`analytics-build.properties`***

Property	Description
`hadoop.installation.path`	Path to the Hadoop installation directory.
`hadoop.hdfs.defaultfs`	The default path in HDFS for writing Analytics raw data.
`hadoop.tasktracker.url`	URL of the Hadoop task tracker web application.
`hadoop.filesystem.url`	URL of the Hadoop file system web application.
`logwriter.output.path`	Local file system path where the data capture application will record raw data.

27.2.3 Step 3. Preparing to Run the Silent Installer

Before running the silent installer, complete the following steps on all servers where you unzipped the silent installer:

1. Make sure that `JAVA_HOME` and `ANT_HOME` in your environment are set to the correct paths.
2. Start all the Analytics application servers (they must be running for the silent installer to work).
3. If you are using UNIX, issue the following command to ensure that the next command (to run the installer) can be executed:

   ```
   chmod +x analytics_install.sh
   ```

4. If you are installing on WebLogic, run one of the following commands to ensure that the classpath is properly set and the WebLogic deployment task will run correctly:

   – UNIX:

```
<WL_HOME>/server/bin/setWLSEnv.sh
```

- Windows:

```
<WL_HOME>\setWLSEnv.bat
```

27.2.4 Step 4. Installing Analytics

In this step, you will run the silent installer, which will install the Analytics product against the `analytics-build.properties` file, in which you specified:

- Where Analytics will be installed on your file system.
- Which application server you wish to use and where it exists.
- The location of your database.
- The location of your WebCenter Sites.
- Where other associated products or software components are located on your system.

This section contains the following installation scenarios:

- Section 27.2.4.1, "Single-Server Analytics Installation"
- Section 27.2.4.2, "Dual-server Analytics Installation"
- Section 27.2.4.3, "Distributed Analytics Installation"

27.2.4.1 Single-Server Analytics Installation

In this scenario, you will install the entire Analytics product on the same server that runs all Analytics supporting software.

To install Analytics on a single server

1. Run the silent installer:

   - UNIX:

   ```
   ./analytics_install.sh
   ```

   - Windows:

   ```
   analytics_install.bat
   ```

   > **Note:** For descriptions of the components that are installed, see Table 27–13. For information about the events that occur when the silent installer runs, see the next step in this procedure.

2. When the `analytics_install` command begins executing, the installation process begins and the script performs the steps listed in Section 27.1.2, "Silent Installer Actions." When the installation process completes successfully, the following message is displayed:

```
The silent install has finished.
--------------------------------------------------------------
Your installation of Analytics is complete. Please review your
application server documentation and make sure that it is
configured for UTF-8 encoding.
```

```
--------------------------------------------------------------
```

Note: If an error occurs, the installation process terminates and displays the following message:

```
"The install script ended with error code nn. Please consult the
log and check for errors."
```

If an error is reported on the console, inspect the `analytics-install.log` to identify the problem (the `analytics-install.log` file is located in the directory where the silent installer resides). Typically, problems arise when one or more properties have been incorrectly configured. Carefully review and correct any property which is not correct, then rerun the silent installer.

If an error occurs because of failure to find a necessary component, make the appropriate adjustments to your `JAVA_HOME`, or `ANT_HOME` environment settings, and rerun the silent installer.

---

3. When the installer has successfully completed its task, initialize WebCenter Sites to Analytics. Go to Step 5. Configuring WebCenter Sites to Enable Analytics.

27.2.4.2 Dual-server Analytics Installation

In this scenario, you will install the Analytics product on the server that hosts Analytics supporting software, except for the database. The Analytics database is installed on its own server.

To install Analytics on two servers

1. Run the silent installer on the server where you wish to install Analytics:

   – UNIX:

   ```
   ./analytics_install.sh sensor hadoopjobs analytics analyticsadmin cs_
   integration verify_install
   ```

   – Windows:

   ```
   analytics_install.bat sensor hadoopjobs analytics
   analyticsadmin cs_integration verify_install
   ```

   When the `analytics_install` command begins executing, the installation process begins. For information about the events that occur when the silent installer runs, see step 2 in Section 27.2.4.1, "Single-Server Analytics Installation."

2. Run the silent installer on the database server, where SQL Plus is already installed. (Running the silent installer initializes the Oracle database with Analytics schema, as explained in Table 27–13. See the Ant target `database.`):

   - UNIX:

   ```
   ./analytics_install.sh database
   ```

   - Windows:

   ```
   analytics_install.bat database
   ```

3. When the installer has successfully completed its task, initialize WebCenter Sites to Analytics. Go to Step 5. Configuring WebCenter Sites to Enable Analytics.

27.2.4.3 Distributed Analytics Installation

For a distributed installation, you will install different parts of the Analytics product on the servers where you unzipped `analytics2.5.zip`. You will run the `analytics_install` command and specify which part (i.e., Ant target) to install on the given server. If your selected target has dependencies on supporting systems, the silent installer will verify that the supporting systems are installed (and running in the case of the Analytics database). Table 27–13 lists each target that is recognized by the silent installer and its dependency on other systems.

To install Analytics in distributed mode

1. Run the installer on each Analytics server where you wish to install an Analytics component:

   – UNIX:

   ```
   ./analytics_install.sh <Ant_target>
   ```

   – Windows:

   ```
   analytics_install.bat <Ant_target>
   ```

> **Note:** The `<Ant_target>` parameter can be an individual Ant target or a space-separated list of targets, defined in Table 27–13. To install targets one at a time, run the silent installer for each target. If you do not specify a target, all Analytics components (defined in Table 27–13) will be installed on the given server.

Table 27–13 Ant targets and their dependencies

Ant Target	Description	Server on which to Unzip Silent Installer	Dependencies
`sensor`	Installs the data capture web application and HDFS Agent (Oracle-specific).	Data Capture	None
`hadoopjobs`	Installs the Oracle-specific Analytics/Hadoop job scheduler.	Master Node	Oracle
`database`	Updates the Oracle database with Analytics schema. The `database` Ant target loads Analytics-specific database definitions into SQL Plus. SQL Plus then updates the Oracle database with the definitions, and so initializes the database with the schema required by Analytics.	Server on which SQL Plus is installed	Oracle
`analytics`	Installs the Analytics reporting engine web application.	Administration, Reporting	Oracle

Table 27–13 (Cont.) Ant targets and their dependencies

Ant Target	Description	Server on which to Unzip Silent Installer	Dependencies
`analyticsadmin`	Installs the Analytics administration web application.	Administration, Reporting	Oracle
`cs_integration`	Modifies `futuretense_xcel.ini` (located on the WebCenter Sites system. WebCenter Sites does not have to be running in order for the installer to modify `futuretense_xcel.ini`). The silent installer locates `futuretense_xcel.ini` against the `cs.local` property (which you set during the installation process).	WebCenter Sites Host	None
`verify_install`	Provides an object impression used to verify Hadoop Jobs. **Note:** You will use this target when verifying the Analytics installation in Chapter 30, "Verifying Your Analytics Installation."	Data Capture	None

2. When the `analytics_install` command begins executing, the installation process begins. For information about the events that occur when the silent installer runs, see step 2 in Section 27.2.4.1, "Single-Server Analytics Installation."
3. When the installer has successfully completed its task, configure WebCenter Sites to enable Analytics and specify the locations of the Analytics servers. Go to Step 5. Configuring WebCenter Sites to Enable Analytics.

27.2.5 Step 5. Configuring WebCenter Sites to Enable Analytics

Once Analytics is successfully installed, complete the steps below on the WebCenter Sites host.

> **Note:** WebCenter Sites does not have to be running in order for the steps in this section to be completed successfully.

1. Specify the location of the WebCenter Sites `futuretense_xcel.ini` property file, so that it can be modified by the silent installer with Analytics-specific settings.

   Open `analytics-build.properties` and modify the properties described in Table 27–14.

Table 27–14 Properties that update `futuretense_xcel.ini`

Property	Description
`cs.local`	Specifies that WebCenter Sites is local (relative to the silent installer). **Verify that this property is set to `true`.**

Table 27–14 (Cont.) Properties that update `futuretense_xcel.ini`

Property	Description

`src.ini.file`	Specifies the location of the WebCenter Sites `futuretense_xcel.ini` property file.
	Sample values:
	■ Windows:
	`C:/JSK_060809\RunTime\ContentServer/7.5.1/futuretense_xcel.ini`
	■ Linux:
	`/home/fatwire/RunTime/contentServer/7.5.1/futuretense_xcel.ini`

`mod.ini.file`	Specifies the path to WebCenter Sites `futuretense_xcel.ini` file. (The path is identical to the path in the `src.ini.file` property.) Confirms to WebCenter Sites that Analytics is installed. This property modifies `futuretense_xcel.ini` (according to the settings you specify in `analytics-build.properties`), and writes the modified `futuretense_xcel.ini` file to the specified location. The original `futuretense_xcel.ini` file (specified by `src.ini.file`) is automatically backed up. **Sample value:** Use the value that is set in `src.ini.file`.

2. Run the silent installer on the WebCenter Sites host:

   – UNIX:

   ```
   ./analytics_install.sh cs_integration
   ```

   – Windows:

   ```
   analytics_install.bat cs_integration
   ```

3. When the silent installer completes its task, continue to Section 27.3, "Next Step."

27.3 Next Step

- Before verifying the Analytics installation, complete the following steps, as necessary:

  – Inspect the Analytics configuration files that have been updated by the silent installer. Refer to Chapter 28, "Tuning Analytics Configuration Parameters." This chapter also contains instructions for setting up logging for the Hadoop Job scheduler.

  – Optional. Modify the visitor tracking configuration. See Chapter 29, "Configuring Visitor Detection" for instructions.

- To complete the Analytics installation, follow the steps in Chapter 30, "Verifying Your Analytics Installation."

28

Tuning Analytics Configuration Parameters

This chapter describes the configuration files that are updated by the silent installer when it installs Analytics. This chapter also provides guidelines for setting properties in the configuration files. The files are:

- Section 28.2, "global.xml"
- Section 28.3, "log4j.properties"
- Section 28.4, "futuretense_xcel.ini"

28.1 Overview of Configuration Files

When installing Analytics, the silent installer sets the values of various properties to match the values that you specified in `analytics-build.properties`. The properties are stored in the following files:

- global.xml
- log4j.properties
- futuretense_xcel.ini

Once Analytics is installed, you can tune various properties directly in their respective files, as necessary. Guidelines are available in the rest of this chapter.

Caution: Installation directories (and related configurations) created by the silent installer must not be moved, renamed, or otherwise modified in any way.

28.2 global.xml

The Analytics silent installer modifies the `global.xml` file on each server where the file is stored and where the installer is executed. Once Analytics is installed, you can customize `global.xml` directly on its respective hosts.The customizable sections are:

- Parameters Within <params> </params>
- Database Connection Parameters
- LFS Logwriter Implementation Parameters

Note: Properties that must not be reset are shaded in the tables below. Each application has its own `global.xml` file, located in the application's classpath.

28.2.1 Parameters Within <params> </params>

This section contains properties that define Analytics installation directories, system administrators' contact information, the handling of raw data, and data processing conditions.

> **Caution:** In Table 28–1, some parameters have an asterisk (*) suffixed to indicate that these parameters MUST NOT be reset.

Table 28–1 Analytics parameters in `global.xml`

Parameter	Description	Sample Value/ Format	Host

`swchart_instdir*`	Absolute path to the installation directory of the Swiff Chart Generator.	**UNIX:**
		`/usr1/software/SwiffChart`
		Windows:
		`C:/Program Files/GlobFX /Swiff Chart Generator3`
			None

`engine_instdir*`	Absolute path to the directory into which the Analytics application was deployed.	**UNIX:**
		`/usr/share/tomcat/analytics`
		Windows:
		`C:/CS/tomcat5/webapps /analytics`
			None

`href_cs*`	Context URL of the WebCenter Sites interface.	`http://analytics .yourcompany.com/analytics`	None

`report_instdir*`	Absolute path to the deployed report configuration `XML` files.	**UNIX:**
		`/data/analytics/reports`
		Windows:
		`C:/CS/reports`
			None

`forgotpassword`	Email address for the Analytics administrator who is responsible for password recovery.	`admin@yourcompany.com`	Reporting Node
`noaccount`	Email address of the Analytics administrator who is responsible for creating accounts.	`admin@yourcompany.com`	Reporting Node
`href_reporting*`	URL of the Analytics Reporting application. This URL is required to link the Analytics Reporting application to the Analytics Administrator application.	`../analytics/`	None
`href_admin*`	URL of the Analytics Administrator application. This URL is required to link the Analytics Administrator application to the Analytics Reporting application.	`../analyticsadmin/`	None

Table 28–1 (Cont.) Analytics parameters in `global.xml`

Parameter	Description	Sample Value/ Format	Host
href_help*	URL at which help relating to Analytics can be obtained.	http://www.oracle.com/technetwork/middleware/webcenter/sites/overview/index.html	None
encoding*	Character encoding to be used for decoding request parameters and encoding response parameters. This encoding should match the encoding of the application server	utf8	None
hadoop.hdfs.defaultfs*	Location of the root directory under which raw data, output, and cache files are stored on the Hadoop file system.	hdfs://<hostname>:<port>/analytics where: <hostname> is the name of the master node <port> is the NameNode port specified in the fs.default.name configuration parameter in hadoop-site.xml.	None
hadoop.local.cachedir*	Local path of the folder that stores the fileEnvObjects at job startup.	/usr/local/cache/	None
analytics.filtercurrentdata	Flag that specifies whether to skip processing data for the current day, week, month, and year. Set this property to true if wish to enable any of the properties listed in the row below. (The properties are: analytics.filtercurrentXXX)	**Default value**: false	Master Node
analytics.filtercurrent*XXX* (where XXX stands for day, week, month, or year)	Flag that specifies whether to skip processing data for the current day/week/month/year. Processing for each day/week/month/year can be set individually by adding the following parameters and setting them to false: analytics.filtercurrentday analytics.filtercurrentweek analytics.filtercurrentmonth analytics.filtercurrentyear **Note**: If you add one or more of the above parameters, make sure that analytics.filtercurrentdata (listed above) is set to true.	analytics.filtercurrentday=false	Master Node
admin.context*	URL of the Analytics Administrator application. Add a trailing slash (/).	http://<hostname>:<port>/analyticsadmin/ where: <hostname> is the name of the Admin Server <port> is the port of the application server	None

Table 28–1 (Cont.) Analytics parameters in global.xml

Parameter	Description	Sample Value/ Format	Host
sensor.context*	URL of the sensor application. Add a trailing slash.	http://<hostname>:<port>/sensor/ where: <hostname> is the name of the Data Capture server <port> is the port of the application server	None
monitoring.registry.port*	Port on which to start the RMI service.	11199	None
href.hadoop.tasktracker*	URL to the Hadoop TaskTracker admin interface.	http://<hostname>:50030/ where: <hostname> is the name of the master node	
href.hadoop.filesystem.browser*	URL to the hdfs file system browser interface.	http://<hostname>:50070/ where: <hostname> is the name of the master node	
importer.sleeptime	Time interval (in minutes) after which hdfsagent will look for raw data to be copied from the local file system to hdfs.	10	Data Capture
NumberOfProcessorThreads	Number of jobs that run simultaneously. Each job is divided into tasks by Hadoop. For high-volume systems, NumberOfProcessorThreads should be set to 1, so that only a single job can run at a given time. For low-volume systems or in demonstration scenarios, the value can be greater (3, 4, or 5).	1	Master Node
sensor.thresholdtime	Time interval (in minutes) after which the sensor will rotate the data.txt.tmp file (where incoming raw data is first written) to data.txt. Set a time interval such that no more than 5 to 10GB of data will be processed during the interval. If this parameter is omitted, the default threshold time (10 min) is used.	240	Data Capture
session.rotate.delay	Interval of time (in minutes) after midnight that raw session data is kept open. The default is 360 minutes. This means that session data will be moved to HDFS 6 hours after midnight; session processing will then start.	360	Data Capture

Table 28–1 (Cont.) Analytics parameters in `global.xml`

Parameter	Description	Sample Value/ Format	Host
`scheduler. checkinterval`	Specifies the frequency at which the scheduler will create new Hadoop jobs for fresh data. The default value is 15 minutes.	`15`	Master Node
`midnight.offset`	Allows the system to derive relative midnight used for file rotation. Relative midnight and `session.rotate.` `delay` determine when the daily cycle for capturing session data ends.	**Default value:** `0` **Format**: minutes	Data Capture
`cs_enabled`	Specifies whether buttons for navigating to the WebCenter Sites interface are enabled or disabled in the Analytics interface.	**Default value:** `true`	Reporting Application
`archive.enabled`	Specifies whether HDFS Agent archiving of raw data files is enabled. When this property is set to `true`, HDFS Agent will automatically create archives of raw analytics data on a periodic basis. The archive directory and start time are specified in the following properties: `archive.output.dir` and `archive.start.time`	**Default value:** `false`	Data Capture
`archive.output.dir`	Path to the directory for storing archived data files. Must be a valid URI.	**Format**: directory path	Data Capture
`archive.start. time`	Start time for archiving. The archiving task will start at `HH:mm` on a daily basis.	**Default value:** `06:00` **Format:** `HH:mm` in 24-hour time format. `HH` ranges from `00-23`; `mm` ranges from `00-59`.	Data Capture
`purgejobs.enabled`	When this property is set to `true` the system will automatically schedule cleanup jobs to remove subfolders and files after they have been successfully processed.	**Default value:** `false`	Master Node
`notification. enabled`	Indicates whether email notifications are enabled. Email notifications are sent when the availability of Analytics services changes.	**Default value:** `false`	Admin Node
`mail.from`	Email address from which notifications are sent.	**Format:** Email address	Admin Node

Table 28–1 (Cont.) Analytics parameters in `global.xml`

Parameter	Description	Sample Value/ Format	Host
`sensor.requestqueue.maxsize`	Specifies `CRITICAL` condition for the Analytics Sensor. This property specifies a threshold value that triggers a `CRITICAL` (red) condition when the sensor cannot respond quickly enough to the amount of raw data that it needs to record. When the threshold is reached or exceeded, the **Analytics Sensor** component is displayed in red. The threshold value for this property is expressed as an object impression, i.e., a single invocation of the `sensor` servlet. (The **Analytics Sensor** component is represented in the **Overview** panel of the **Components tab** of the Analytics Administration interface.)	**Default value:** `10000`	Admin Node
`sensor.requestqueue.warnsize`	Specifies `WARNING` condition for the Analytics Sensor. This property specifies a threshold value that triggers a `WARNING` (yellow) condition when the sensor cannot respond quickly enough to the amount of raw data that it needs to record. When the threshold is reached or exceeded, the **Analytics Sensor** component is displayed in yellow. The threshold value for this property is expressed as an object impression, i.e., a single invocation of the `sensor` servlet. (The **Analytics Sensor** component is represented in the **Overview** panel of the **Components tab** of the Analytics Administration interface.)	**Default value:** `3000`	Admin Node

Table 28–1 (Cont.) Analytics parameters in `global.xml`

Parameter	Description	Sample Value/ Format	Host

`ipBlacklist`	Stores IP addresses whose records will be ignored during the normalization process. This parameter's value is a comma-separated list of IP addresses (and/or fragments) to which the requesting IP address is compared. If the requesting address matches any of the listed values, it is dropped.
	■ If you wish to restrict specified IP addresses from accessing all sites, enclose the `ipBlacklist` parameter within the following element:
	`<params host="default"`
	`...`
	`</params>`
	■ If you wish to restrict specified IP addresses from accessing specific sites, then for each site enclose the `ipBlacklist` parameter within the following element:
	`<params host="siteName">`
	`...`
	`</params>`
		Format:
		`<param type="string"`
		`name="ipBlacklist"`
		`value="ip_address-1,`
		` ip_address-2, ..."/>`
		■ **Sample IP address blacklist for all sites:**
		`<params host="default"`
		`<param type="string"`
		`name="ipBlacklist"`
		`value="123.45.67.89,`
		`        192.168.1.4"`
		`/>`
		`</params>`
		■ **Sample IP address blacklist for a** specific site:
		`<params host=`
		`"CompanySite.com">`
		`<paramtype="ipBlacklist`
		`"value="123.45.67.89,`
		`192.168.90.5"/>`
		`</params>`
			Data Capture

28.2.1.1 Database Connection Parameters

Typically, users require only one database connection. Custom reporting may require multiple connections. If you need to define your own JDBC resources or reference the existing JDBC connections via JNDI, use the following tag:

```
<connection
    name="<connection_name>"
    default="true"
    type="<jdbc_or_resource>"
    classname="<database_driver_classname>"
    url="<database_url>"
    user="<database user name>"
    password="<database password>" />
```

Table 28–2 describes database connection parameters.

Table 28–2 Database Connection Parameters

Parameter	Description
`name`	Name of the connection. Example: `localhostDB`
`default`	There must be exactly one connection marked with `default="true"`
`type`	Type of connection: `jdbc` (JDBC) or `resource` (JNDI)
`resourcename`	JNDI attribute; JNDI name **Note:** Used only if `type` is set to `resource`

Table 28–2 (Cont.) Database Connection Parameters

Parameter	Description
classname	JDBC driver class. **Example:** oracle.jdbc.driver.OracleDriver
url	JDBC URL
user	JDBC attribute; database user name
password	JDBC attribute; database password

Example 28–1 Example JDBC:

```
<connection
    name="jdbcsample"
    default="true"
    type="jdbc"
    classname="oracle.jdbc.driver.OracleDriver"
    url="jdbc:oracle:thin:@dbserver:1521:sid"
    user="analytics"
    password="analytics" />
```

Example 28–2 Example JNDI:

```
<connection
    name="conn1"
    default="false"
    type="resource"
    resourcename="java:comp/env/jdbc/tadev" />
```

28.2.2 LFS Logwriter Implementation Parameters

The LFS logwriter implementation writes incoming raw data to the local file system. If you wish to change the root path (the location to which raw data will be written), use the following tag:

```
<logwriters>
    <logwriter type="LFS" name="LFS" rootpath="C:/analytics/
      sensorlocal" />
</logwriters>
```

Caution: In Table 28–3, some parameters have asterisk (*) suffixed to indicate that these parameters MUST NOT be reset.

Table 28–3 Logwriter Parameters

Parameter	Description
type*	Type of logwriter. **Legal value:** LFS
name*	Alias name of the logwriter. **Legal value:** LFS

Table 28–3 (Cont.) Logwriter Parameters

Parameter	Description

`rootpath`	Location, on the local file system, to which raw data will be written.
	Examples:
	UNIX:
	`rootpath="/analytics/sensor"`
	Windows:
	`rootpath="c:/analytics/sensor"`

28.3 log4j.properties

The silent installer modifies the `log4j.properties` file on each server where the file is stored and the installer is executed. Once the Analytics installation is complete, you can customize properties directly in all or selected log4j.properties files, as shown in Table 28–4. (The `log4j.properties` file is located in `<HADOOP_HOME>/conf`.)

Table 28–4 Parameters in `log4j.properties`

Property	Description	Example / Format

`log4j.rootLogger`	Specify the log level and the appender of the root logger. Multiple appenders can be specified, separated by commas.	`log4j.rootLogger=INFO, DaRoFiAppender`
		- or -
		`log4j.rootLogger=INFO, DaRoFiAppender, ConsoleAppender`

`log4j.category. com.fatwire. analytics`	Specify the log level. The following explains the log levels in decreasing order of severity: ■ `FATAL` – Severe errors that cause premature termination. ■ `ERROR` – Runtime errors, or unexpected conditions. ■ `WARN` – Other runtime situations that are undesirable or unexpected, but not necessarily "wrong." ■ `INFO` – Provides informative messages about the workflow and status of the application. ■ `DEBUG` – Various kinds of debug information. ■ `TRACE` – All logging information Note: In production mode, this property should be set to `WARN`.	`INFO`
`log4j.appender. DaRoFiAppender`	Specify the appenders to be used for logging.	`org.apache.log4j. DailyRollingFileAppender`

Table 28–4 (Cont.) Parameters in log4j.properties

Property	Description	Example / Format
log4j.appender.DaRoFiAppender.datePattern	Specify the date pattern in the following format:	'.'yyyy-MM-dd
log4j.appender.DaRoFiAppender.file	Specify the location of the log file, along with the name of the log file.	../logs/xxx.log
log4j.appender.DaRoFiAppender.layout	Specify the layout.	org.apache.log4j.PatternLayout
log4j.appender.DaRoFiAppender.layout.ConversionPattern	Specify the layout pattern.	%d{ISO8601}%- 5p[%t]%c:%m%n

28.3.1 Setting Up Logging for the Hadoop Job Scheduler

Edit log4j.properties file by adding the following parameters:

```
hadoop.root.logger=WARN,console, DRFA
hadoop.log.file=hadoop.log
log4j.rootLogger=${hadoop.root.logger}, DRFA, EventCounter

#
# Daily Rolling File Appender
#
log4j.appender.DRFA=org.apache.log4j.DailyRollingFileAppender
log4j.appender.DRFA.File=${hadoop.log.dir}/${hadoop.log.file}

# Rollver at midnight
log4j.appender.DRFA.DatePattern=.yyyy-MM-dd

# 30-day backup
log4j.appender.DRFA.MaxBackupIndex=30
log4j.appender.DRFA.layout=org.apache.log4j.PatternLayout

# Pattern format: Date LogLevel LoggerName LogMessage
log4j.appender.DRFA.layout.ConversionPattern=%d{ISO8601} %p %c: %m%n
```

28.4 futuretense_xcel.ini

The silent installer modifies the futuretense_xcel.ini file of the WebCenter Sites application that resides on the machine where the silent installer is running. The purpose of modifying the file is to specify the location of the Analytics application and the authorized user.

Caution: In Table 28–5, some parameters have asterisk (*) suffixed to indicate that these parameters MUST NOT be reset.

Table 28–5 Analytics Properties in `futuretense_xcel.ini`

Property	Description	Example
`analytics.datacaptureurl*`	URL where the Analytics data capture servlet (`sensor` servlet) is running	`http://<ipaddress>:<port>/sensor/statistic`
`analytics.enabled`	Indicates whether Analytics is available. **Note:** If set to `false`, this property disables data capture.	`true`
`analytics.piurl*`	URL where the Analytics performance indicator servlet is running. For information about the performance indicator, see the "Integrating Oracle WebCenter Sites: Analytics with Oracle Web Center Sites" chapter in the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide*.	`http://<ipaddress>:<port>/analytics/PI`
`analytics.reporturl*`	URL where the generated report is displayed.	`http://<ipaddress>:<port>/analytics/Report.do`
`analytics.user*`	Pre-configured Analytics user who logs in to Analytics from WebCenter Sites.	`csuser` Default in Analytics. Changing the name is not recommended.

29

Configuring Visitor Detection

This chapter contains the following sections:

- Section 29.1, "Overview"
- Section 29.2, "Methods"

29.1 Overview

Visitor tracking is done by the data capture application (sensor). Visitors can be tracked in the following ways:

- Self-organized method, which supports tracking of new visitors
- `Sessionfingerprint` method
- Cookie method

Note: By default, the `Sessionfingerprint` method is enabled on all sites.

29.2 Methods

Methods for tracking visitors are implemented by adding visitor tracking code to `global.xml` (and to the `img` tag, depending on the tracking method).

Note: When changing the visitor tracking method, you must also update the `global.xml` configuration file for each sensor instance. The location of `global.xml` depends on the application server you are using.

When modifying parameters in the `global.xml file(s),` stop the Analytics Sensor web application, modify its `global.xml` file, then start the sensor to effect the change.

29.2.1 Self-Organized Visitor Tracking

The self-organized visitor tracking method uses the value of the `cookieid` parameter appended to the image request.

To track the visitor

1. Add the following parameters to the `img` tag:

   `cookieid` – the visitor ID

newvisitor – determines whether the user has already visited the site

2. If you need to change the name of the cookieid parameter (to visitorid for example), you must also do the following:

   a. Change the following line in global.xml as shown below:

      From:

      ```
      <alias name="cookieid" for="cookieid"/>
      ```

      To:

      ```
      <alias name="visitorid" for="cookieid"/>
      ```

   b. Pass the visitorid and newvisitor parameters to the img tag.

29.2.2 Sessionfingerprint Method

The Sessionfingerprint method identifies each visitor by a combination of the IP address, screen resolution, and agent string to assign sessions to visitors. You can enable Sessionfingerprint for all sites or a selected site. Sessionfingerprint is enabled for all sites by default when Analytics is first installed.

> **Note:** The Sessionfingerprint method does not support *new* visitor tracking.

To enable the sessionfingerprint method

- To enable visitor tracking on all sites, add the lines in bold type below to global.xml (nest them in the default <params> ... </params> tag, as shown):

  ```
  <params host="default">
  <param type="string" name="sessionIdGenerator"
      value="AppServerId"/>
  <param type="string" name="visitorIdGenerator"
      value="SessionfingerprintId"/>
  ...
  </params>
  ```

- To enable visitor tracking on a selected site, add the lines in bold type below to global.xml (nest them in the <root> tag, as shown):

  ```
  <root>
  <params site="sitename">
  <param type="string" name="sessionIdGenerator"
      value="AppServerId"/>
  <param type="string" name="visitorIdGenerator"
      value="SessionfingerprintId"/>
  </params>

  <params host="default"> ...
  ```

Table 29–1 describes parameters for Sessionfingerprint visitor tracking.

Table 29–1 Parameters for `Sessionfingerprint` visitor tracking

Parameter	Description
`sessionIdGenerator`	Specifies the ID generator that is used to identify sessions. **Caution:** The default value is `AppServerID`. Do not modify this value. It is a reference to the object that generates the ID.
`visitorIdGenerator`	Specifies the ID generator that is used to identify a visitor to the site. **Required value for this method:** `SessionfingerprintId`
`site`	Name of the site that is passed via the `img` tag

29.2.3 Cookie Method

Using the cookie method, you can enable visitor tracking across all sites or within a selected site.

To enable the cookie method

- **To enable visitor tracking on all sites**, add the lines in bold type below to `global.xml` (nest them in the default `<params> ... </params>` tag, as shown):

> **Note:** When using the code below, replace the sample values with values of your own. For parameter definitions, see Section 29–2, " Parameters for cookie-based visitor tracking."

```
<params host="default">
<param type="string" name="visitorIdGenerator"
    value="CookieId"/>
<param type="string" name="visitorIdCookieName" value="visid"/>
<param type="string" name="visitorIdCookieMaxAge"
    value="31536000"/>
<param type="string" name="visitorIdCookieDomain" value="firstsiteii"/>
<param type="string" name="visitorIdCookiePath" value="/"/>
<param type="string" name="visitorIdCookieSecure"
    value="false"/>
...
</params>
```

- **To enable visitor tracking on a selected site**, add the lines in bold type below to `global.xml` (nest them in the `<root>` tag, as shown):

> **Note:** When using the code below, replace the sample values with values of your own. For parameter definitions, see Section 29–2, " Parameters for cookie-based visitor tracking."

```
<root>
<params site="FirstSiteII">
    <param type="string" name="sessionIdGenerator" value="AppServerId"/>
    <param type="string" name="visitorIdGenerator" value="CookieId"/>
    <param type="string" name="visitorIdCookieName" value="visid"/>
    <param type="string" name="visitorIdCookieMaxAge" value="31536000"/>
    <param type="string" name="visitorIdCookieDomain" value=".firstsiteii.at"/>
```

```
        <param type="string" name="visitorIdCookiePath" value="/"/>
        <param type="string" name="visitorIdCookieSecure" value="false"/>
    </params>
    <params host="default">
    ...
```

Table 29–2 describes parameters for cookie-based visitor tracking.

Table 29–2 Parameters for cookie-based visitor tracking

Parameter	Description
sessionIdGenerator	Specifies the ID generator that is used to identify sessions. **Caution:** The default value is AppServerID. Do not modify this value. It is a reference to the object that generates the ID.
visitorIdGenerator	Specifies the ID generator that is used to identify a visitor to the site. **Required value for this method:** CookieId
visitorIdCookieName	Name of the cookie set on the client side. **Example:** visid
visitorIdCookieMaxAge	Age of the cookie in milliseconds. After this time, the cookie is invalid. (The value that you set is added to the current time.) **Example:** 31536000
visitorIdCookieDomain	Domain on which the cookie should be set. Example: firstsiteii **Note**: If you wish to set visitorIdCookieDomain on your local host, do not use the localhost value. It is not accepted by most browsers. Instead, add the following type of entry to your hosts file: **Example:** 127.0.0.1 firstsiteii.at and change the analytics.datacaptureurl in futuretense_xcel.ini to: http\://firstsiteii.at\:8081/sensor/statistic
visitorIdCookiePath	Sub-path on which the cookie should be set. **Example:** "/"
visitorIdCookieSecure	Specifies that the cookie will be sent via a secure channel, such as an SSL connection. **Do not change the default value (false)**.
site	Name of the site that is passed via the img tag. **Example:** FirstSite II

Note:

1. Regarding the JSESSIONID cookie, this is managed by your application server's settings, application server administrator will enable secure cookies. For more information on Weblogic Server regarding enabling secure cookies see, *Administration Guide for Oracle Identity Manager* section.
2. For the CASTGC cookie, it can be made secure by setting 'p:cookieSecure=true' in <deployed>/cas/WEB-INF/spring-configuration/ticketGrantingTicketCookieGenerator.xml. During the Sites installation with HTTPS and using the HTTPS secure certificates as well as URLs the p:cookieSecure=true would be set. Once you have set your application server to use secure cookies you can verify that this flag is set to true in order to enable the secure attribute on the CASTGC cookie.

30

Verifying Your Analytics Installation

This chapter contains procedures for verifying the operation of your newly installed Analytics application and its components.

This chapter contains the following sections:

- Section 30.1, "Verification Steps"
- Section 30.2, "Recommended Configurations"
- Section 30.3, "Next Step"

30.1 Verification Steps

In this section, you will start Hadoop, the HDFS Agent, and Hadoop Jobs and then complete the following steps:

- Verify that data capture works:
- Verify the HDFS Agent:
- Verify Hadoop Jobs:
- Verify that the Analytics reporting interface can be displayed:
- Verify that the Analytics administrator interface can be displayed:

To verify your Analytics installation

1. Start Hadoop.
2. Start the HDFS Agent (execute `run.sh`, located in the `bin` folder):

   ```
   cd ${ANALYTICS_HOME}/hdfsagent/bin
   chmod +x run.sh
   ./run.sh
   ```

   where `ANALYTICS_HOME` is the Analytics installation directory.
3. Before starting Hadoop Jobs, do the following to ensure uninterrupted operation of Hadoop Jobs:

   a. Edit the file `/etc/security/limits.conf`

   b. Change the value of `soft nofile` and `hard nofile` to `65536`.

   c. Open a new shell where Hadoop will be run. Make sure `ulimit -n` is at least `1024`.
4. Start Hadoop Jobs (execute `run.sh`, located in the `bin` folder):

Note: The JVM must have at least 512MB of memory in order for the Hadoop Job scheduler to run when Hadoop starts. If memory is insufficient, add the memory parameter `-Xmx512m` to the `run.sh` file in Hadoop Jobs.

As the Hadoop Job scheduler is the master of the HDFS, all HDFS Agents must be restarted when you restart the scheduler.

```
cd ${ANALYTICS_HOME}/hadoop-jobs/bin
chmod +x run.sh
./run.sh
```

5. Verify that data capture works:

   a. Make sure that the application server is running on the server where the Analytics Sensor is installed.

   b. Run the `verify_install` utility to access the URL of the Analytics Sensor. On a multi-server installation, run the `verify_install` utility on all the data capture servers. (The `verify_install` utility contains a sample object impression for testing Hadoop Jobs.)

      – UNIX:

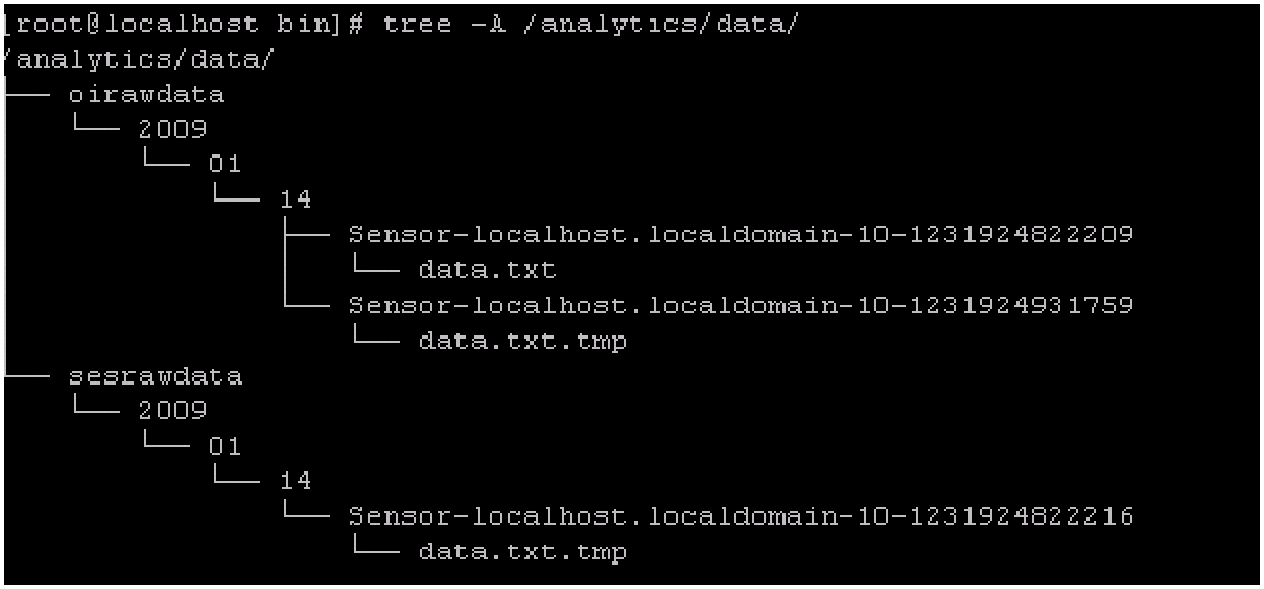
      ```
      ./analytics_install.sh verify_install
      ```

      – Windows:

      ```
      analytics_install.bat verify_install
      ```

      The system returns a preview thumbnail (as a 1x1-pixel `gif` file). The incoming raw data is written by the Analytics Sensor to the `data.txt.tmp` file (see Figure 30–1).

Figure 30–1 Storage of Captured Data



```
[root@localhost bin]# tree -A /analytics/data/
/analytics/data/
├── oirawdata
│   └── 2009
│       └── 01
│           └── 14
│               ├── Sensor-localhost.localdomain-10-1231924822209
│               │   └── data.txt
│               └── Sensor-localhost.localdomain-10-1231924931759
│                   └── data.txt.tmp
└── sesrawdata
    └── 2009
        └── 01
            └── 14
                └── Sensor-localhost.localdomain-10-1231924822216
                    └── data.txt.tmp
```

The `data.txt.tmp` file is stored in the local file system, in a folder. The folder is specified in the `logwriter` element (`rootpath` attribute) in `global.xml` (see

Table 28–3). The sensor will rotate the data.txt.tmp file to data.txt when either the threshold interval is reached (see the sensor.threshold property in Table 28–1), or the application server is restarted.

For example, if the folder is specified as <logwriter type="LFS" name="LFS" rootpath="C:/analytics/data" />, then the raw captured data will be written to the folder structure shown in Figure 30–1, starting with the /analytics/data/ folder. The raw data is collected into data.txt.tmp and, after rotation, stored in data.txt. The folders named oirawdata and sesrawdata are system defaults.

6. Verify the HDFS Agent:

   a. Make sure that the HDFS Agent has successfully copied data.txt (and its directory structure) from the local file system to HDFS. The HDFS Agent handles data.txt as follows:

      – Copies the data.txt file from the local file system to a folder in HDFS. The HDFS folder is specified by the hadoop.hdfs.defaultfs parameter in global.xml (see Table 28–1).

      – When data.txt is successfully copied, it is removed from the local file system.

   b. To verify the actions of the HDFS Agent and inspect the file system, open the Hadoop HDFS web interface, running on your master node (http://<hostname_MasterNode>:50070/) and browse the folder.

7. Verify Hadoop Jobs:

   Hadoop Jobs processes the data that is copied by HDFS Agent from the local file system to HDFS. To view the status of the jobs, open the JobTracker web interface URL:

   ```
   http://<hostname_MasterNode>:50030/
   ```

   In the JobTracker web interface, the "RUNNING Jobs" section displays the OIProcessor jobs. After the OIProcessor jobs are completed, OIInjection job will start and will insert the data into the Analytics database.

   > **Note:** A series of session and visitor jobs will also run. When the jobs are completed, their status will be reported in the "Status Summary" panel of the Analytics Administrator application. You will verify the jobs in 9, when you log in to the Administrator application.

8. Verify that the Analytics reporting interface can be displayed:

   Log in to the Analytics reporting application at the following URL, with user name csuser and password csuser:

   ```
   http://<hostname:port>/analytics
   ```

   where <hostname> is the host name of the server on which the reporting application is installed.

> **Note:** The `csuser` (password `csuser`) is a default that ships with Analytics. This user has administrative rights in Analytics. Log in as `csuser/csuser` when you need to work in the Analytics administration interface (for example, in the Integrating Oracle WebCenter Sites: Analytics with Oracle Web Center Sites chapter in the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide* you will integrate Analytics with WebCenter Sites).
>
> If the database connection fails, the system will display a message denoting that fact.

9. Verify that the Analytics administrator interface can be displayed:

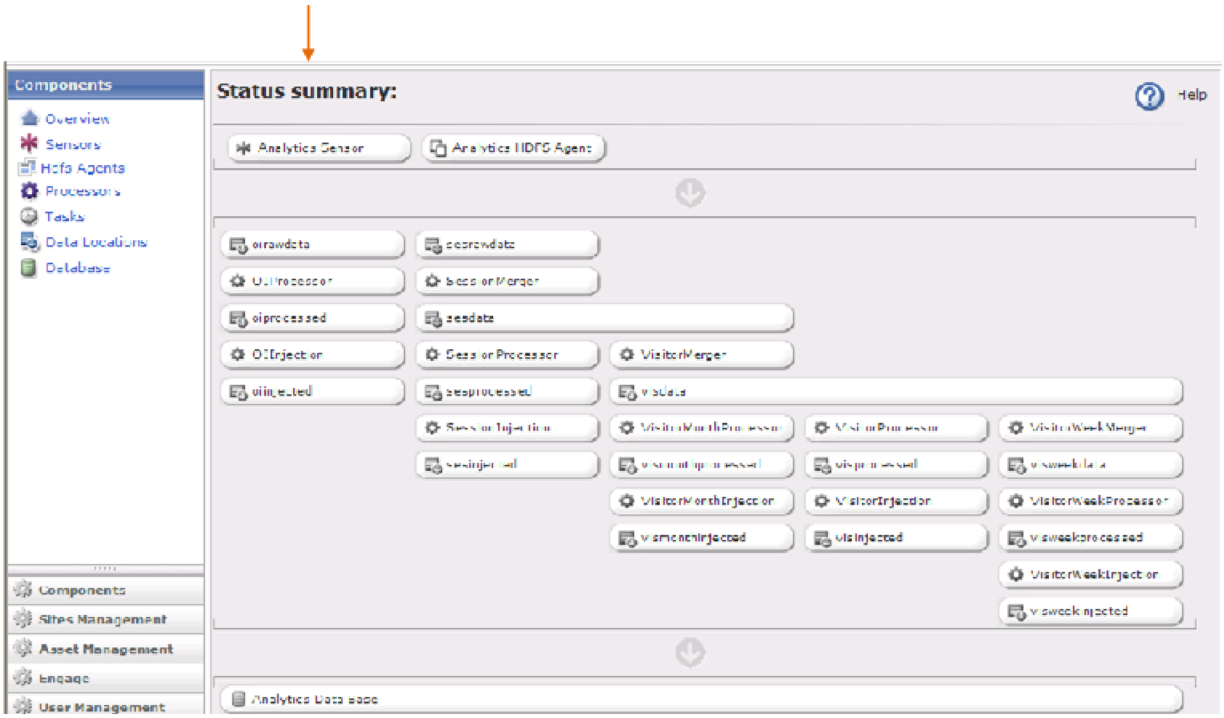
   Log in to the Analytics Administrator application at the following URL, with user name `csuser` and password `csuser`:

   ```
   http://<hostname:port>/analyticsadmin
   ```

   where `<hostname>` is the host name of the server on which the administrator application is installed.

   The opening screen (Figure 30–2) displays the "Status Summary" panel of locations and processors.

Figure 30–2 Analytics Administrator Interface (In the Components tab, the Overview option)



Status Summary: Locations and Processors

Here, you can monitor the status of various components. For example:

- You can monitor the Analytics Sensor to ensure that it is functioning and responding properly to site traffic. (For more information about monitoring the Sensor and responding to alerts, see Sensor Overload Alerts.)

- You can stop the Analytics Sensor.
- You can verify Hadoop Jobs by clicking on a location or processor to view its status. (For more information about Hadoop Jobs and the "Status Summary" panel, see the chapter "Reference: Hadoop Jobs Processors and Locations" in the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide*.

30.2 Recommended Configurations

- Installer Files
- Sensor Overload Alerts
- Geolocation Database

30.2.1 Installer Files

When you have verified your Analytics installation, keep the installer files on their respective hosts. Should you need to modify operating parameters at a later time, you can rerun the installer. **Modifying operating parameters manually is not recommended.**

Installer files are the following:

- `lib` is the folder that contains all the additional `jar` files needed to support the silent installer when it runs.
- `analytics-build.properties` contains all the properties that define how the Analytics product should be installed on your system.
- `analytics-build.xml` is the Ant build script to perform the silent installation.
- `analytics_install.bat` is the batch file that runs the silent installer in a Windows command window.
- `analytics_install.sh` is the UNIX/Linux shell script that runs the silent installer.
- `analytics-silent-install.jar` contains all the custom Java classes that are required by the silent installer.
- `log4j.properties` is used to configure logging behavior such as output target, type and level of message, and the format of messages at runtime.

30.2.2 Sensor Overload Alerts

During heavy site traffic, the Analytics Sensor can become overloaded with incoming data and stop responding normally. The Analytics Sensor will stop writing to the file system and will instead store incoming data in memory, until an out-of-memory condition is reached.

The Analytics Administrator interface (Figure 30–2) alerts you to an "overload" condition by displaying the **Analytics Sensor** button in either yellow or red. Yellow indicates a `WARNING` condition. Red indicates a `CRITICAL` condition (assuming the sensor is running).

> **Note:** A stopped or non-functional Analytics Sensor is also displayed in red.

30.2.2.1 Setting an 'Overload Alert' Threshold

Properties in `global.xml` determine the threshold that triggers a `WARNING` or `CRITICAL` condition. The properties are `sensor.requestqueue.warnsize` and `sensor.requestqueue.maxsize`. Set these properties to a threshold that is compatible with the configuration of your Analytics installation and the volume of site traffic. For more information about these properties, see Table 28–1.

30.2.2.2 Responding to a "red" Condition

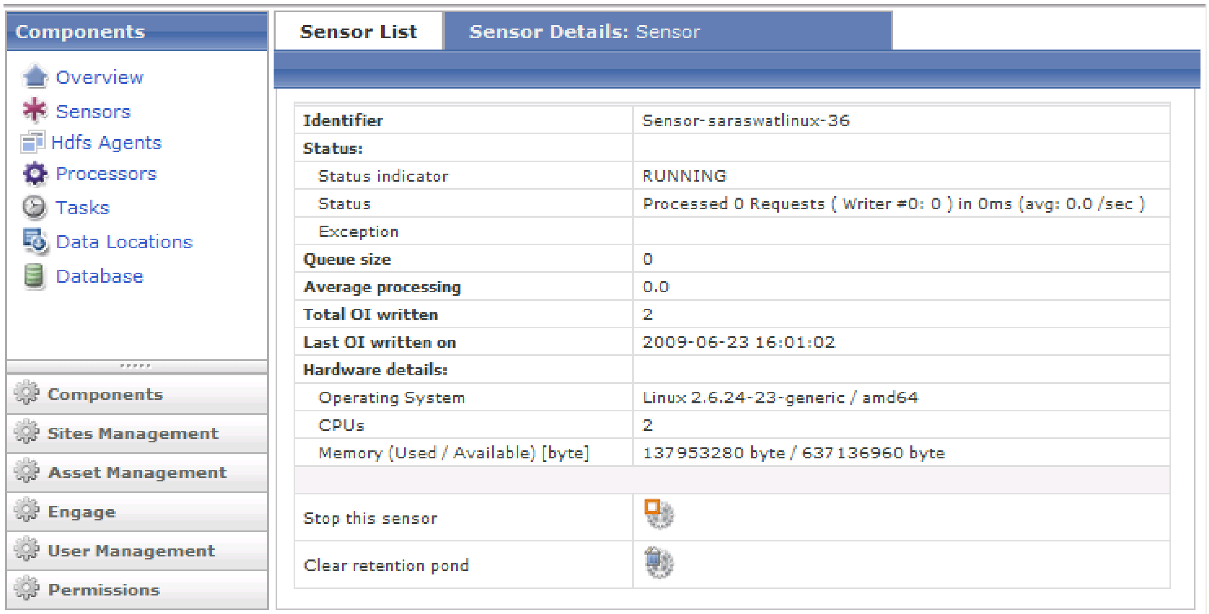
If you are monitoring the "Status Summary" panel (in the Administration interface, Figure 30–2) and you notice that the **Analytics Sensor** button is displayed in red, you need to determine whether the Sensor has stopped, has failed, or is overloaded. In case of overload, you will need to clear the memory in order to reset the system and resume normal functioning. **Data cleared from memory cannot be retrieved and will be lost.**

To respond to a "red" condition

Click the **Analytics Sensor** button and note the main panel.

- If you see `"No data available"`, the Analytics Sensor has either stopped or failed.
- If you see the "Sensor Details" panel, the Analytics Sensor is running, but it is overloaded. Click the icon labeled "Clear retention pond" to clear the memory (Figure 30–3).

Figure 30–3 Retention Pond



Identifier	Sensor-saraswatlinux-36
Status:	
Status indicator	RUNNING
Status	Processed 0 Requests (Writer #0: 0) in 0ms (avg: 0.0 /sec)
Exception	
Queue size	0
Average processing	0.0
Total OI written	2
Last OI written on	2009-06-23 16:01:02
Hardware details:	
Operating System	Linux 2.6.24-23-generic / amd64
CPUs	2
Memory (Used / Available) [byte]	137953280 byte / 637136960 byte
Stop this sensor	
Clear retention pond	

30.2.3 Geolocation Database

GeoLite City is a highly optimized geolocation database provided by MaxMind. GeoLite City is in binary format for performing fast lookups. It is used by Analytics for the information it contains: country, region, area code, metro code, city, and postal code. The GeoLite City database is updated monthly, at the beginning of each month, by MaxMind.

To install the latest GeoLite City database

1. Download the GeoLite City database in binary format from the following location:

   http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz

2. Uncompress the file:

   – On Windows: Use the winzip or zip program to unzip the file.

   – On UNIX: Use the `tar` command to uncompress the file:

   ```
   tar xvzf GeoLiteCity.dat.gz
   ```

3. Copy the `GeoLiteCity.dat` file to the `CACHE` folder under the analytics installation folder on the server where the Hadoop Jobs application is installed:

   On UNIX:

   ```
   cp GeoLiteCity.dat <ANALYTICS_INSTALL_DIR>/CACHE
   ```

30.3 Next Step

If you have successfully completed all the steps in this chapter, then Analytics is ready to be integrated with WebCenter Sites and enabled for data capture as well as report generation.

Integrating and enabling Analytics is a collaborative process among administrators and developers.

- An administrator registers pages and other asset types with Analytics to enable their recognition and inclusion in reports. An administrator also grants users permissions to generate reports.
- Developers ensure that reports are filled with data by enabling data capture on published pages and verifying that the set of default Hadoop Jobs processes raw data and stores the results for retrieval into reports.

For more information about integration, data capture, and data processing, see the *Oracle Fusion Middleware WebCenter Sites: Analytics Administrator's Guide* and the *Oracle Fusion Middleware WebCenter Sites: Analytics Developer's Guide*.

Part VIII

Installing Oracle WebCenter Sites: Content Integration Platform

Part VIII contains the following chapters:

31

Installing Content Integration Platform for File Systems and Microsoft SharePoint

This chapter contains procedures for installing and configuring the Content Integration Platform to support publishing from file systems and Microsoft SharePoint systems to WebCenter Sites.

This chapter contains the following sections:

- Section 31.1, "Installation Overview"
- Section 31.2, "Installing Content Integration Platform"
- Section 31.3, "Verifying the Installation"
- Section 31.4, "Publishing Production Data"

31.1 Installation Overview

This section contains the following topics:

- Section 31.1.1, "Prerequisites"
- Section 31.1.2, "Packaging"
- Section 31.1.3, "Where to Install the Files"

31.1.1 Prerequisites

- Microsoft Visual C++ 2008 redistributable (x86), which can be downloaded from `http://www.microsoft.com`
- OpenSSL, which can be downloaded from `http://www.openssl.org`

31.1.2 Packaging

Content Integration Platform is delivered as the following set of files:

File	Description
`cipagent-vNo.msi` (for Windows) `cipagent-vNo.rpm.bin` (for Linux)	These files install Content Integration Agent (`CIPCommander` and service), and configuration files used to control the Agent process.
`csagentservices.war`	This file installs Sites Agent Services, including property files used to set the detail of log files and regulate access to the WebCenter Sites database.
`cs_filesystem_schema.zip`	This file installs the `FileSystem` flex family.

File	Description
cs_sharepoint_schema.zip	This file installs the SharePoint flex family.

31.1.3 Where to Install the Files

Content Integration Platform uses native executables.

To install Content Integration Platform:

1. Install Content Integration Agent on any computer that runs a supported operating system and can access both the source and target systems.

   > **Note:** For information about supported systems in this release, see the *Oracle WebCenter Sites Certification Matrix*.

2. Deploy csagentservices.war to a system that has access to the WebCenter Sites Shared directory.

   The WebCenter Sites system must be fully functional. **It must not be a production (delivery) system.**

3. Install the applicable schema on WebCenter Sites:

   - cs_filesystem_schema.zip
   - cs_sharepoint_schema.zip

Complete installation instructions are provided in Section 31.2, "Installing Content Integration Platform."

31.2 Installing Content Integration Platform

> **Note:** To ensure a smooth installation process, read the steps below to gain an understanding of the installation procedure and the information you will be asked to provide. To efficiently complete "Step 3: Installing Schema on WebCenter Sites," you must be an experienced WebCenter Sites administrator.

In this section, you will complete the following steps:

- Step 1: Installing Content Integration Agent
- Step 2: Installing Sites Agent Services
- Step 3: Installing Schema on WebCenter Sites

31.2.1 Step 1: Installing Content Integration Agent

1. If you are using a Windows operating system, install Microsoft Visual C++ 2008 Redistributable Package (x86) on the same computer that will host Content Integration Agent. (The redistributable package is available for download from the Microsoft web site.)

2. Run the cipagent file on a computer that runs a supported operating system and can access both the source and target systems.

   - Windows:

Run cipagent-11.1.1.6.0-36.msi and follow the steps on the screen. The 11.1.1.6.0 CIP Agent installer is located in **CIP 11.1.1.6.0 Installation Binaries** under the **Agent** folder.

The following folders are created in the target directory:

```
bin
    cipagent.exe
    cipcommander.exe
conf
    ..all conf files
security
    ..all certificates and private keys
logs
    ..log file
licenses
    ..licenses
```

- Linux:

  The 11.1.1.6.0 CIP Agent installer is located in **CIP 11.1.1.6.0 Installation Binaries** under the **Agent** folder.

  Run as a root user the following command on the source system:

  ```
  ./cipagent-11.1.1.6.0-38.el5.i386.rpm.bin
  ```

  This command installs the following directories:

  ```
  usr
    local
      bin
        cipagent -exe
        cipcommander
      lib
        cipagent
          ..all librariesshare
  share
    cipagent
      conf
      security
      logs
      licenses
  ```

3. Back up the configuration file catalog.xml (located in [path_to_cipagent]/conf/ and in Linux in /usr/local/share/conf/).

4. Edit catalog.xml.

   The catalog.xml file stores configuration settings that are required by Content Integration Agent to connect to the source system and WebCenter Sites. You will edit this file to provide Content Integration Agent with system location and user information.

   a. Using a text editor, open catalog.xm l.

   b. Edit the adapter for Oracle WebCenter Sites.

      Locate the provider element with name "cs" and id "70b1e307-26a1-499c-9295-cf0b6bd01342" and set the following parameters:

      – urlAS: Point to the Web Services module deployed with WebCenter Sites. Only the host name and port need to be modified. Typically, they are the name of the host and port where WebCenter Sites is running. Do not alter the context name and context-related path unless you are sure they differ from the default (`http://localhost:8080/csagentservices/InfostoriaService`).

- `username`: User name of the account that has permissions to modify WebCenter Sites database tables (e.g., `fwadmin`, the general administrator).
- `password`: Above user's password (e.g., `xceladmin`, assuming `fwadmin` as the username).
- `context`: Leave this blank

c. If you are using Microsoft SharePoint, edit the adapter for the SharePoint installation.

Locate the provider element with name `"sharepoint"` and id `"7137dd5d-9ed7-4327-b4fd-8caeebd5889a"`, and set the following parameters:

- `urlSharepoint`: URL pointing to the SharePoint site from which you plan to publish. Typically you need to modify only the host name (the default value is `http://localhost`).
- `username`: User name for the account that has permissions to publishable content.
- `password`: Above user's password.

d. Save `catalog.xml`.

5. Restart the Content Integration Agent executable:

   - **Windows:** Restart the Content Integration Agent service.
   - **Linux:** Type as root user: **`/sbin/service cipagent restart`**

> **Note:** The Content Integration Agent executable can be run as a standalone process or as a system daemon. The executable will start a simple HTTP server on the default port `7070`, which is reserved for `CIPCommander` communications with Content Integration Agent. Port `7070` is bound to the `localhost`, and therefore does not expose your system to any additional security risks.
>
> The `fileserver` facility default configuration takes port 7071 and attempts to automatically detect the host name. If you have more than one network interface installed on the machine where Agent is running, we advise changing `auto` to the DNS name or the IP address that is accessible from the Sites Agent Services installation.
>
> Should you need to change the port, edit the port designation in `facilities.xml` and add `-p <port>` to all commands that start `CIPCommander`.

6. Continue to the next step, "Step 2: Installing Sites Agent Services."

31.2.2 Step 2: Installing Sites Agent Services

> **Note:** Sites Agent Services can be installed on any WebCenter Sites system other than production (delivery). We recommend a content management (staging) system.

1. Edit the following files in csagentservices.war (all the files are located in csagentservices/WEB-INF/classes):

   – commons-logging.properties: defines the log file and log detail settings

   – csAgentServices.properties: enables access to the WebCenter Sites database

   a. Using a text editor, edit commons-logging.properties to point to the Agent Services log file (agentservices.log).

   b. Create a data source specific to the application server (more information is available in Part I, "Installing Oracle WebCenter Sites.")

   c. Modify csAgentServices.properties to enable access to the WebCenter Sites database.

      Using a text editor, set the following properties:

      * uploader.username: User name of an account with permissions to edit flex families.

      * uploader.password: Password for the provided user name.

      * cs.installDir: WebCenter Sites installation directory (e.g., C:\CS)

      * cs.url: WebCenter Sites URL. Point to the WebCenter Sites web application. The default value is: http://localhost:8080/cs

      Save csAgentServices.properties.

2. Deploy csagentservices.war on the application server on the WebCenter Sites host.

3. Restart the application server.

4. Continue to the next step, "Step 3: Installing Schema on WebCenter Sites."

31.2.3 Step 3: Installing Schema on WebCenter Sites

In this step, you will import the applicable zip file(s), listed below, into WebCenter Sites:

- cs_filesystem_schema.zip
- cs_sharepoint_schema.zip

To install schema

1. Run catalogmover.bat (or catalogmover.sh on Linux) from the WebCenter Sites installation directory.

> **Note:** To use CatalogMover, you must connect it to WebCenter Sites:
>
> 1. Choose **Server**, and then **Connect**.
> 2. Provide the following information:
>
>   Server: The name of the HTTP server you want to connect to, and the port on which the server is running.
>
>   Name: `ContentServer`
>
>   Password:`<password>`
>
>   Below the "Password" field, select (or enter) a value that applies to your WebCenter Sites installation.
> 3. Click **Connect**.

2. Go to **Catalog**, and then **Auto Import Catalog(s)**.

   a. Select the file to import.

   b. In the import dialog, fill in the fields as shown below:

   Catalog Data Directory: Leave the default value

   Catalog ACL List: `Browser,SiteGod,xceleditor,xceladmin`

   c. If necessary, import the remaining files.

3. Log in to the WebCenter Sites Admin interface as the general administrator (`fwadmin/ xceladmin`, by default) and continue as follows:

   a. Enable each imported flex family for an existing content management site (names of flex family members begin with the name of the source system). You can also create a new site for the flex family (or families).

   b. For easy access to published content, create a tree tab (for example, FileSystem tab or SharePoint tab).

   For instructions on enabling flex families, creating sites, and creating tree tabs, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

4. Continue to the next step, "Step 4: (Optional) Configuring Event Notification."

31.2.4 Step 4: (Optional) Configuring Event Notification

When CIP administrators must be notified of events at the source system and their synchronization, you (or the administrators) can enable any or all of the sample workflows.

> **Note:** The following workflows must be installed and enabled before an asset is published: `CIPAssetDeleted` and `CIPAssetDeletionFailed`

For information and instructions on installing sample workflows, see the chapter "Configuring Event Notification" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

31.3 Verifying the Installation

In this step, you will publish a test folder to WebCenter Sites. You will also verify the synchronization process by adding, deleting, and modifying the test folder's contents.

To verify the CIP installation

1. Back up `mappings.xml` (located on the server that hosts Content Integration Agent).
2. Select (or create) a test folder with default metadata (defined in the *default* `mappings.xml` file). That is:
   - The folder type matches the folder type in `mappings.xml` (for quick reference, see the table "Source-System Metadata Supported by Default" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.
   - The test folder contains subfolders and documents whose folder type, document type, and attributes match those in `mappings.xml` (for more information, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*).

   > **Note:** If you are verifying the publishing process from a Microsoft SharePoint system, run a test on both the document library and picture library, using `mappings.xml` (or See Table 1-1 "Source-System Metadata Supported by Default" in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*) to determine the default content types and attributes.

3. Test the publishing process. For instructions, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.
4. Test the synchronization process by renaming, moving, deleting, and creating subfolders and documents (including pictures in SharePoint). Test the attributes by deleting and adding them. (When adding attributes, follow the instructions in the "Remapping" chapter in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.)
5. Test the unpublish process by running the `unpublish` command (see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*).

31.4 Publishing Production Data

The quickest way to publish objects is to use the default `mappings.xml` file and flex families provided with CIP. Complete one of the following steps, depending on how your source system is configured:

- Objects are ready for publishing if their schema matches the default schema in `mappings.xml` and the default flex family. Follow the steps in Publishing chapter in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.
- Objects cannot be published successfully if their schema differs from the default schema in `mappings.xml` and the default flex family. Before publishing the objects, remap the schema. This requires you to update the flex family for your source system and reconfigure `mappings.xml`. For instructions, refer to in the "Remapping" chapter in the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

31.5 Customizing Your CIP Installation

Content Integration Platform can be customized in many ways. For example:

- Default flex families can be modified for custom scenarios, or they can be replaced with custom flex families.
- Flex filters can be added to flex families. (If you implement flex filters, make sure to add the corresponding `jar` files to both the WebCenter Sites and Sites Agent Services applications.)
- Attributes can be modified on the source system, or added to the source system.
- New document types can be added to the source system.
- Workflows can be implemented in order to notify administrators of CIP-related events that occur or fail to occur in WebCenter Sites, in response to changes on the source system.

Information about customizing a CIP installation can be found in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

32 Installing Content Integration Platform for EMC Documentum

This chapter contains procedures for installing and configuring the Content Integration Platform for EMC Documentum to support content transfer between WebCenter Sites and Documentum installations.

This chapter contains the following sections:

32.1 Installation Overview

Content Integration Platform for EMC Documentum enables bidirectional communication between WebCenter Sites and EMC Documentum:

- Publishing EMC Documentum objects to WebCenter Sites.
- Archiving WebCenter Sites assets on EMC Documentum.

In the archival process, the CS DataStore, rather than WebCenter Sites, is treated by CIP as the source of data. The CS DataStore is created when assets are published from WebCenter Sites to a RealTime destination with archiving enabled. The CS DataStore stores the published assets in its file system.

CIP supports publishing and archiving by replicating the source system's metadata to the target system. Schema that stores the source system's content is thus established on the target system. Content on the source system can then be transferred to the target system via the CIP command-line interface and `cipcommander publish` command. Once the publishing session ends, the synchronization engine starts monitoring the source system's published (or archived) workspaces for changes to content and automatically replicates the modified content to the target system.

This chapter shows you how to install CIP components to support publishing, archiving, or both processes, depending on your requirements:

- To support publishing or archiving, you must install Content Integration Agent, which manages metadata and the associated content. Content Integration Agent includes:
  - `mappings.xml`, which maps the source system's metadata to the target system.

  - cipcommander, a command-line application used to run the CIP publish command. The same command publishes to WebCenter Sites (via Sites Agent Services) and archives to Documentum.
  - synchronization engine, which monitors the source's published (or archived) workspaces in order to keep source and target systems in agreement.
  - catalog.xml, which stores information about published and archived items and serves as a configuration file for tuning the synchronization process.
- To enable publishing, you will install a programmatic publishing interface and database tables to store published objects, including their object type definitions:
  - The programmatic interface, named "Sites Agent Services," receives Documentum data from Content Integration Agent and stores the data to WebCenter Sites' database tables. The stored data consists of metadata in WebCenter Sites-compliant format and objects associated with the metadata.
  - Database tables for storing published objects are in the form of the Documentum flex family, a data model that must be installed on the WebCenter Sites management system and enabled on one of its content management sites. By default, the Documentum flex family also contains definitions of the Documentum object types dm_folder and dm_document. Both object types are mapped in the default mappings.xml file.
- To enable archiving, you will install the fw_asset and fw_document object types on Documentum to provide the schema for storing WebCenter Sites assets as Documentum objects. The object types are mapped in mappings.xml.

32.2 Prerequisites

Before installing Content Integration Platform, do the following:

- Download the *Oracle WebCenter Sites Certification Matrix* and release notes for this content integration product.
- Download the following:
  - Microsoft Visual C++ 2008 Redistributable (x86) from http://www.microsoft.com
  - OpenSSL from http://www.openssl.org
- Read this chapter to gain an understanding of the installation process and note the information you will need to provide.
- If you wish to set up CIP for publishing operations, create or select a content management site on the WebCenter Sites staging system for the Documentum flex family you will be installing.
- In archiving operations, CIP uses the WebCenter Sites: Web Experience Management (WEM) Framework.
- At the end of the CIP installation process, you have the option to configure event notification. If you choose to configure event notification, you must do so before publishing or archiving.

Also consider the following post-installation scenario: If your WebCenter Sites delivery system is running the Oracle WebCenter Sites: Community application and you wish to archive site visitors' comments and reviews, you will need to RealTime publish the comments and reviews from the delivery system to a separate WebCenter Sites system. The system must be running the WEM Framework. For more information, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

32.3 Packaging

Content Integration Platform is delivered as the following set of files:

File	Description
`cipagent-vNo.msi` (for Windows) `cipagent-vNo.rpm.bin` (for Linux)	For publishing and archiving. These files install Content Integration Agent (`cipcommander` and `service`) and configuration files that control the Content Integration Agent's process. One of these files must be installed on a machine with the supported operating system and the ability to access both the source and target systems.
`csagentservices.war`	For publishing. This file installs Sites Agent Services, including property files for setting the detail of log files and regulating access to the WebCenter Sites database. This file must be deployed on a system other than delivery. The system must have access to the WebCenter Sites `Shared` directory.
`cs_documentum_schema.zip`	For publishing. This file installs the `Documentum` flex family on WebCenter Sites, on the CM site that you specify.
`documentum_cs_schema.dar`	For archiving. This file installs schema on the Documentum system to support the content of the CS DataStore.

32.4 Installation Steps

In this section, you will install Content Integration Platform for EMC Documentum. Your steps are the following:

- Step 1. Installing Content Integration Agent
- Step 2. Installing CIP Publishing Components
- Step 3. Installing Schema to Support Archiving to Documentum
- Step 4. Backing Up the Default mappings.xml File

32.4.1 Step 1. Installing Content Integration Agent

1. If you are using Windows, install Microsoft Visual C++ 2008 Redistributable Package (x86) on the machine that will host Content Integration Agent.
2. Run the `cipagent` file on a machine with a supported operating system and the ability to access the source and target systems.
   - Windows:

     Run `cipagent-2.0.0-127.msi` and follow the prompts.

     The following folders are created in the target directory:

     ```
     bin
       cipagent.exe
       cipcommander.exe
     conf
     ```

```
  .. all conf files
security
  .. all certificates and private keys
logs
  ..log file
licenses
  ..licenses
```

- Linux:

  Run (as a root user) the following command on the source system:

  ```
  ./cipagent-2.0.0-128.el5.i386.rpm.bin
  ```

  The following directories are installed:

  ```
  usr
    local
       bin
         cipagent -exe
         cipcommander
       lib
         cipagent
           ..all libraries
  share
    cipagent
       conf
       security
       logs
       licenses
  ```

3. Back up the configuration file `catalog.xml` (located in `integration_agent/conf/`).
4. Edit `catalog.xml`.

   The `catalog.xml` file stores configuration settings that Content Integration Agent requires to connect to the source system and Oracle WebCenter Sites. You will edit this file to provide Content Integration Agent with system location and user information.

   a. Open `catalog.xml` in a text editor.

   b. Edit the adapter for Oracle WebCenter Sites:

      Locate the provider element with name "`cs`" and id "`70b1e307-26a1-499c-9295-cf0b6bd01342`" and set the following parameters:

      – `urlAS`: Point to the Web Services module deployed with WebCenter Sites. Only the host name and port need to be modified. Typically, they name the host and the port number where WebCenter Sites is running. Do not alter the context name and context-related path unless you they differ from the default (`http://localhost:8080/csagentservices/InfostoriaService`).

      – `username`: User name of the account that has permissions to modify WebCenter Sites database tables (e.g., `fwadmin`, the general administrator).

      – `password`: Above user's password (e.g., `xceladmin`, assuming `fwadmin` is the username).

      – `context`: Leave this blank.

   c. Edit the adapter for the EMC Documentum installation:

Locate the provider element with name "documentum" and id "d7a96a63-e78c-407c-8d7f-e84988806e49" and set the following parameters:

- urlDocumentum: URL pointing to the server where Documentum Foundation Services (DFS) are running. Include the context name, but omit context-related paths. Typically you need to modify only the host name (the default value is http://localhost:9090).
- username: User name for the account that has permissions to publishable content.
- password: Above user's password.
- filter: This parameter contains a where clause which is applied to filter out document versions that are not intended for publication. For example, to publish the latest approved version of a document, you can specify values such as the following:

  ```
  r_current_state = 1
  ```

  - or -

  ```
  r_policy_id = '0000000000000000'
  ```

  (taking into account that the approved state has "1" as an index)

d. Save catalog.xml.

5. Restart the Content Integration Agent executable:
   - **Windows:** Restart the Content Integration Agent service.
   - **Linux:** Type as root user: /sbin/service restart cipagent

Note: The Content Integration Agent executable can be run as a standalone process or as a system daemon. The executable will start a simple HTTP server on the default port 7070, which is reserved for CIPCommander communications with Content Integration Agent. Port 7070 is bound to the localhost, and therefore does not expose your system to any additional security risks.

The fileserver facility default configuration takes port 7071 and attempts to automatically detect the host name. If multiple network interfaces are installed on the machine where Agent is running, we advise changing auto to the DNS name or the IP address that is accessible from the Sites Agent Services installation.

Should you need to change the port, edit the port designation in facilities.xml and add **-p <port>** to all commands that start CIPCommander.

6. In facilities.xml, enable the java facility section. Specify the path to Java on your system.
7. Continue to the next step, Section 32.4.2, "Step 2. Installing CIP Publishing Components."

32.4.2 Step 2. Installing CIP Publishing Components

If you wish to install only archival capability, skip to Section 32.4.3, "Step 3. Installing Schema to Support Archiving to Documentum." Otherwise, complete all the steps in this section to install publishing capability.

32.4.2.1 A. Installing Sites Agent Services

> **Note:** Sites Agent Services can be installed on any WebCenter Sites system other than delivery. We recommend a content management (staging) system.

1. Edit the following files in `csagentservices.war` (all the files are located in `csagentservices/WEB-INF/classes`):
   - `commons-logging.properties`: defines the log file and log detail settings.
   - `csAgentServices.properties`: enables access to the WebCenter Sites database.

   a. Using a text editor, edit `commons-logging.properties` to point to the Sites Agent Services log file (`agentservices.log`).

   b. Create a data source specific to the application server. (More information is available in the guide for installing WebCenter Sites on the application server you are using.)

   c. Modify `csAgentServices.properties` to enable access to the WebCenter Sites database.

      – Using a text editor, set the following properties:

        `uploader.username:` User name of an account with permissions to edit flex families.

        `uploader.password:` Password for the above user name.

        `cs.installDir`: WebCenter Sites installation directory (e.g., `C:\CS`)

        `cs.url`: WebCenter Sites URL. Point to the WebCenter Sites web application. The default value is: `http://localhost:8080/cs`

      – Save `csAgentServices.properties`.

2. Deploy `csagentservices.war` on the application server on the WebCenter Sites' host.
3. Restart the application server.
4. Continue to the next step, Section 32.4.2.2, "B. Installing EMC Documentum Schema on Oracle WebCenter Sites."

32.4.2.2 B. Installing EMC Documentum Schema on Oracle WebCenter Sites

In this step, you will import `cs_documentum_schema.zip` into WebCenter Sites to support the publication of Documentum objects to WebCenter Sites.

To install the Documentum schema

1. Run `catalogmover.bat` (or `catalogmover.sh` on Linux) from the WebCenter Sites installation directory.

Note: Before using CatalogMover, connect it to WebCenter Sites:

Below the "Password" field, select (or enter) a value that applies to your WebCenter Sites installation.

- Choose **Server**, then select **Connect**.
- Provide the following information:

  `Server`: The name of the HTTP server you want to connect to, and the port on which the server is running.

  `Name:ContentServer`

  `Password:<password>`
- Click **Connect**.

2. Go to **Catalog**, then select **Auto Import Catalog(s)**.

   a. Select the file to import.

   b. In the import dialog, fill in the fields as shown below:

   Catalog Data Directory: Leave the default value

   Catalog ACL List: Browser, SiteGod, xceleditor, xceladmin

3. Log in to the WebCenter Sites' Admin interface as a general administrator (`fwadmin/xceladmin`, by default) and continue as follows:

   a. Enable the `Documentum` flex family on the content management site you have chosen or created, as explained in Section 32.2, "Prerequisites."

   b. For quick access to published content, create a tree tab (named `Documentum`, for example).

   For instructions on enabling flex families, creating sites, and creating tree tabs, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

4. Continue to the next step, Section 32.4.3, "Step 3. Installing Schema to Support Archiving to Documentum."

32.4.3 Step 3. Installing Schema to Support Archiving to Documentum

Install `documentum_cs_schema.dar` using Documentum Composer (installation instructions and user information are available in the *Documentum Composer User Guide):*

```
CREATE TYPE fw_asset (
fw_id char(255),
fw_name char(64),
fw_createdby char(64),
fw_createddate time,
fw_description char(128),
fw_publist char(255),
fw_status char(2),
fw_subtype char(32),
fw_updatedby char(64),
fw_updateddate time,
fw_publisheddate time
) WITH SUPERTYPE dm_document
```

```
CREATE TYPE fw_document (
fw_publisheddate time,
fw_name char(64)
) WITH SUPERTYPE dm_document
```

32.4.4 Step 4. Backing Up the Default mappings.xml File

The `mappings.xml` file is located on the server that hosts Content Integration Agent.

32.5 Next Steps

Once CIP is installed, complete the following steps, as necessary:

- `Optional`. Configure event notification workflows to inform CIP administrators about events that are triggered on the target system by changes to data in the source system's monitored workspaces (that is, published or archived workspaces).

  > **Note:** If you choose to configure event notification workflows, you must do so before publishing any Documentum objects or archiving any WebCenter Sites assets. For instructions on configuring event notification workflows, see the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide*.

- To publish from Documentum to WebCenter Sites, refer to the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* for background information and instructions.
- To archive WebCenter Sites assets to Documentum, refer to the *Oracle Fusion Middleware WebCenter Sites Administrator's Guide* for background information and instructions.

Part IX

Reference: Backing Up and Recovering

Part IX contains the following chapters:

33 Backup and Recovery

This document contains the following sections:

- Section 33.1, "Prerequisites for Backup and Recovery"
- Section 33.2, "Procedures for Backup and Recovery"
- Section 33.3, "Running the Recovered WebCenter Sites Application"

33.1 Prerequisites for Backup and Recovery

- Before backing up any part of WebCenter Sites, it is best to shut down the application server. If you are backing up an active WebCenter Sites, ensure the active site is not in use and no requests are being made to the application server.
- Before recovering any part of WebCenter Sites, ensure the application server is shut down.

33.2 Procedures for Backup and Recovery

- Section 33.2.1, "WebCenter Sites Installation Directory"
- Section 33.2.2, "Shared Directory"
- Section 33.2.3, "Web Application .war and .ear Files"
- Section 33.2.4, "WebCenter Sites Database" (excluding statistics)
- Section 33.2.5, "LDAP"

33.2.1 WebCenter Sites Installation Directory

This section contains the following subsections:

- Section 33.2.1.1, "Backup"
- Section 33.2.1.2, "Recovery"

33.2.1.1 Backup

Jar or tar the WebCenter Sites installation directory.

For example:

```
tar -cvf ContentServer_backup.tar /u01/CS/Install
jar cvf ContentServer_backup.jar /u01/CS/Install
```

33.2.1.2 Recovery

Unjar or untar the backed up WebCenter Sites installation directory.

For example:

```
tar -xvf ContentServer_backup.tar
jar xvf ContentServer_backup.jar
```

33.2.2 Shared Directory

This section contains the following subsections:

- Section 33.2.2.1, "Backup"
- Section 33.2.2.2, "Recovery"

33.2.2.1 Backup

Note: Backing up the `Shared` directory is unnecessary if it is located in the WebCenter Sites installation directory, and the installation directory was backed up.

To back up the `Shared` directory, jar or tar the directory.

For example:

```
tar -cvf Shared_backup.tar /u01/CS/Shared
jar cvf Shared_backup.jar /u01/CS/Shared
```

33.2.2.2 Recovery

Note: Recovering the `Shared` directory is unnecessary if it is located in the WebCenter Sites installation directory, and the installation directory was recovered.

To recover the `Shared` directory, unjar or untar the backed up `Shared` directory and use it to replace the existing one.

For example:

```
tar -xvf Shared_backup.tar
jar xvf Shared_backup.jar
```

33.2.3 Web Application .war and .ear Files

Table 33–1 summarizes which types of files in a WebCenter Sites web application must be backed up and recovered.

Table 33–1 Backup and Recovery Information for .war and .ear files

.war/.ear Deployment Method	Backup	Recovery
Non-Exploded. For WebCenter Sites on WebSphere	Unnecessary if the WebCenter Sites installation directory is backed up. (The `.war` and `.ear` files are contained in the WebCenter Sites installation directory: `<sites_install_dir>/ominstallinfo/app/`)	Refer to the steps in Section 33.2.3.2.1, "Recovering Non-Exploded Files."
Exploded. For WebCenter Sites on: - Tomcat - WebLogic	No "development changes" to the files.[1] Backup is unnecessary if: - The files have not been changed during development, - and - - the WebCenter Sites installation directory is backed up. (The `.war` and `.ear` files are contained in the WebCenter Sites installation directory: `<sites_install_dir>/ominstallinfo/app/`)	Refer to the steps in Section 33.2.3.2.2, "Recovering Exploded Files with No Development Changes."
	With "development changes" to the files.[1] Backup is required. Refer to the steps in Section 33.2.3.1.3, "Backing Up Exploded Files With Development Changes."	Refer to the steps in Section 33.2.3.2.3, "Recovering Exploded Files with Development Changes."

[1] The term "development changes" denoted changes to the content, the data model, template code, publishing destinations, and so on.

33.2.3.1 Backup

This section contains the following subsections:

- Section 33.2.3.1.1, "Backing Up Non-Exploded Files"
- Section 33.2.3.1.2, "Backing Up Exploded Files With No Development Changes"
- Section 33.2.3.1.3, "Backing Up Exploded Files With Development Changes"

33.2.3.1.1 Backing Up Non-Exploded Files

See Table 33–1.

33.2.3.1.2 Backing Up Exploded Files With No Development Changes

See Table 33–1.

33.2.3.1.3 Backing Up Exploded Files With Development Changes

Jar or tar the deployed WebCenter Sites web application. Label the backup to distinguish it from the compressed `cs.war` file.

For example:

```
tar -cvf cs_web_backup.tar /u01/software/Tomcat/webapps/cs
jar cvf cs_web_backup.jar /u01/software/Tomcat/webapps/cs
```

33.2.3.2 Recovery

This section contains the following subsections:

- Section 33.2.3.2.1, "Recovering Non-Exploded Files"
- Section 33.2.3.2.2, "Recovering Exploded Files with No Development Changes"
- Section 33.2.3.2.3, "Recovering Exploded Files with Development Changes"

33.2.3.2.1 Recovering Non-Exploded Files

1. Undeploy the WebCenter Sites application.
2. Recover the WebCenter Sites installation directory (for instructions, see Section 33.2.1.2, "Recovery").
3. Redeploy the WebCenter Sites application.

33.2.3.2.2 Recovering Exploded Files with No Development Changes

1. Stop the application server.
2. Delete the exploded web application.

   For example:

   ```
   rm -rf /u01/software/Tomcat/webapps/cs/*
   ```

3. Recover the WebCenter Sites installation directory (for instructions, see Section 33.2.1.2, "Recovery").
4. Unjar the backed up WebCenter Sites web application:

   For example:

   ```
   cd /u01/software/Tomcat/webapps/cs
   jar xvf <sites_install_dir>/ominstallinfo/app/cs.war
   ```

33.2.3.2.3 Recovering Exploded Files with Development Changes

1. Stop the application server.
2. Delete the exploded web application.

   ```
   rm -rf /u01/software/Tomcat/webapps/cs
   ```

3. Unjar or untar the backed up WebCenter Sites web application:

   ```
   cd /u01/software/Tomcat/webapps
   tar -xvf <path to backup directory>/cs_web_backup.tar
   ```

33.2.4 WebCenter Sites Database

This section contains the following subsections:

- Section 33.2.4.1, "Backup"
- Section 33.2.4.2, "Recovery"

33.2.4.1 Backup

- Section 33.2.4.1.1, "Backing up SQL Server 2008 and 2008R2"
- Section 33.2.4.1.2, "Backing Up Oracle 11g"

- Section 33.2.4.1.3, "Backing Up DB2 9.7"

33.2.4.1.1 Backing up SQL Server 2008 and 2008R2

1. Open **SQL Server Management Studio**.
2. Connect to the database server.
3. Expand **Databases**.
4. Right-click on the database to be backed up. Click **Tasks**, then click **Backup**.
5. Under **Backup type**, select **Full**. Under **Destination**, click **Add**.
6. Enter the path and file name for the backup file. Click **OK**.
7. Under **Destination**, select the newly created backup location.
8. Click **OK**.

33.2.4.1.2 Backing Up Oracle 11*g*

1. Log in to the server as the oracle user.
2. Set `ORACLE_HOME` to the oracle database directory and `ORACLE_SID` to the database name.
3. Log in to sqlplus as `sys`. Then, do the following:
   a. Use the `create directory` sql to map to the OS directory, where the exports will be stored. For example:

      ```
      SQL> create directory exp_dp_dir as '/u01/backup/exports';
      ```

   b. Grant permissions for database export and directory access. For example:

      ```
      SQL> grant read, write on directory exp_dp_dir to system;
      SQL> grant EXP_FULL_DATABASE to system
      ```

4. Run the `expdp` comm and to start exporting. For example:

   ```
   expdp system/<password> DIRECTORY=exp_dp_dir DUMPFILE=<file_name>.dmp
   FULL=y;
   ```

33.2.4.1.3 Backing Up DB2 9.7

1. Log in to the server as the `db2inst` user.
2. Run the following command:

   ```
   db2 force applications all
   ```

3. Run the following command:

   ```
   db2 backup db <db_name> to <backup_dir>
   ```

   Note: After the command has completed, note the timestamp. This timestamp will be included in the name of the database backup file.

33.2.4.2 Recovery

- Section 33.2.4.2.1, "Recovering SQL Server 2008 and 2008R2"
- Section 33.2.4.2.2, "Recovering Oracle 11g"
- Section 33.2.4.2.3, "Recovering DB2 V9.7"

33.2.4.2.1 Recovering SQL Server 2008 and 2008R2

1. Open **SQL Server Management Studio**.
2. Expand **Databases**.
3. Right-click on the database to be restored. Select **Tasks**, then select **Restore**, and then select **Database**.
4. Under **Source for restore**, select **From database**.
5. Under **Select the backup sets to restore**, select the most recent backup.
6. Click **Script** at the top.

   This step opens the query editor containing the `restore` statement.
7. Edit the query by adding **REPLACE**, after **WITH**.

   This will allow the database log to be overwritten.
8. Click **Execute**.

33.2.4.2.2 Recovering Oracle 11*g*

1. Log in to the server as the oracle user.
2. Set `ORACLE_HOME` to the oracle database directory and `ORACLE_SID` to the database name.
3. Go to `$ORACLE_HOME/bin`.
4. Run the following command:

```
./impdp system/<password> DIRECTORY=exp_dp_dir DUMPFILE=<file_name>.dmp
    SCHEMAS=<YOUR_SCHEMA>;
```

33.2.4.2.3 Recovering DB2 V9.7

1. Log in to the server as the `db2inst` user.
2. Run the following command:

```
db2 force applications all
```

3. Run the following command:

```
db2 restore db <db_name> from <backup_dir> replace existing
```

33.2.5 LDAP

This section contains the following subsections:

- Section 33.2.5.1, "Backup"
- Section 33.2.5.2, "Recovery"

33.2.5.1 Backup

Use an LDAP browser and export the LDAP configuration.

33.2.5.2 Recovery

Start with a clean LDAP server. Then use an LDAP browser and import the backed up LDAP configuration.

33.3 Running the Recovered WebCenter Sites Application

1. Before starting the application server:

   a. Clear the search indexes:

   ```
   rm -rf <shared_dir>/lucene/Global/*
   ```

   b. If you are using Tomcat, clear the application server cache:

   ```
   rm -rf <server_dir>/tmp/*
   rm -rf <server_dir>/work/*
   ```

   Note: For the WebLogic application server, the cache will be cleared upon redeployment of the application.

   c. If you are using Tomcat or WebLogic, clear the compiled JSPs:

   ```
   rm -rf <web_app_dir>/jsp/cs_deployed/*
   ```

2. After starting the application server, ensure that all caches are synchronized.

Glossary for the Oracle WebCenter Sites: Analytics Application

Analytics Data Capture Application

Also referred to as the "sensor."

Asset Registration

Enabling report generation for assets. Because WebCenter Sites assets are specific to a WebCenter Sites installation, you must register their asset types with Analytics by assigning them to reports through the Analytics Administration interface. This enables Analytics to:

- Recognize WebCenter Sites asset types
- Configure report menu options in the "General Information" and "Content Information" report groups
- Generate reports on assets of the registered asset types

Data Capture

The process of recording each visitor's clicks and the associated information—the date and time of each click, the assets that are clicked, the IP address from which the clicks are issued, the site being visited, and so on. The information is captured in real time by the `sensor` servlet and recorded in a `data.txt.tmp` file on the local file system (local to the Analytics data capture application). The `data.txt.tmp` file will be rotated by the sensor to `data.txt` when either the threshold interval is reached (see the `sensor.threshold` property on sensor.thresholdtime), or the application server is restarted.

Analytics can capture data on the usage of WebCenter Sites assets and on their visitors only if published pages are tagged for data capture. In the case of Engage assets, the assets themselves must be tagged for data capture.

Hadoop Jobs

Runs jobs in a parallel and distributed fashion in order to efficiently compute statistics on the raw data that is stored in the Hadoop Distributed File System.

Hadoop implements a computational paradigm named `Map/Reduce`, which divides a large computation into smaller fragments of work, each of which may be executed on any node in the cluster. `Map/Reduce` requires a combination of `jar` files and classes, all of which are collected into a single `jar` file that is usually referred to as a "job" file. To execute a job, you submit it to a `JobTracker`. Hadoop Jobs then responds with the following actions:

- Schedules and submits the jobs to `JobTracker`.
- Processes raw data captured by the data capture application into statistical data and injects the statistics into the Analytics database.

(Hadoop provides a web interface to browse HDFS and to determine the status of the jobs.)

Hadoop jobs pre-calculate commonly requested site usage statistics (such as average number of requests for a piece of content per unit time) in order to shorten report generation time. Statistical computation is typically resource-intensive and time-consuming. Therefore, it is performed not on-the-fly, each time a report is generated, but in advance so that it can be available by the time it is needed. Thus, precalculated statistics are immediately available for retrieval into reports. Statistics include, for example:

- Current information, such as today's total hits to each site, visiting countries, total number of visits from a given country, types of browsers, and average session duration.
- Historical results, such as:

  Daily, weekly, and monthly statistics—for example, the total number of requests for a given asset on a given site during a certain month in the reporting period. Yearly statistics—a histogram in the performance indicator indicating the frequency with which certain assets were accessed during each week of the past year.

  How long a Hadoop job runs depends on a number of factors, including site activity within the latest data capture time frame, the cumulative volume of captured data, and the configuration of the Analytics application. When data analysis is complete, the resulting statistics are available, at any time, for report generation.

Integration

Integrating Analytics with your WebCenter Sites system means enabling report generation for asset types and users on your online site. Integration involves registering CM sites, WebCenter Sites users, and asset types with Analytics, configuring the Pageview Object (through the "Page Views" Report), and granting users the appropriate permissions through membership in the appropriate user groups. The steps necessary to accomplish these tasks are described in Integrating Analytics with WebCenter Sites.

Internal Search

A search performed by a visitor using the site's built in search engine. This search returns results from within the site's contents.

Object

An Analytics construct. The subject of a report.

When storing and processing information, Analytics uses objects, whereas WebCenter Sites uses assets and asset types. To allow Analytics to recognize a WebCenter Sites asset type and track assets of that type, administrators define an Analytics object in terms of a WebCenter Sites asset type. They do so by configuring an Analytics report for the object and assigning the desired asset type to that object. The process of configuring a report defines the underlying asset.

Note: A special instance of an object is the Pageview Object, which administrators must configure (by configuring the "Page Views" Report) in order for reports in the "General Information" group to work.

The "Page Views" report supports multiple asset types.

Object Impression

A single invocation of the sensor servlet. For more information, see Object Impressions.

Page View

An Analytics construct. A group of one or more assets, whose asset types are enabled for tracking by the Analytics data capture application.

Asset types are enabled for tracking when they are defined in the Pageview Object and when published pages displaying those asset types are tagged with `AddAnalyticsImgTag` (data capture tag). For more information about tracking, see Data Capture.

Pageview Object

A default Analytics object which you configure through the "Page Views" report. The `Pageview` object is the basis for the "Page Views," "Site Information" and "Clickstream" reports, and thus it should be assigned asset types whose assets make the most sense (from the marketing standpoint) to be included in these reports.

A `Pageview` object can be assigned multiple asset types. The "Page Views" report will contain statistics on the usage of those asset types.

"Page Views" Report

A report, based on the Pageview Object. The "Page Views" report displays statistics on Page View activity on your site.

Processed Data

Visitor activity data that has been processed by Hadoop Jobs into statistical data. When processing is complete, the data is injected into the Analytics database, where it is immediately available for the reports that users request from the Analytics reporting interface.

Raw Data

Unprocessed data describing visitor activity on the site, recorded during the Data Capture process and stored in the local file system for future processing. This is the data on which statistics are calculated by the Hadoop Jobs for display in reports. (This data cannot be directly used for report generation.)

Sensor

Also referred to as the "Analytics data capture application."

Site Registration

Identifying a WebCenter Sites CM site to Analytics in order to enable Analytics to track visitor activity on that site.

Statistical Data

See Processed Data.

Work Package

A collection of object impressions. For more information, see Object Impressions and Work Packages.