Oracle® WebCenter Sites

Installation Guide for the Community Application

11*g* Release 1 (11.1.1)

February 2012

ORACLE®

Oracle® WebCenter Sites: Installation Guide for the Community Application 11*g* Release 1 (11.1.1)

Primary Author: Tatiana Kolubayev

Contributing Author: Eric Gandt, Gaurang Mavadiya, Guthrie Taber

Contributor: Dhirendra Kumar, Igor Dzyuba, Sailaxmi Rajanala

Table of

Contents

About This Guide 5
Audience 5
Related Documents 6
Conventions 6
Graphics in This Guide 6
Third-Party Libraries 6

1 Introduction 7
WebCenter Sites: Community 8
Components of WebCenter Sites: Community 8
Production and Management Components 9
Production and Management Environments 9
WebCenter Sites Communications 9
Community Application Communications 10
WebCenter Sites: Community Configurations 10
Basic Configuration 10
Production Configurations 12

2 Prerequisites 17
Before Installing WebCenter Sites: Community 18
Prerequisites for All Installations 18
Prerequisites for Clustered Environments 19
Prerequisites for LDAP Configuration 19
Path and Directory Naming Conventions 20

3 Configuring Application Servers 23
Overview 24
Configuring for Non-Clustered Deployments 24
Configuring Apache Tomcat 24
Configuring Oracle WebLogic Application Server 25

Configuring IBM WebSphere Application Server 27
Configuring for Clustered Deployments 29
Configuring Apache Tomcat (Clustered) 29
Configuring Oracle WebLogic Application Server (Clustered) 32
Configuring IBM WebSphere Application Server (Clustered) 35

4 Installing Oracle WebCenter Sites: Community 39
Installing Graphically 40
Installing Silently 59
Community Application Directory Structure 60
Post-Installation Steps 61
A. Configuring inCache Invalidation 61
B. Copying Installer-Generated Files 61
C. Proxy Connection Settings 63

5 Deploying Oracle WebCenter Sites: Community 65
Overview 66
Deploying a Non-Clustered Community Application 66
Deploying on Apache Tomcat 66
Deploying on Oracle WebLogic Application Server 68
Deploying on IBM WebSphere Application Server 69
Deploying a Clustered Community Application 71
Deploying on Apache Tomcat (Clustered) 71
Deploying on Oracle WebLogic Application Server (Clustered) 73
Deploying on IBM WebSphere Application Server (Clustered) 75

6 Verifying and Registering Oracle WebCenter Sites: Community 77
Verifying Community URLs 78
Registering the Community Application 80
Authorizing Users 81

About This Guide

This guide described the process of installing and configuring the Oracle WebCenter Sites: Community web application in single and clustered mode.

Applications discussed in this guide are former FatWire products. Naming conventions are the following:

- *Oracle WebCenter Sites* is the name of the product previously known as *FatWire Content Server.* In this guide, *Oracle WebCenter Sites* is also called *WebCenter Sites*.
- *Oracle WebCenter Sites: Community* is the name of the product previously known as *FatWire Community Server.* In this guide, *Oracle WebCenter Sites: Community* is also called *Community.*
- *Oracle WebCenter Sites: Web Experience Management Framework* is the name of the environment previously known as *FatWire Web Experience Management Framework.* In this guide, *Oracle WebCenter Sites: Web Experience Management Framework* is also called *WEM Framework*.
- *Oracle WebCenter Sites: Explorer* is the name of the utility previously known as *FatWire Content Server Explorer.* In this guide, *Oracle WebCenter Sites: Explorer* is called *Sites Explorer.*

The Community application integrates with Oracle WebCenter Sites according to specifications in the *Oracle WebCenter Sites 11g Release 1 (11.1.1.x) Certification Matrix.* For additional information, see the release notes for the Community application. Check the WebCenter Sites documentation site regularly for updates to the *Certification Matrix* and release notes.

Audience

This guide is for installation engineers and anyone else who has expertise with Oracle WebCenter Sites and the process of installing enterprise-level software. Users of this guide should be familiar with the WebCenter Sites Admin interface, especially mirror publishing. Also required is experience with the Oracle WebCenter Sites: Web Experience Management framework, the process of registering applications, and authorizing users to access the applications.

Related Documents

For more information, see the following documents:

- *Oracle WebCenter Sites User's Guide for the Community Application*
- *Oracle WebCenter Sites Administrator's Guide for the WEM Framework*
- *Oracle WebCenter Sites Developer's Guide*

Conventions

The following text conventions are used in this guide:

- **Boldface** type indicates graphical user interface elements that you select.
- *Italic* type indicates book titles, emphasis, or variables for which you supply particular values.
- `Monospace` type indicates file names, URLs, sample code, or text that appears on the screen.
- **`Monospace bold`** type indicates a command.

Graphics in This Guide

Many steps in this guide display screen captures or dialog boxes. These screen captures and examples are shown to help you follow the installation process. They are not intended to be sources of specific information such as parameter values, options to select, or product version numbers.

Third-Party Libraries

Oracle WebCenter Sites and its applications include third-party libraries. For additional information, see *Oracle WebCenter Sites 11gR1: Third-Party Licenses*.

Chapter 1

Introduction

This chapter provides an overview of WebCenter Sites: Community and describes its configuration for both a simple development or content management environment, and a complex production environment.

This chapter contains the following sections:

- WebCenter Sites: Community
- Production and Management Environments
- WebCenter Sites: Community Configurations

WebCenter Sites: Community

Oracle WebCenter Sites: Community is a Java EE web application that integrates with Oracle WebCenter Sites and works in a distributed environment as a social computing application designed to gather visitors' comments, reviews, and ratings on website content. The Community application also enables you to create and manage polls which can be used to survey site visitors about desired topics.

Before proceeding to describe the configurations in which the Community application can be installed, this chapter provides an overview of the Community application, its relationship to WebCenter Sites, its components, and their purposes.

Components of WebCenter Sites: Community

The Community application integrates with WebCenter Sites through the Web Experience Management Framework to make use of the following resources:

- **Asset repository**. The Community application requires an asset repository. WebCenter Sites provides an asset repository, that is, a database. The Community application does not directly access the database. All interaction with the database is managed by WebCenter Sites.
- **WebCenter Sites: Web Experience Management Framework**, which provides the Community application with REST services and the Central Authentication Service (CAS).
  - REST services enable the Community application to communicate with WebCenter Sites in order to make use of its asset model.
  - CAS is used by both the Community application and WebCenter Sites. Both applications require their own CAS instances, which are used for different purposes.

    The production instance of the Community application (described in "Production and Management Components") uses CAS to manage website visitors who interact with Community widgets. This CAS instance is referred to as "visitor CAS." WebCenter Sites uses an instance of CAS to provide authentication services for the REST API.

    CAS also provides a single sign-on service, enabling users of the Community application to log in to WebCenter Sites once and access all applications intended for their use. More information about CAS is available at the following URL: `http://www.jasig.org/cas`

    At the code level, the CAS web applications are nearly identical. However, their configurations differ and they are not interchangeable, nor can they be installed on the same application server or cluster.

Production and Management Components

The Community application consists of two parts: a production (delivery) application and a management application.

Note

The management and production Community applications must be deployed on separate application server instances to be supported. Deployment on a single physical server is not supported.

The management instance of the Community application is used for administration of user-generated content (UGC) and is normally situated such that it is accessible only internally. The production instance of the Community application is accessed by visitors through widgets deployed onto web pages and manages visitor authorization via visitor CAS. Hence, both the Community and the CAS applications must be externally accessible.

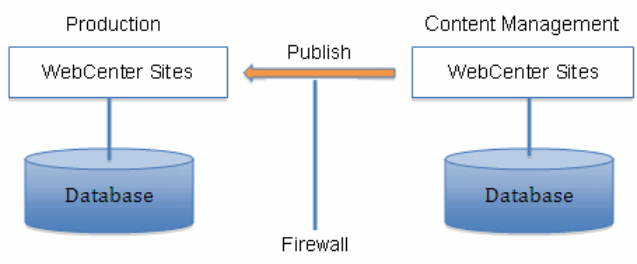
Production and Management Environments

To manage user-generated content, the Community application requires two environments: a management environment and a production environment. The management and production deployments for the Community application are similar to the content management and production deployments for WebCenter Sites. However, unlike the WebCenter Sites applications, the management and production Community applications are in constant communication with each other. The sections below explain the differences in communications between WebCenter Sites applications and between Community applications.

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. Additionally, templates are generated on the content management system, then published to the production system to configure the appearance of published content.

Figure 1: WebCenter Sites Communications.
Website content is generated, managed, and published strictly by users of the WebCenter Sites content management application.



Community Application Communications

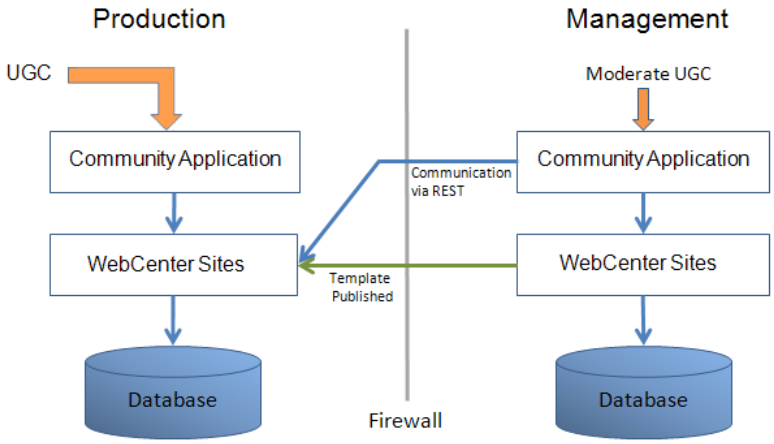
In the Community application, content generation and management processes differ from the processes on WebCenter Sites because they involve several types of users: Designers, moderators, and website visitors, all performing activities on user-generated content.

On the management instance of the Community application, designers configure widgets (such as comments, reviews, and polls) for the production instance. Also on the Management instance, moderators monitor user-generated content, while visitors on the Production instance generate new content in deployed widgets. Figure 2 summarizes the communication pathways.

Note

Figure 2 indicates the use of a firewall. The firewall must allow free information exchange on all ports used by WebCenter Sites and the Community application.

Figure 2: Community Application Communications.
Content is generated by website visitors using the production Community application. The same content is managed by users of the management Community application.



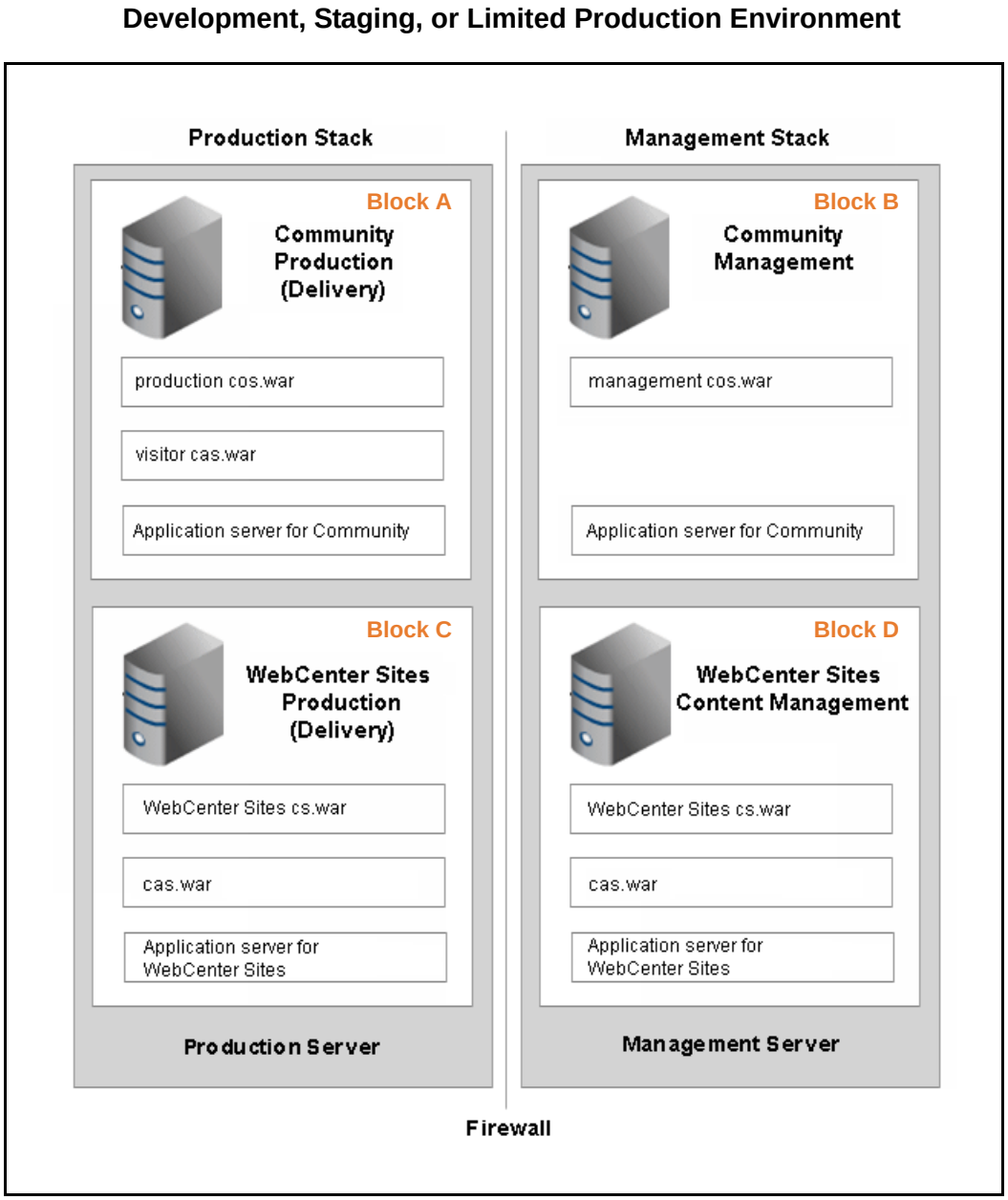
WebCenter Sites: Community Configurations

The Community environment can be configured in many ways. This section describes the commonly used configurations: basic and production.

Basic Configuration

A basic Community environment consists of four self-contained blocks: Community (production), Community (management), WebCenter Sites (production), and WebCenter Sites (content management). These blocks are depicted in Figure 3, where they are labeled as A, B, C, and D.

Figure 3: Basic Community Environment



The blocks shown in Figure 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 physical server is not supported.

The configuration shown in Figure 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 3, two independent physical servers are used. Each server has an independent stack consisting of an instance of WebCenter Sites: Community 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).

Production Configurations

Deploying the Community application on a production environment uses the same four basic blocks introduced in "Basic Configuration," on page 10. 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 4, on page 13 shows a sample management stack in a production environment.
- Figure 5, on page 14 shows a sample production (delivery) stack in a production environment.
- Figure 6, on page 15 shows a community application production block divided into two sub-blocks.
- Figure 7, on page 16 shows a community application production block divided into four sub-blocks.

Figure 4: Sample Management Stack in a Production Environment

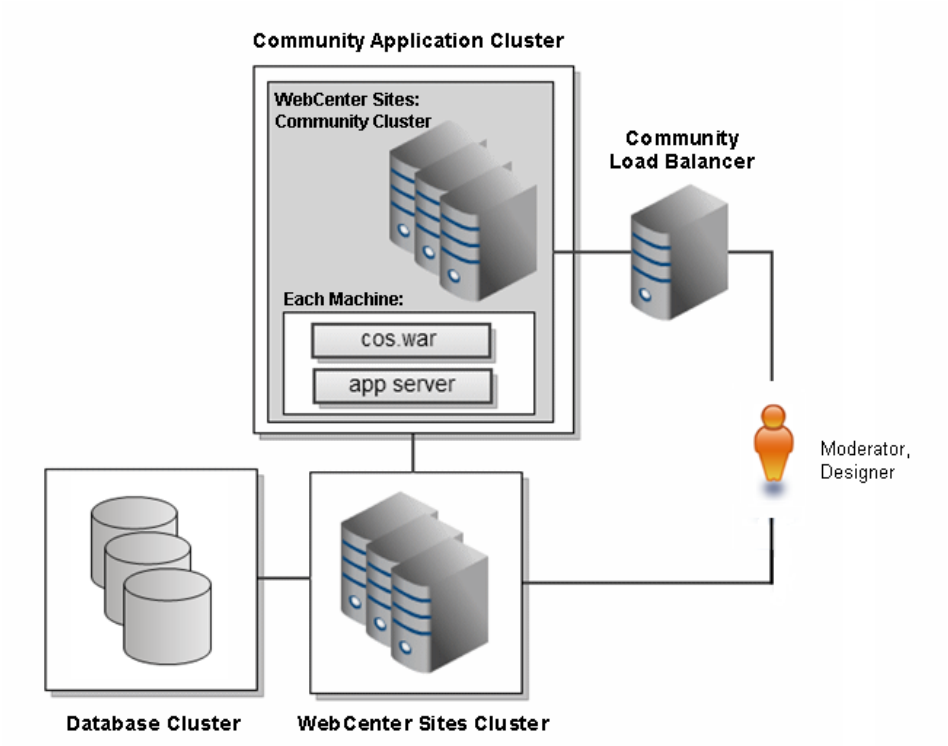


Figure 4 shows a clustered version of the management stack illustrated in Figure 3, on page 11. In Figure 4, the Community application (Block B) is now clustered. The WebCenter Sites application (Block D) is also clustered and uses a database cluster.

Figure 5: Sample Production (Delivery) Stack in a Production Environment

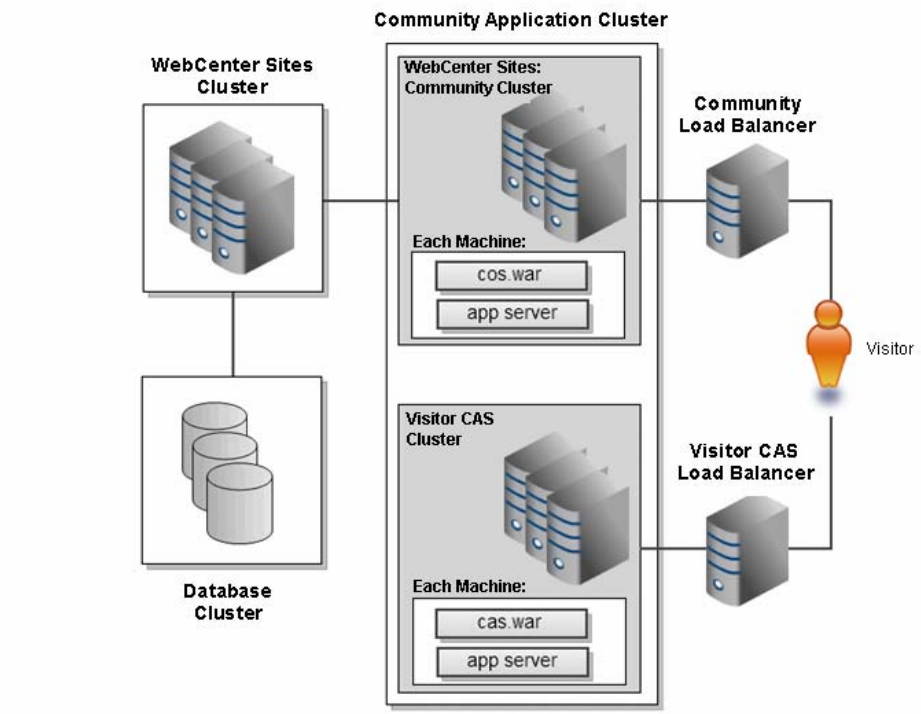
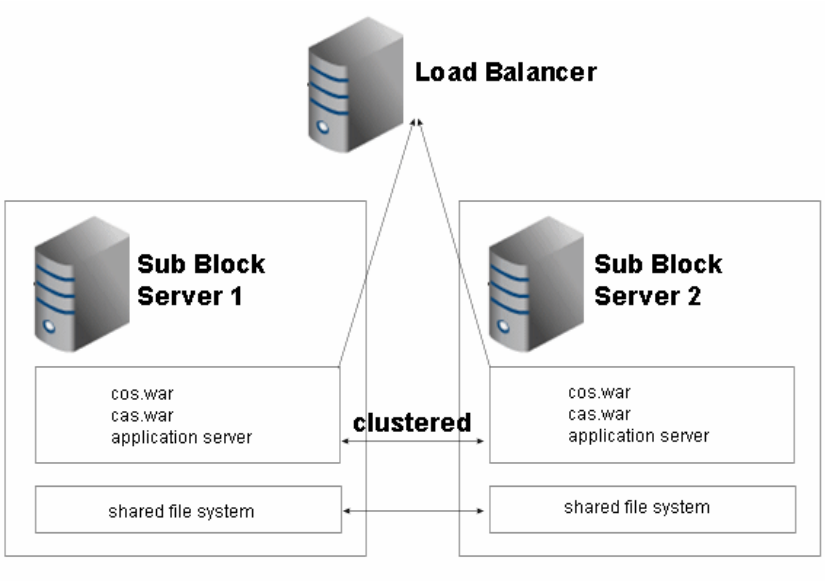


Figure 5 shows a clustered version of the production stack illustrated in Figure 3, on page 11. In Figure 5, the Community application (Block A) is now clustered. The WebCenter Sites application (Block C) is also clustered and uses a database cluster.

Figure 6: Community Application Production Block Consisting of Two Sub-Blocks



In Figure 6, the Community production block (Block A in Figure 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 7, on page 16).

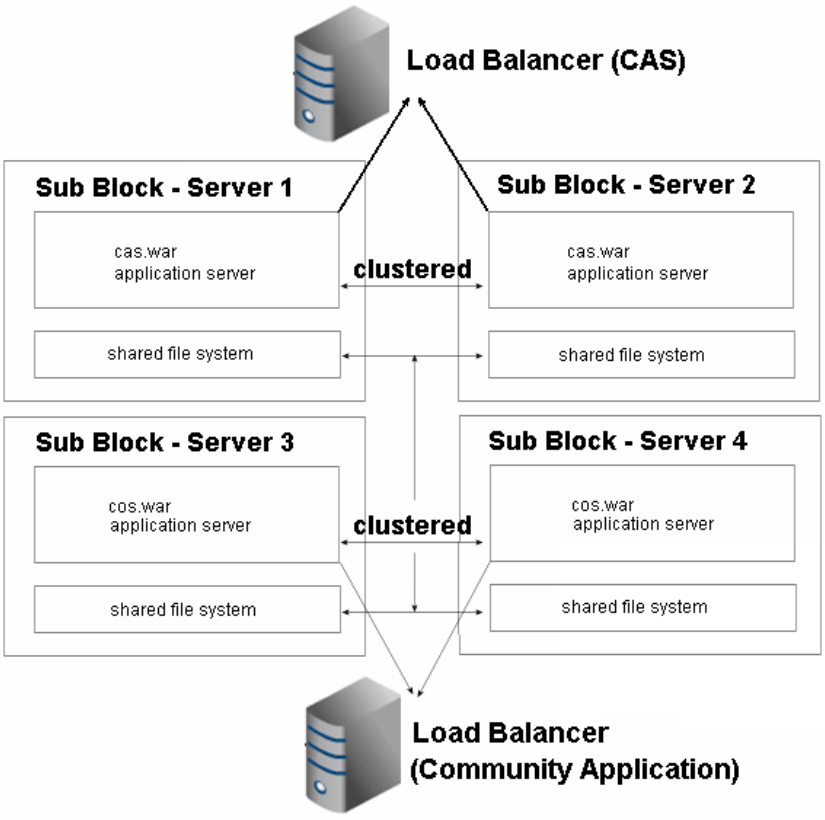
- Critical to the functionality of each sub-block is the fact that the application servers are fully clustered and include session failover.
- The file system is duplicated on each instance, as each cluster member requires unique configuration files for both Community and CAS.

Note

The example in Figure 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 7: Community Application Production Block Consisting of Four Sub-Blocks



In Figure 7, the Community production block (Block A in Figure 3) is divided into four sub-blocks, on four servers. The Community production block consists of two independent clusters: one for `cas.war` and one for `cos.war`. While this kind of breakdown is possible, it is not recommended. Typically, a cluster of two sub-blocks, as shown in Figure 6, provides the required capabilities and failover, and with less administrative overhead.

- In Figure 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 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.

- Two load balancers have been introduced: one for the Community application and one for Community visitor CAS.

Chapter 2

Prerequisites

This chapter provides prerequisites for installing the WebCenter Sites: Community application and lists the path and directory naming conventions used in this guide.

This chapter contains the following sections:

- Before Installing WebCenter Sites: Community
- Path and Directory Naming Conventions

Before Installing WebCenter Sites: Community

Installing WebCenter Sites: Community requires expertise with WebCenter Sites, the WEM Framework, application servers, and the process of installing and configuring enterprise-level software. Before installing the Community application, complete the prerequisite steps in the sections below.

Prerequisites for All Installations

- Review the *Oracle WebCenter Sites Certification Matrix* and Community application release notes for the latest information about platforms for the Community application and its installation.
- Read this guide to determine your installation options:
  - The Community application can be installed to work with either its native identity provider (the database used by the WEM Framework) or an LDAP identity provider.
  - The Community application can be installed via the graphical installer or silent installer.
  - The Community application can be installed as a single node or in a cluster.
  - All mention of the Central Authentication Service (CAS) application refers to visitor CAS (on the production server).
- Ensure that the Community application has a dedicated application server for each management node and each production node.

  Note

  Installing the Community application on a single physical server is not supported.

- The Community application requires JDK 1.6. Ensure that the `JAVA_HOME` environment variable is set to the path of JDK 1.6.
- Ensure that you have a fully functional WebCenter Sites deployment, consisting of a content management system and a production (delivery) system.

  Note

  The Community application uses the WebCenter Sites: Web Experience Management (WEM) Framework and CAS to communicate with WebCenter Sites.

- On the WebCenter Sites management system:
  - Create or select a content management site, `{cs_site}`, to which you will assign the Community interface once the Community application is installed and

registered. Mirror publish the site to the WebCenter Sites production system (before installing the Community application).

Note

Identical sites are required to support communication between the production and management Community applications as follows: On the production side, `{cs_site}` will be used to store visitor-generated content. The same site on the management side displays visitor-generated content in the Community interface for moderation and related tasks.

Assigning the management Community interface to `{cs_site}` on the WebCenter Sites management system automatically assigns the production Community applications to `{cs_site}` on the WebCenter Sites production system. The production Community applications can then communicate with the management Community application.

- On the WebCenter Sites production system:
  - Enable WebCenter Sites for millisecond date format (required by the Community application) by adding the `-Dcs.useMilliseconds=true JVM` parameter to `JAVA_OPTS` on the WebCenter Sites application server.
  - Enable the WebCenter Sites inCache framework (required by the Community application) by adding the following JVM parameters to `JAVA_OPTS` on the WebCenter Sites application server:

    ```
    -Dcs.useEhcache=true
    -Dnet.sf.ehcache.enableShutdownHook=true
    ```
  - Enable searches as follows: Start the Lucene search engine. Configure search indexing, using the steps in the "Public Site Search" section of the *WebCenter Sites Developer's Guide*.
  - In the `futuretense.ini` file, add the following values:

    ```
    rsCacheOverInCache=true
    cc.PageCSz=2
    cc.PageTimeout=1
    ```

Prerequisites for Clustered Environments

If installing a clustered environment application, ensure you have the load balancer configured.

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 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_cos_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>
```

Path and Directory Naming Conventions

This guide uses the following paths and directory conventions (see Table 1):

Table 1: Path and Directory Names Used in This Guide

Convention	Description
`<cos_install_dir>`	Path to the directory where the Community application will be installed.
`<cos_install_extracted_dir>`	Path to the directory in which the Community application is extracted.
`<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 physical server ‘A’ with unique host name and IP address.
`<ServerB>`	Represents physical server ‘B’ with unique host name and IP address.

Table 1: Path and Directory Names Used in This Guide

Convention	Description
<cosA_M*i*>	Path to the application server on which the management instance of the Community application will be deployed. • 'A' stands for the physical server on which the application server is installed; in this case, <ServerA>. • 'M' stands for the management instance of the Community application. • '*i*' stands for the *i*th instance of the management Community application
<cosB_P*i*>	Path to the application server instance on which the production instance of the Community application and visitor CAS will be deployed. • 'B' stands for the physical server on which the application server is installed; in this case, <ServerB>. • 'P' stands for the production instance of the Community application. • '*i*' stands for the *i*th instance of the production Community application
{cs_site}	The content management site to which you will assign the management instance of the Community application.

Chapter 3

Configuring Application Servers

This chapter contains procedures for configuring application servers to support non-clustered and clustered deployments of the Community application.

This chapter contains the following sections:

- Configuring for Non-Clustered Deployments
- Configuring for Clustered Deployments

Overview

The Community 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 the Community application on two servers. The management Community application is deployed on `<ServerA>` and the production Community application is deployed on `<ServerB>`.
- Clustered deployments – WebCenter Sites: Community can be deployed as either a vertically or horizontally clustered application. In this guide, we vertically cluster the Community application as follows: The management Community application is deployed as a cluster on `<ServerA>`, and the production Community application is deployed as a cluster on `<ServerB>`. Horizontal clustering can be achieved by replicating the above configuration on as many additional servers as needed.

Configuring for Non-Clustered Deployments

This section provides steps for configuring selected application servers to support the deployment of a non-clustered Community application.

This section covers the following configurations:

- Configuring Apache Tomcat
- Configuring IBM WebSphere Application Server
- Configuring Oracle WebLogic Application Server

Note

In this section, you will configure an application server to support a non-clustered Community application. As discussed in "Introduction," on page 7, 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 application on two servers, as explained in the "Overview" of this chapter.

Configuring Apache Tomcat

In this section, you will configure Apache Tomcat to support deployment of a non-clustered Community application, as follows:

1. On `<ServerA>`:
   a. Create a Tomcat instance (`<cosA_M1>`) on which to deploy the management Community application.
   b. Edit `catalina.sh` in `<cosA_M1>/bin` by adding the following line after the first comment block:

      `CATALINA_HOME=`*`path_to`*`_<cosA_M1>`
2. On `<ServerB>`:
   a. Create a Tomcat instance (`<cosB_P1>`) on which to deploy the production Community application.

b. Edit `catalina.sh` in `<cosB_P1>/bin` by adding the following line after the first comment block:

`CATALINA_HOME=`*`path_to`*`_<cosB_P1>`

3. Continue to "Installing Oracle WebCenter Sites: Community," on page 39.

Configuring Oracle WebLogic Application Server

In this section, you will configure WebLogic Application Server to support deployment of a non-clustered Community 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 non-clustered, management Community application, do the following:

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 following configurations (see Table 2 below):

Table 2: WebLogic Configuration Example for the Management Community Application

Configuration	Property	Value
Domain	Domain Name	`<cosMgmt>`
	Admin Server	`<ServerA>`
	Admin Server Port	For example: `7001` Or use your own value as appropriate for your configuration.
Machine	Machine Name	`<wlcosA_M>`
	Listen Address	`<ServerA>`
	Listen Port	For example: `5556` Or use your own value as appropriate for your configuration.
Managed Server	Managed Server	`<cosA_M1>`
	Domain	`<cosMgmt>`

Table 2: WebLogic Configuration Example for the Management Community Application

Configuration	Property	Value
	Listen Address	<ServerA>
	Listen Port	For example: 7003 Or use your own value as appropriate for your configuration.
	Machine	<wlcosA_M>

2. Similarly, configure WebLogic Application Server to support deployment of a non-clustered, production Community 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 following configurations (see Table 3):

Table 3: WebLogic Configuration Example for the Production Community Application

Configuration	Property	Value
Domain	Domain Name	<cosProd>
	Admin Server	<ServerB>
	Admin Server Port	For example: 7001 Or use your own value as appropriate for your configuration.
Machine	Machine Name	<wlcosB_P>
	Listen Address	<ServerB>
	Listen Port	For example: 5556 Or use your own value as appropriate for your configuration.
Managed Server	Managed Server	<cosB_P1>
	Domain	<cosProd>
	Listen Address	<ServerB>
	Listen Port	For example: 7003 Or use your own value as appropriate for your configuration.
	Machine	<wlcosB_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 application. In this guide, we created a directory named `applications` in the following paths:

```
<WL_HOME>/user_projects/domains/cosMgmt/applications
<WL_HOME>/user_projects/domains/cosProd/applications
```

5. Continue to "Installing Oracle WebCenter Sites: Community," on page 39.

Configuring IBM WebSphere Application Server

In this section, you will configure IBM WebSphere Application Server to support deployment of a non-clustered Community 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 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 following configurations (see Table 4):

Table 4: WebSphere Configuration Example for the Management Community Application

Configuration	Property	Value
Profile	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.
Application Server	Application Server Name	`<cosA_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 application, and federate that node to your Deployment Manager.

   For reference, this guide uses the following configurations (see Table 5):

   Table 5: WebSphere Configuration Example for the Production Community Application

Configuration	Property	Value
Profile	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.
Application Server	Application Server Name	<cosB_P1>
	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 "Installing Oracle WebCenter Sites: Community," on page 39.

Configuring for Clustered Deployments

This section describes the configuration of supported application servers to support clustered deployments of the Community application.

This section covers these clustered deployment configurations:

- Configuring Apache Tomcat (Clustered)
- Configuring Oracle WebLogic Application Server (Clustered)
- Configuring IBM WebSphere Application Server (Clustered)

Note

In this section, you will configure an application server to support a clustered Community installation. As discussed in "Introduction," on page 7, you have multiple ways to configure a clustered system. For illustration purposes, the steps in this section describe vertical clustering of the Community application, as explained in the "Overview" of this chapter.

Configuring Apache Tomcat (Clustered)

In this section, you will configure Apache Tomcat to support deployment of a clustered Community application.

1. On `<ServerA>`:
   a. Create a Tomcat instance (`<cosA_M1>`) on which to deploy the management Community application. Edit `catalina.sh` in `<cosA_M1>/bin` by adding the following line after the first comment block:

      `CATALINA_HOME=`*`path_to`*`_<cosA_M1>`

   b. Create a Tomcat instance (`<cosA_M2>`) on which to deploy the management Community application. Edit `catalina.sh` in `<cosA_M2>/bin` by adding the following line after the first comment block:

      `CATALINA_HOME=`*`path_to`*`_<cosA_M2>`

   c. Edit the `server.xml` file ensure there are no port conflicts.
2. On `<ServerB>`:
   a. Create a Tomcat instance (`<cosB_P1>`) on which to deploy the production Community application. Edit `catalina.sh` in `<cosB_P1>/bin` by adding the following line after the first comment block:

      `CATALINA_HOME=`*`path_to`*`_<cosB_P1>`

   b. Create a Tomcat instance (`<cosB_P2>`) on which to deploy the production Community application. Edit `catalina.sh` in `<cosB_P2>/bin` by adding the following line after the first comment block:

      `CATALINA_HOME=`*`path_to`*`_<cosB_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 this example (see Table 6):

Table 6: Tomcat Configuration Example for the Community Application

Host	Tomcat Instance	Port Number
<ServerA>	<cosA_M1>	8080
	<cosA_M2>	8081
<ServerB>	<cosB_P1>	8080
	<cosB_P2>	8081

3. Configure the application server cluster. Follow the steps below for <ServerA> and <ServerB>.

   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="4180"
    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>
```

4. 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, `<cosB_P1>` and `<cosB_P2>` must have the same values for multicast address and port.
5. Continue to “Installing Oracle WebCenter Sites: Community,” on page 39.

Configuring Oracle WebLogic Application Server (Clustered)

In this section, you will configure WebLogic Application Server to support deployment of a clustered Community 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 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 following configurations (see Table 7):

   Table 7: WebLogic Configuration Example for the Management Community Application (Clustered)

Configuration	Property	Value
Domain	Domain Name	`<cosMgmt>`
	Admin Server	`<ServerA>`
	Admin Server Port	For example: `7001` Or use your own value as appropriate for your configuration.
Machine	Machine Name	`<wlcosA_M>`
	Listen Address	`<ServerA>`
	Listen Port	For example: `5556` Or use your own value as appropriate for your configuration.
Managed Server	Managed Server 1	`<cosA_M1>`
	Domain	`<cosMgmt>`
	Listen Address	`<ServerA>`
	Listen Port	For example: `7003` Or use your own value as appropriate for your configuration.

Table 7: WebLogic Configuration Example for the Management Community Application (Clustered)

Configuration	Property	Value
	Machine	<wlcosA_M>
	Managed Server 2	<cosA_M2>
	Domain	<cosMgmt>
	Listen Address	<ServerA>
	Listen Port	For example: 7005 Or use your own value as appropriate for your configuration.
	Machine	<wlcosA_M>
Cluster	Cluster Name	<wlcos_M >
	Cluster Address	<ServerA>
	Cluster Messaging Mode	Unicast (or Multicast depending on your environment)
	Managed Servers as part of Cluster	<cosA_M1>, <cosA_M2>

2. Similarly, configure WebLogic Application Server to support the deployment of a clustered production Community 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 following configurations (see Table 8):

Table 8: WebLogic Configuration Example for the Production Community Application (Clustered)

Configuration	Property	Value
Domain	Domain Name	<cosProd>
	Admin Server	<ServerB>
	Admin Server Port	For example: 7001 Or use your own value as appropriate for your configuration.
Machine	Machine Name	<wlcosB_P>
	Listen Address	<ServerB>
	Listen Port	For example: 5556 Or use your own value as appropriate for your configuration.
Managed Server	Managed Server 1	<cosB_P1>

Table 8: WebLogic Configuration Example for the Production Community Application (Clustered)

Configuration	Property	Value
	Domain	<cosProd>
	Listen Address	<ServerB>
	Listen Port	For example: 7003 Or use your own value as appropriate for your configuration.
	Machine	<wlcosB_P>
	Managed Server 2	<cosB_P2>
	Domain	<cosProd>
	Listen Address	<ServerB>
	Listen Port	For example: 7005 Or use your own value as appropriate for your configuration.
	Machine	<wlcosB_P>
Cluster	Cluster Name	<wlcos_P >
	Cluster Address	<ServerB>
	Cluster Messaging Mode	Unicast (or Multicast depending on your environment)
	Managed Servers as part of Cluster	<cosB_P1>, <cosB_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/cosMgmt/applications
   <WL_HOME>/user_projects/domains/cosProd/applications
   ```
5. For each cluster, go to its "Advanced Cluster Configuration" section and select the **WebLogic Plug-In Enabled** parameter.
6. Continue to "Installing Oracle WebCenter Sites: Community," on page 39.

Configuring IBM WebSphere Application Server (Clustered)

In this section, you will configure WebSphere Application Server to support deployment of a clustered Community 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 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 following configurations (see Table 9):

   Table 9: WebSphere Configuration Example for the Management Community Application (Clustered)

Configuration	Property	Value
Profile	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.
Application Server	Application Server Name	<cosA_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	<cosA_M2>
	Profile	<AppSrv01>
	Node	<ServerA_Node01>
	Listen Address	<ServerA>

Table 9: WebSphere Configuration Example for the Management Community Application (Clustered)

Configuration	Property	Value
	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 application, and federate that node to your Deployment Manager.

   For reference, this guide uses the following configurations (see Table 10):

Table 10: WebSphere Configuration Example for the Production Community Application (Clustered)

Configuration	Property	Value
Profile	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.
Application Server	Application Server Name	`<cosB_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 Name	`<cosB_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 11), using the WebSphere Administration Console.

   For example:

   `http://<ServerA>:9060/ibm/console`

   Table 11: WebSphere Cluster Configuration Example for the Community Application

Configuration	Property	Value
Cluster 1	Cluster Name	`<wscos_M>`
	Configure HTTP memory-to-memory replication	`Yes`
	Cluster Members	`<cosA_M1>, <cosA_M2>`
Cluster 2	Cluster Name	`<wscos_P>`
	Configure HTTP memory-to-memory replication	`Yes`
	Cluster Members	`<cosB_P1>, <cosB_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** > **Replication domains** in the left frame).
6. Add ports for all cluster members (go to **Environment** > **Virtual hosts** > **default_host** > **Host Aliases** and add the ports).
7. Continue to "Installing Oracle WebCenter Sites: Community," on page 39.

Chapter 4

Installing Oracle WebCenter Sites: Community

This chapter describes how to install the Community application both graphically and silently and provides post-installation steps.

This chapter contains the following sections:

- Installing Graphically
- Installing Silently
- Community Application Directory Structure
- Post-Installation Steps

Installing Graphically

Follow these steps to install the Community application graphically:

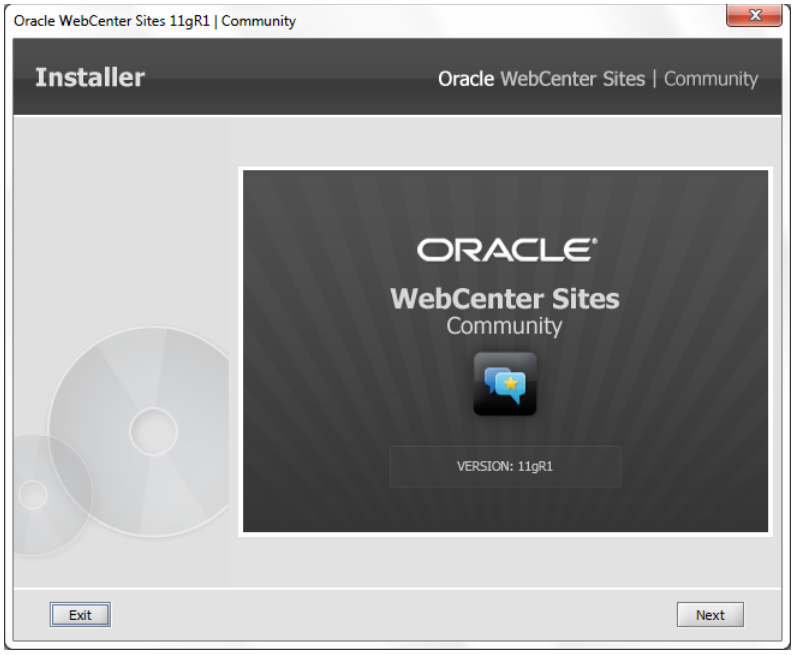
1. Download the Community distribution `zip` file to your server and extract it into a temporary directory. The extracted folder contains the `cosInstall.bat` and `cosInstall.sh` files, which are needed to run the Community installer.
2. Create an installation directory in which the Community installer will create the necessary folders and files.
3. Set `JAVA_HOME` to the path of JDK 1.6.

   For example:

   `export JAVA_HOME=/opt/jdk1.6.0_20`
4. On UNIX, set the `DISPLAY` environment variable.

   For example:

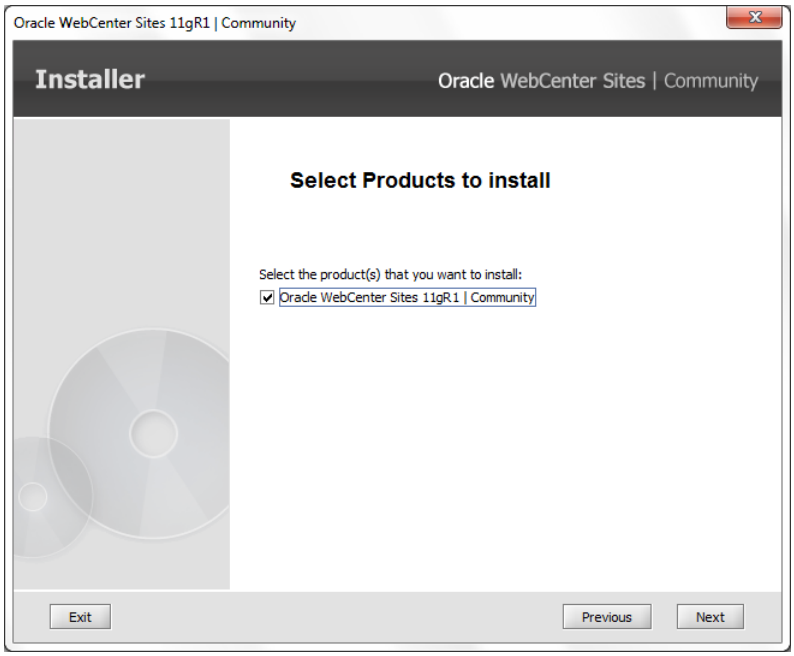
   `export DISPLAY=10.120.15.20:0.0`
5. Execute the installer script:
   - On Windows: `cosInstall.bat`
   - On UNIX: `cosInstall.sh`
6. The first screen displayed to you is the welcome screen. Click **Next**.



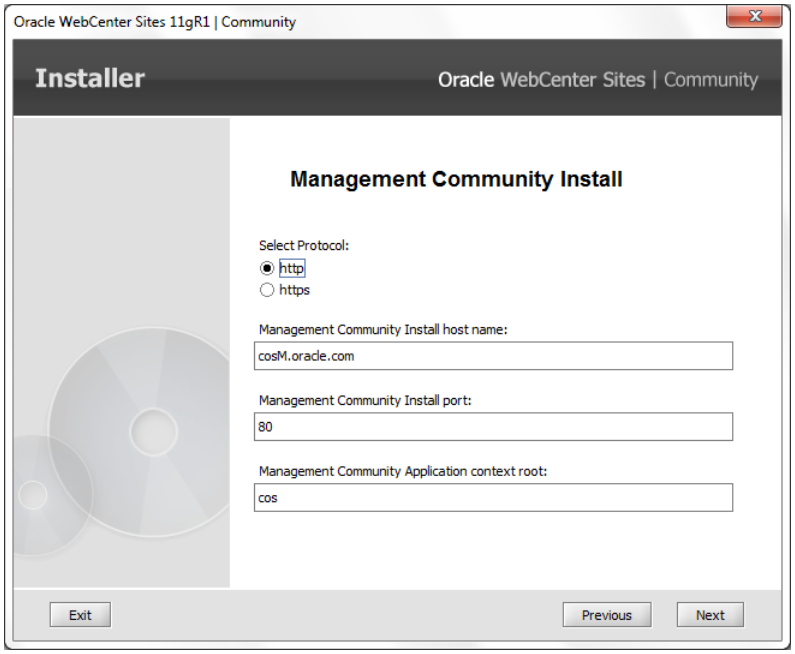
7. On the “Installation Directory” screen, specify the path to the directory (created in step 2 on page 40) where the Community application will be installed, or click **Browse** and navigate to a directory.

   The path must not contain spaces. If the directory you specify does not exist, the installer creates it.

8. On the “Select Products to Install” screen, select the **Oracle WebCenter Sites 11gR1 Community** option.

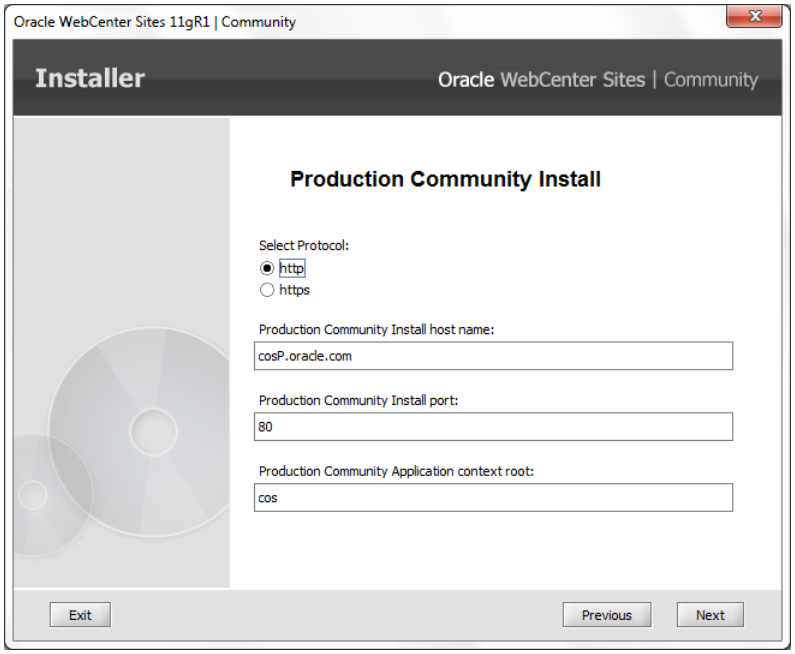


9. On the “Management Community Install” screen:
   - Select the **http** or **https** (HTTP over SSL) protocol.
   - Enter the host name or IP address of the management Community application server. If installing a cluster, enter the host name or IP address of the load balancer.
   - Enter the port number of the management Community application server. If installing a cluster, enter the port number of the load balancer.
   - Enter the management Community application context root.



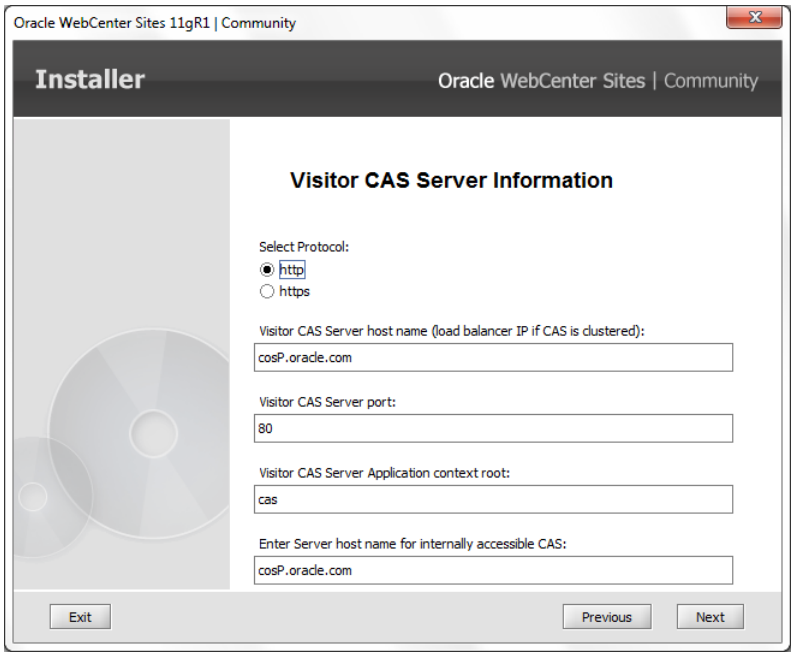
10. On the “Production Community Install” screen:

- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the production Community 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 application server. If installing a cluster, enter the port number of the load balancer.
- Enter the production Community application context root.

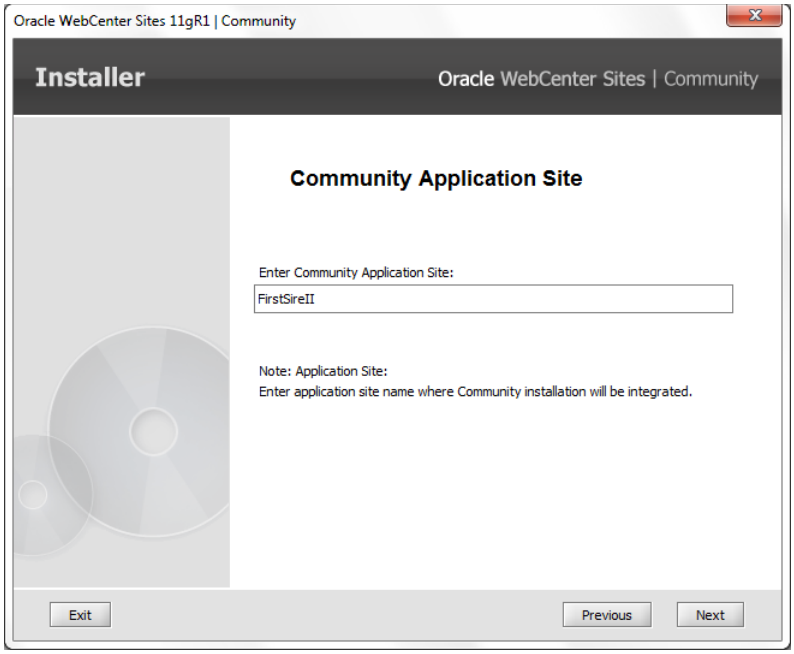


11. On the "Visitor CAS Server Information" screen:

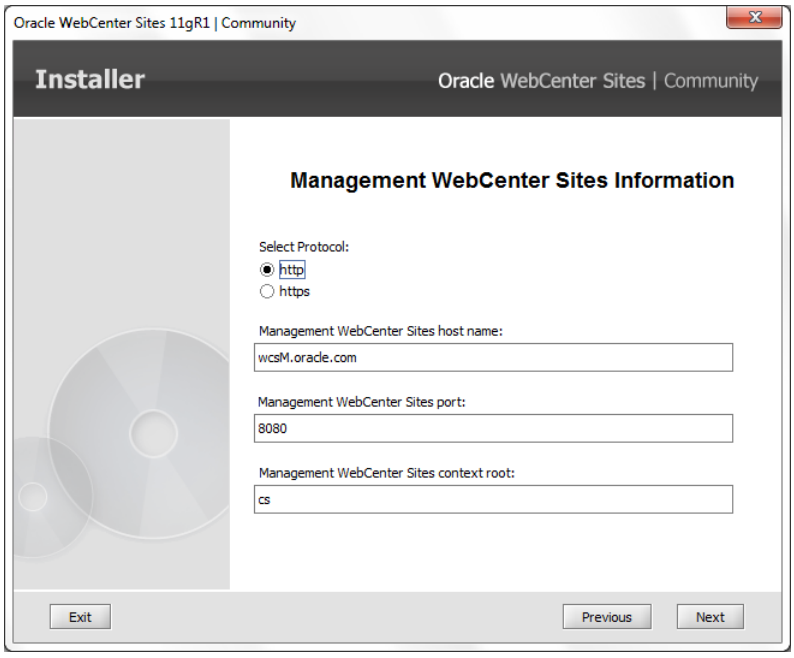
- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the visitor CAS application. If installing a cluster, enter the host name or IP address of the load balancer.
- Enter the port number of the visitor CAS application. If installing a cluster, enter the port number of the load balancer.
- Enter the visitor CAS application context root.
- Enter the host name or IP address of the internally accessible visitor CAS application.



12. On the “Community Application Site” screen, enter the name of `{cs_site}` to which you will assign the management Community application once it is installed.



13. On the “Management WebCenter Sites Information” screen:
    - Select the **http** or **https** (HTTP over SSL) protocol.
    - Enter the host name or IP address of the management WebCenter Sites application server.
    - Enter the port number of the management WebCenter Sites application server.
    - Enter the management WebCenter Sites application context root.



14. On the “Management WebCenter Sites Admin User” screen:

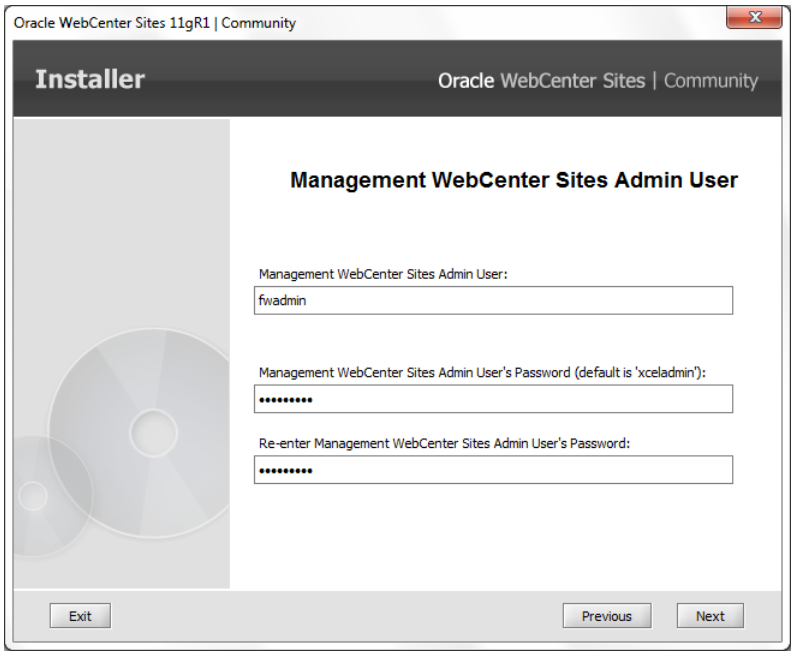
- Enter your general administrator user name, which you use to log in to the WebCenter Sites management system.

  The user name is case-sensitive and allows from 8 to 64 alphanumeric characters. The default user is: `fwadmin`

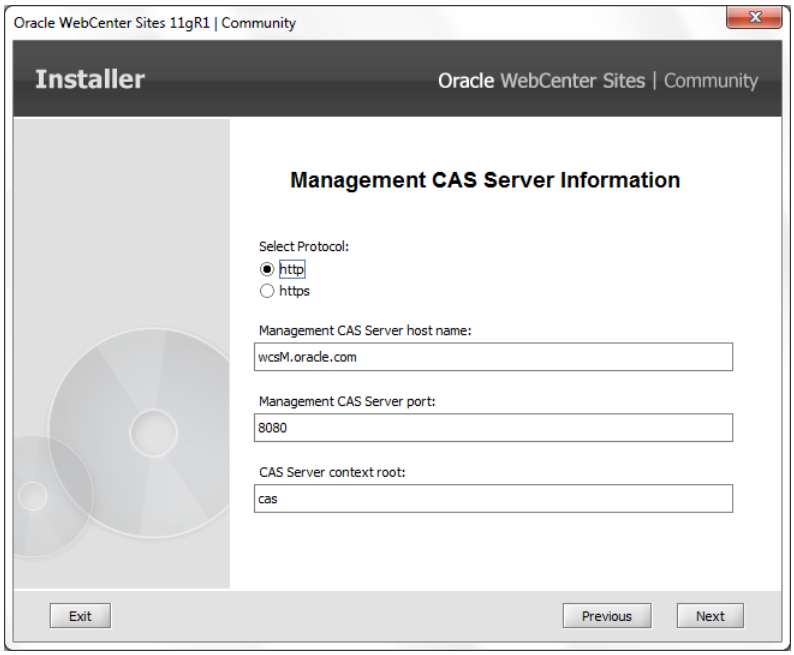
- Enter your administrator password.

  The password is case-sensitive and allows from 8 to 64 alphanumeric characters.

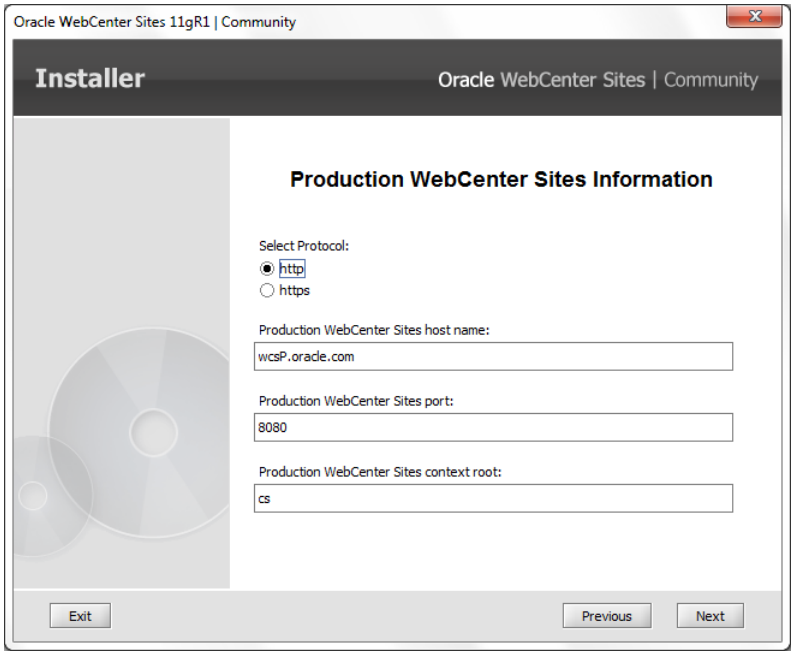
- For verification re-enter your password.



15. On the “Management CAS Server Information” screen:
    - Select the **http** or **https** (HTTP over SSL) protocol.
    - Enter the host name or IP address of the management CAS server.
    - Enter management CAS server port.
    - Enter the management CAS application context root.



16. On the “Production WebCenter Sites Information” screen:
    - Select the **http** or **https** (HTTP over SSL) protocol.
    - Enter the host name or IP address of the production WebCenter Sites application server.
    - Enter the port number of the production WebCenter Sites application server.
    - Enter the production WebCenter Sites application context root.



17. On the “Production WebCenter Sites Admin User” screen:

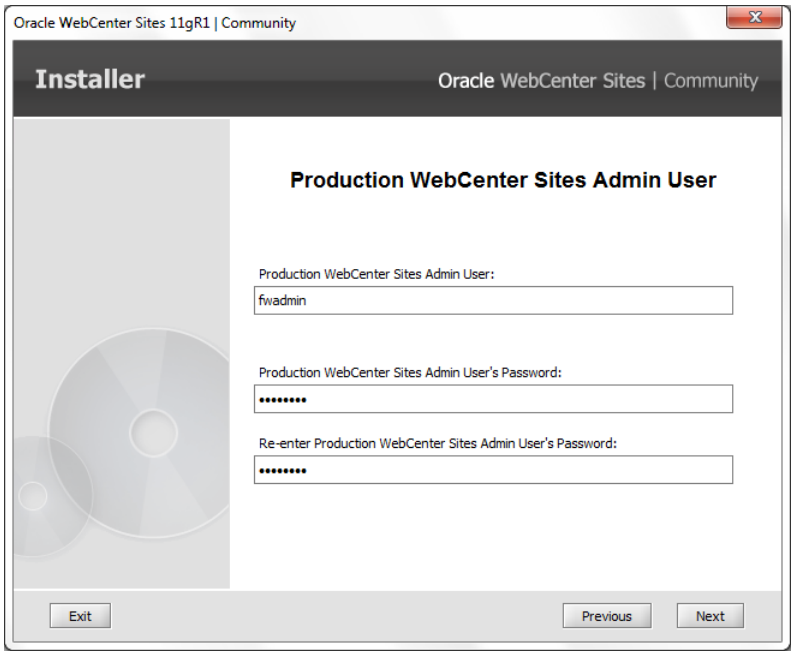
- Enter your general administrator user name, which you use to log in to the WebCenter Sites production system.

  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.



18. On the “Production WebCenter Sites Satellite User” screen:

- Enter the user name for the Satellite Server that is configured for the production WebCenter Sites application.

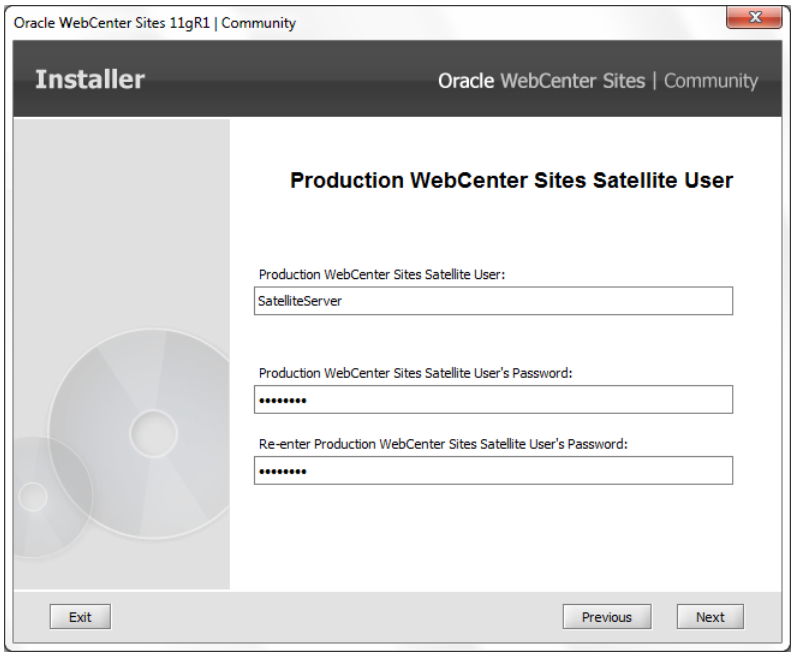
  The user name is case-sensitive and allows from 8 to 64 alphanumeric characters. The default user is: `SatelliteServer`

- Enter the Satellite Server user’s password.

  The password is case-sensitive and allows from 8 to 64 alphanumeric characters.

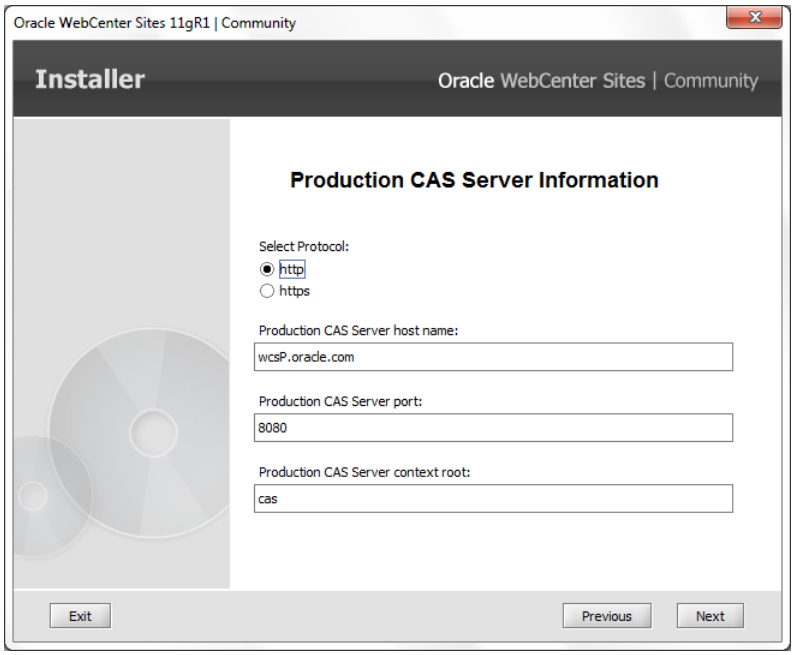
- For verification, re-enter the password.

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.



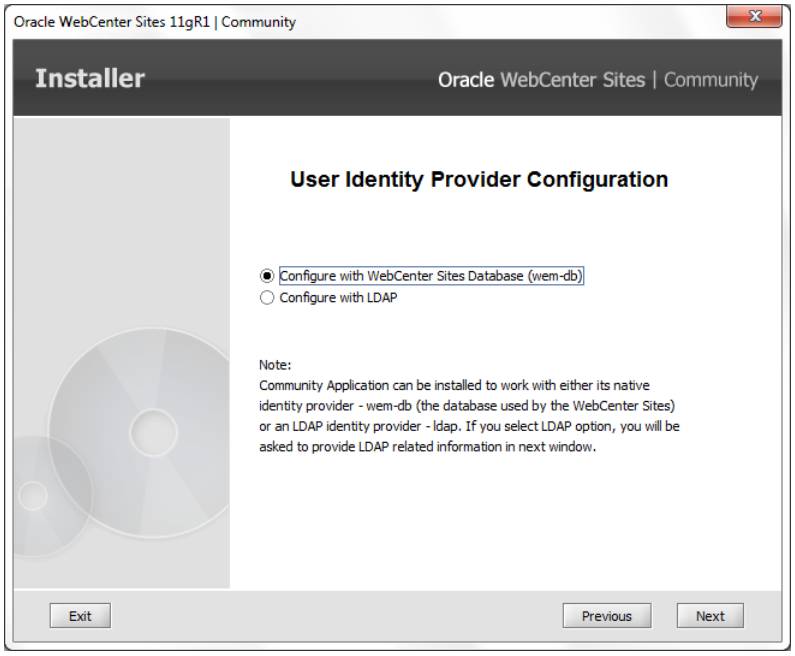
19. On the “Production CAS Server Information” screen:

- Select the **http** or **https** (HTTP over SSL) protocol.
- Enter the host name or IP address of the production CAS server.
- Enter the port on which the production CAS listens.
- Enter the production CAS application context root.

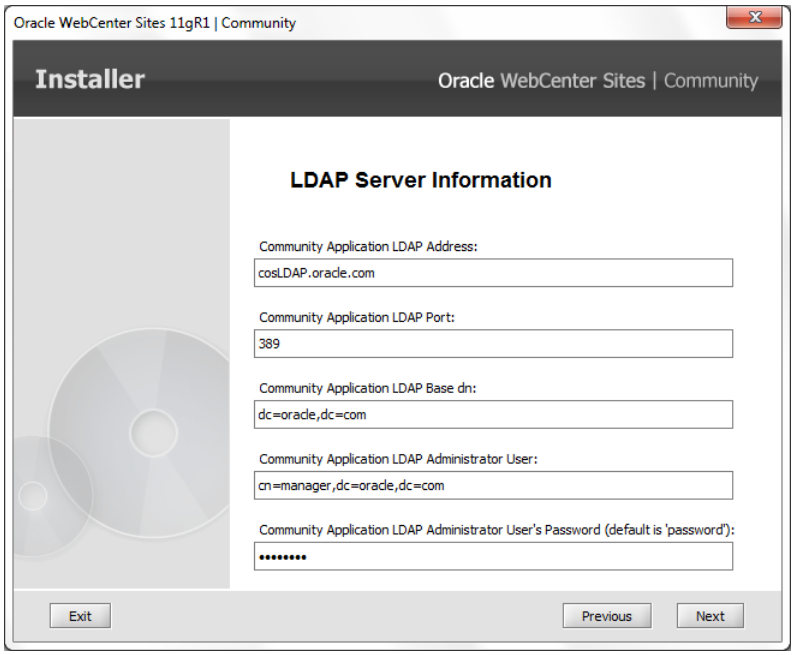


20. On the "User Identity Provider Configuration" screen, select one of these options:

- **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**.



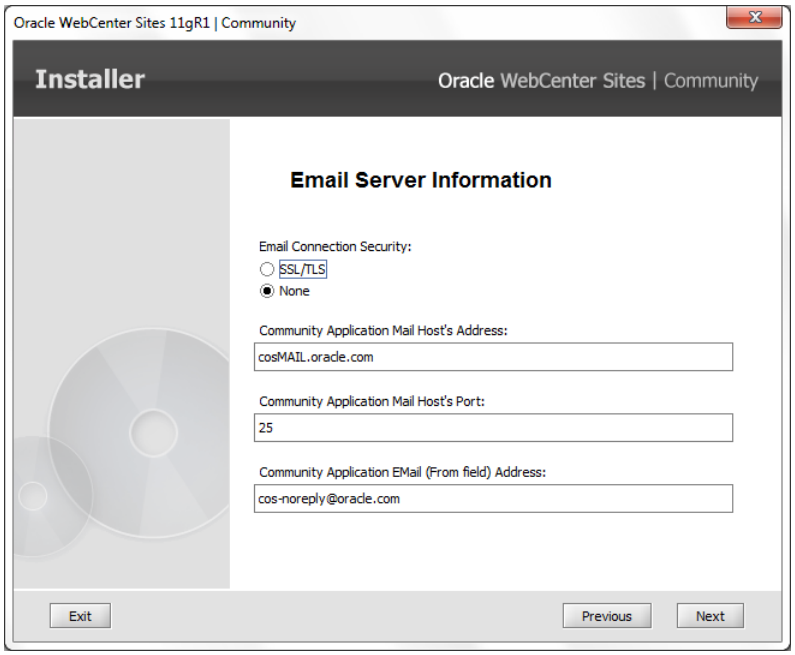
21. If you selected **Configure with LDAP** in the previous step, provide the following values on the "LDAP Server Information" screen (for examples, see "Prerequisites for LDAP Configuration," on page 19).
    - LDAP server host name or IP address.
    - LDAP server port.
    - LDAP server Base DN.
    - LDAP administrator user.
    - LDAP administrator's password.



22. On the “Email Server Information” screen:

- Select **SSL/TLS** or **None** for email connection security.
- Enter the email server host name or IP address.
- Enter the email server port.
- Enter the email address to be used in the “From” field.

> **Note**
>
> Email server information is used by the Community application to communicate with visitors.

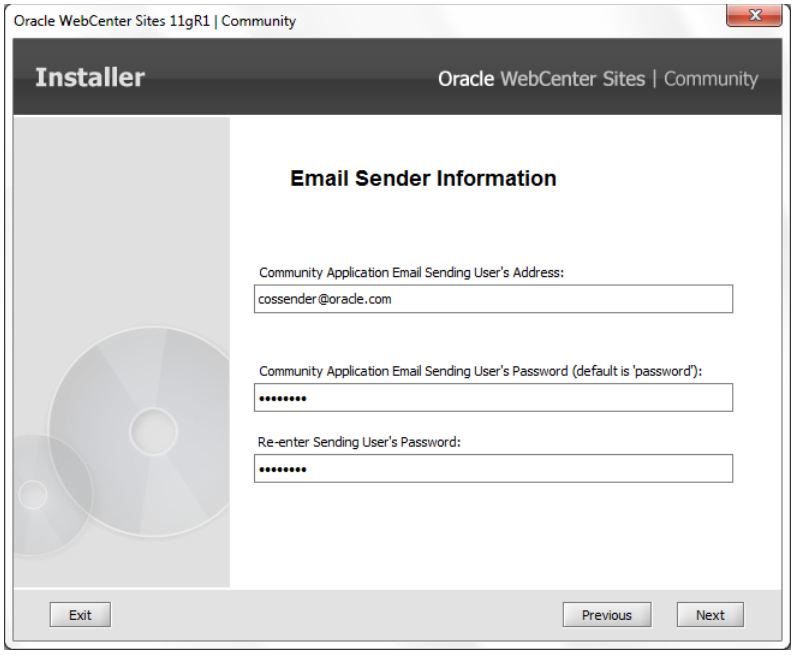


23. On the “Email Sender Login” screen, select one of these options:

- **Yes**: Login required to send mail. If you select this option, the “Email Sender Information” screen is displayed when you click **Next**.
- **No**: Login not required to send mail.

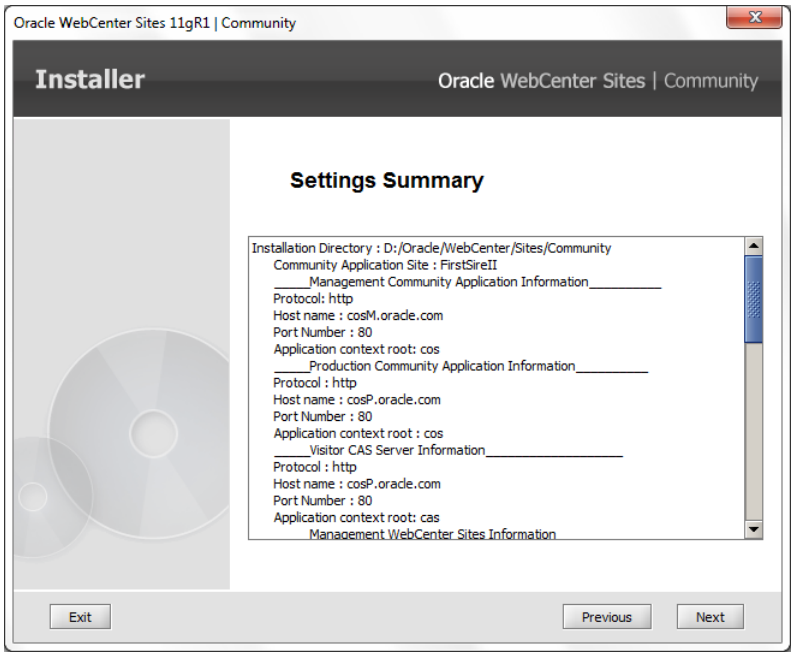
24. If you selected **Yes** in the previous step, provide these values on the "Email Sender Information" screen:

- Email address of the sending user.
- Password of the sending user.
- For verification, re-enter the password of the sending user.

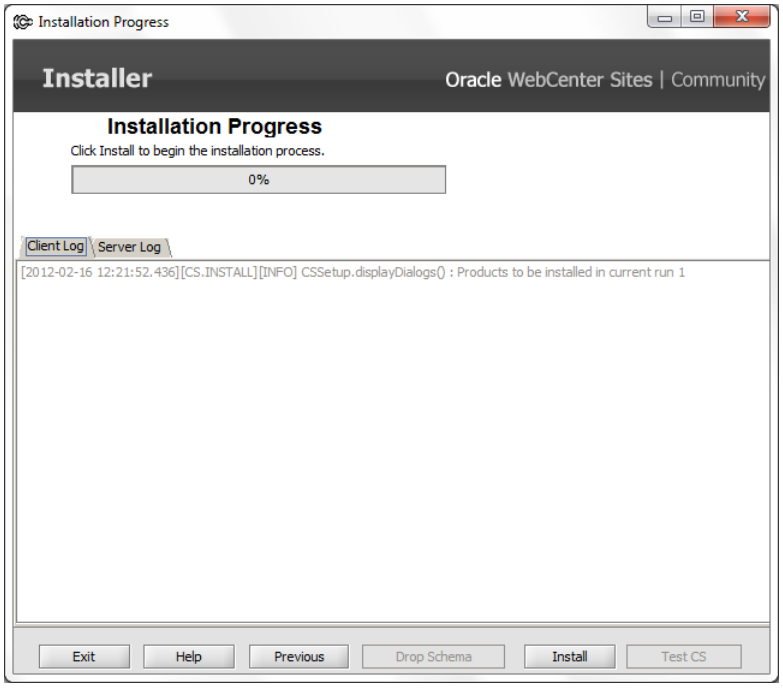


25. The “Settings Summary” screen summarizes the configuration choices you have made for this installation.

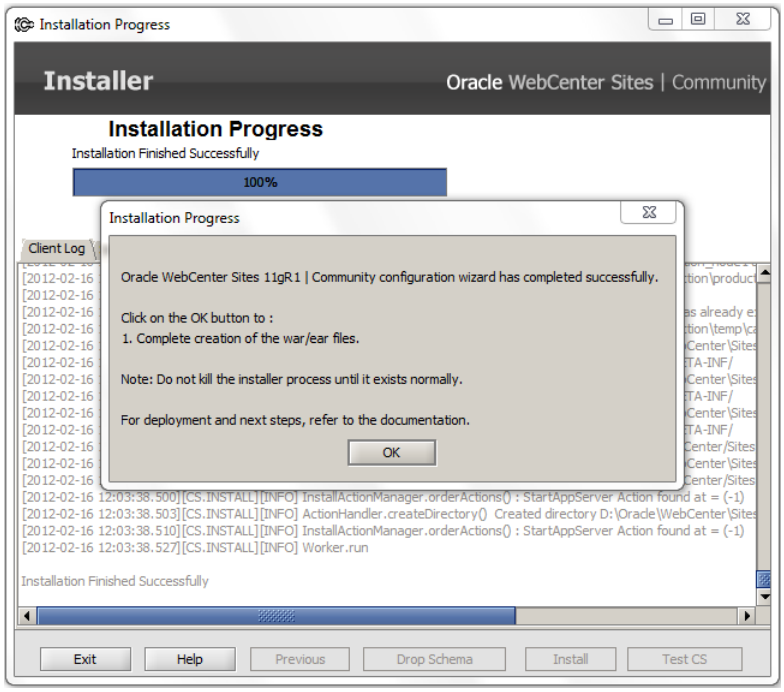
- Review the settings summary to make sure all options are configured correctly.
- If you find an option that needs to be modified, click **Previous** to return to the installer screen containing that option.



26. On the "Installation Progress" screen, click **Install** to start the installation process. You can monitor the installation process from the **Client Log** tab.



27. When the installation is completed successfully, a confirmation note is displayed. Click **OK** to create the `war` file for the Community web application. **Do not end the installer process.** Allow it to exit normally.



28. Before deploying the Community application, review the Community Application Directory Structure" and complete "Post-Installation Steps," on page 61.

Installing Silently

Follow these steps to install the Community application silently:

1. Run the graphical Community installer to generate the `omii.ini` file. For instructions, see "Installing Graphically," on page 40.
2. Copy the `omii.ini` file to the server on which you will install the Community application.
3. Extract the Community installer on the server where you will install the Community application.
4. Edit the `install.ini` file (in the extracted Community installer) as follows:
   - Set `loadfile=<path to installer directory>/omii.ini`

     (to point to the `omii.ini` file in step 2).
   - Set `nodisplay=true`
   - Set `productautoselect=true`
5. Create an installation directory in which the Community installer will create the necessary folders and files.

   For example:

   `/u01/software/Apps/CoS`
6. Set `JAVA_HOME` to the path of JDK 1.6.

   For example:

   `export JAVA_HOME=/opt/jdk1.6.0_20`
7. Run the Community silent installer:
   a. Run the `cosInstall.sh` or `cosInstall.bat` script.
   b. Wait until the `"Installation Finished Successfully"` message appears and the process is complete.
8. Before deploying the Community application, review the section "Community Application Directory Structure," on page 60 and complete "Post-Installation Steps," on page 61.

Community Application Directory Structure

Once the Community installation process is complete, the installation directory has the following file structure (the main directories are described below):

```
<cos_install_dir>
 |--cos_licenses
 | |--cas_thirdparty_licenses
 |--deploy
 | |--management
 | |  |--Community.ear
 | |  |--cos.war
 | |  |--management_node1
 | |     |--cas-cache.xml
 | |     |--cos-cache.xml
 | |     |--log4j-cos.properties
 | |--production
 |    |--cas.ear
 |    |--cas.war
 |    |--Community.ear
 |    |--cos.war
 |    |--production_node1
 |       |--cas.properties
 |       |--cos-cache.xml
 |       |--host.properties
 |       |--jbossTicketCacheReplicationConfig.xml
 |       |--log4j-cos.properties
 |       |--log4j.properties
 |--logs
 |--ominstallinfo
 | |--install_log.log
 | |--management_coswar.log
 | |--omii.ini
 | |--omproduct.ini
 | |--production_caswar.log
 | |--production_coswar.log
```

- `cos_licenses`: This directory contains the necessary license files.
- `deploy`: This directory contains the `war` and `ear` files created for the management and production sides of the Community application. The user is required to deploy the `war` /`ear` files manually.
- `logs`: This directory stores logs that will be created by the Community application during use.
- `ominstallinfo`: This directory contains information about the Community installation. The installation log and installation settings file are stored in this directory.

Post-Installation Steps

After successfully running the Community installer, follow the post-installation steps in this section.

A. 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 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 application and once for each production node of the Community 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>`
      - **host**: `<host name or IP address of the management community application or production Community application>`
      - **port**: `<port number of the management or production Community application server>`
      - **flushservletpath**: `/cos/incache/inCacheManager?command=invalidate&region=common`

        Note

        If the standard Community application context root is not used, use:

        `/<new context root>/incache/inCacheManager?command=invalidate&region=common`

      - **username**: `<of the production Satellite Server user>` (the default is `SatelliteServer`)
      - **password**: `<password of the above user>`
3. Continue to step "B. Copying Installer-Generated Files."

B. Copying Installer-Generated Files

The Community installer creates property files and other configuration files that will be read by the application server to start the deployed Community application correctly.

1. For ease of management and scalability, we recommend copying the files such that each instance of the Community application has its own set of files:
   a. In `<cos_install_dir>`, create a directory called `<cos-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 <cos_install_dir>/<cos-standalone-configs>
cd <cos_install_dir>/<cos-standalone-configs>
mkdir <cosA_M1> <cosA_M2> <cosB_P1> <cosB_P2> …
```

2. Copy the installer-generated configuration and property files to the node subdirectories as follows:

For each Management Community application node:

a. Copy the configuration files from `<cos_install_dir>/deploy/management/management_node1` to the subdirectories that you created in step 1b for the management Community application.

For example, in Linux:

```
cd <cos_install_dir>/deploy/management/management_node1
cp -p ./* <cos_install_dir>/<cos-standalone-configs>/
 <cosA_M1>
```

b. If needed, change the log file names by modifying `log4j-cos.properties` for each management Community application on a particular node.

For each Production Community application node:

a. Copy the configuration files from `<cos_install_dir>/deploy/production/production_node1` to the to the subdirectories that you created in step 1b (page 62) for the production Community application.

For example, in Linux:

```
cd <cos_install_dir>/deploy/production/production_node1
cp -p ./* <cos_install_dir>/<cos-standalone-configs>/
 <cosB_P1>
```

b. 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 visitor CAS cluster member. Ensure that all cluster members are listening on the same mcast address and port.

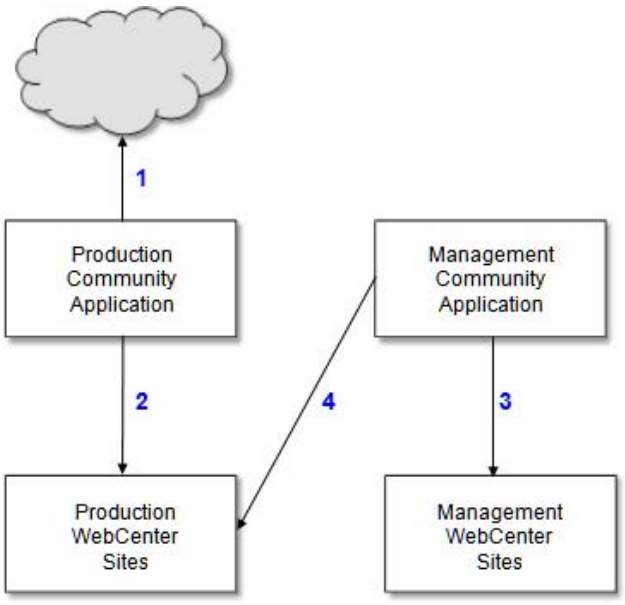
c. If needed, change the log file names by modifying `log4j-cos.properties` for each production Community application on a particular node.

3. If you are using proxy authentication, continue to step "C. Proxy Connection Settings." Otherwise, continue to Chapter 5, "Deploying Oracle WebCenter Sites: Community."

C. Proxy Connection Settings

The Community application supports up to four proxy connection settings (disabled by default) as shown in Figure 8.

Figure 8: Proxy Connection Settings Supported by the Community Application



- Proxy connection 1 is an Internet connection proxy that supports communication from the production Community application to the Internet. This proxy connection is used for external authentication via third-party services such as Facebook, Twitter, and Janrain.
- Proxy connection 2 supports communication from the production Community application to the production WebCenter Sites application.
- Proxy connection 3 supports communication from the management Community application to the management WebCenter Sites application.
- Proxy connection 4 supports communication from the management Community 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 `WEB-INF/classes` directory of your deployed Community application. The `setup_proxy.properties` file contains four 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 you wish to enable proxy authentication, go to the appropriate section, set `<proxy>.proxy.attrs.auth.required` to `true` and specify the password in its encrypted form. To encrypt the existing password, use `cos-password-encryptor.jar` provided by the Community installer.

For example, in Linux:

```
cd <cos_install_extracted_dir>/CommunityServer/
    FatwireProducts
java -jar cos-password-encryptor.jar
```

For reference a sample configuration section is given below:

Sample Internet connection proxy configuration (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.us.oracle.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. Should be in the encrypted
  form
cos-production_www.proxy.attrs.auth.password=
  J4Vi0Afe8NBMHixgOinSQVbGdyc=
```

2. Continue to Chapter 5, "Deploying Oracle WebCenter Sites: Community."

Chapter 5

Deploying Oracle WebCenter Sites: Community

This chapter describes the steps for deploying the Community application on supported application servers.

This chapter contains the following sections:

- Overview
- Deploying a Non-Clustered Community Application
- Deploying a Clustered Community Application

Overview

When deploying the Community application, you will do the following:

- On the management system, you will deploy only the management Community application.
- On the production system, you must deploy both the production Community application and visitor CAS (for authentication).

Instructions are available in this chapter.

Note

Before proceeding with this chapter, ensure you have completed all requirements in the previous chapters of this guide.

Deploying a Non-Clustered Community Application

This section describes the steps for deploying a non-clustered Community application to supported application servers.

This section covers the following deployment configurations:

- Deploying on Apache Tomcat
- Deploying on Oracle WebLogic Application Server
- Deploying on IBM WebSphere Application Server

Deploying on Apache Tomcat

In this section, you will deploy a non-clustered Community 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 application, ensure that each application server's classpath points to the respective Community configuration directories (created in section "B. Copying Installer-Generated Files," on page 61):

   For reference, this guide uses the following configurations (see Table 12):

   Table 12: Classpath Configuration for Tomcat Servers (Non-clustered)

Tomcat Server	Server CLASSPATH
<cosA_M1>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosA_M1>:${CLASSPATH}"

Table 12: Classpath Configuration for Tomcat Servers (Non-clustered)

Tomcat Server	Server CLASSPATH
<cosB_P1>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosB_P1>:${CLASSPATH}"

2. The Community installer creates `war` and `ear` files in the `<cos_install_dir>/deploy/management` directory and the `<cos_install_dir>/deploy/production` directory. Deploy the Community 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 13.

Table 13: `war` File Locations for Tomcat (Non-clustered)

Application Server Instance	Location of war Files	Sample Deployment Location
<cosA_M1>	<cos_install_dir>/deploy/management/cos.war	<cosA_M1>/webapps/cos
<cosB_P1>	<cos_install_dir>/deploy/production/cos.war <cos_install_dir>/deploy/production/cas.war	<cosB_P1>/webapps/cos <cosB_P1>/webapps/cas

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 2, "Prerequisites."
4. Start the production deployment servers before starting the management deployment servers.

   For example, in Linux:

   On <ServerB>:
   ```
   cd <cosB_P1>/bin
   ./catalina.sh run
   ```

   On <ServerA>:
   ```
   cd <cosA_M1>/bin
   ./catalina.sh run
   ```

5. Continue to "Verifying and Registering Oracle WebCenter Sites: Community," on page 77.

Deploying on Oracle WebLogic Application Server

In this section, you will deploy a non-clustered Community application on Oracle WebLogic Application Server.

Note

- 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 application on WebLogic application server, add the following JVM parameter to `JAVA_OPTIONS` of WebLogic application server: `-DUseSunHttpHandler=true`

1. Before deploying the Community application, ensure that each application server's classpath points to the respective Community configuration directories (created in section "B. Copying Installer-Generated Files," on page 61):

   For reference, this guide uses the following configurations (see Table 14):

   Table 14: Classpath Configuration for Managed Servers (WebLogic, Non-clustered)

Managed Server	Server CLASSPATH
`<cosA_M1>`	`CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosA_M1>:${CLASSPATH}"`
`<cosB_P1>`	`CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosB_P1>:${CLASSPATH}"`

2. The Community installer creates `war` and `ear` files in the `<cos_install_dir>/deploy/management` directory and the `<cos_install_dir>/deploy/production` directory. 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 15.

   Table 15: `war` File Locations for WebLogic (Non-clustered)

Managed Server	Location of war Files	Sample Deployment Location
`<cosA_M1>`	`<cos_install_dir>/deploy/management/cos.war`	`<WL_HOME>/user_projects/domains/<cosMgmt>/applications/cos`
`<cosB_P1>`	`<cos_install_dir>/deploy/production/cos.war`	`<WL_HOME>/user_projects/domains/<cosProd>/applications/cos`

Table 15: `war` File Locations for WebLogic (Non-clustered)

Managed Server	Location of war Files	Sample Deployment Location
	`<cos_install_dir>/deploy/production/cas.war`	`<WL_HOME>/user_projects/domains/<cosProd>/applications/cas`

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 2, "Prerequisites."
4. Copy the following files from the Community application to its application server's JRE: `jaxb-api-2.2.2.jar` and `jaxb-impl-2.2.3-1.jar`. The files must be placed into the `<JDK>/jre/lib/endorsed` folder (if the `endorsed` folder does not exist, create it).
5. Start the production deployment servers before starting the management deployment servers.

   For example, in Linux:

   ```
   cd <WL_HOME>/user_projects/domains/<cosProd>/bin
   ./startManagedServer <cosB_P1> http://<ServerB>:7001

   cd <WL_HOME>/user_projects/domains/<cosMgmt>/bin
   ./startManagedServer <cosA_M1> http://<ServerA>:7001
   ```
6. Continue to "Verifying and Registering Oracle WebCenter Sites: Community," on page 77.

Deploying on IBM WebSphere Application Server

In this section, you will deploy a non-clustered Community 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 application, ensure that each application server's classpath points to the respective Community configuration directories (created in section "B. Copying Installer-Generated Files," on page 61). 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** > **Process Definition** > **Java Virtual Machine** and edit the `CLASSPATH` field.

   For reference, this guide uses the following configurations (see Table 16):

Table 16: Classpath Configuration for WebSphere (Non-clustered)

Managed Server	Server CLASSPATH
<cosA_M1>	/<cos_install_dir>/<cos-standalone-configs>/<cosA_M1>
<cosB_P1>	/<cos_install_dir>/<cos-standalone-configs>/<cosB_P1>

2. The Community installater creates war and ear files in the `<cos_install_dir>/deploy/management` directory and the `<cos_install_dir>/deploy/production` directory. Locations of the `ear` files are shown in Table 17.

Table 17: `ear` File Locations for WebSphere (Non-clustered)

Managed Server	Location of ear Files
<cosA_M1>	<cos_install_dir>/deploy/management/Community.ear
<cosB_P1>	<cos_install_dir>/deploy/production/Community.ear <cos_install_dir>/deploy/production/cas.ear

3. Complete the following steps in the WebSphere Admin Console for each Community application and the visitor CAS application:
   a. Select **Applications** > **Application Types** > **WebSphere enterprise applications** > ***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 application is `/cos`, and for the visitor 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 application to its application server's JRE: `jaxb-api-2.2.2.jar` and `jaxb-impl-2.2.3-1.jar`. The files must be placed into the `<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 2, "Prerequisites."
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 servers before starting the management servers.

   For example, in Linux:

   On <ServerB>:
   ```
   ./startServer.sh cosB_P1 -profileName <AppSrv02>
   ```

   On <ServerA>:
   ```
   ./startServer.sh cosA_M1 -profileName <AppSrv01>
   ```

8. Continue to "Verifying and Registering Oracle WebCenter Sites: Community," on page 77.

Deploying a Clustered Community Application

This section describes the steps for deploying a clustered Community application to supported application servers.

This section covers the following deployment configurations:

- Deploying on Apache Tomcat (Clustered)
- Deploying on Oracle WebLogic Application Server (Clustered)
- Deploying on IBM WebSphere Application Server (Clustered)

Deploying on Apache Tomcat (Clustered)

In this section, you will deploy a clustered Community 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 application, ensure that each application server's classpath points to the respective Community configuration directories (created in section "B. Copying Installer-Generated Files," on page 61):

   For reference, this guide uses the following configurations (see Table 18):

Table 18: Classpath Configuration for Tomcat Servers (Clustered)

Managed Server	Server CLASSPATH
<cosA_M1>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosA_M1>:${CLASSPATH}"
<cosB_P1>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosB_P1>:${CLASSPATH}"

Table 18: Classpath Configuration for Tomcat Servers (Clustered)

Managed Server	Server CLASSPATH
<cosA_M2>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosA_M2>:${CLASSPATH}"
<cosB_P2>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosB_P2>:${CLASSPATH}"

2. The Community installer creates `war` and `ear` files in the `<cos_install_dir>/deploy/management` directory and the `<cos_install_dir>/deploy/production` directory. Deploy the Community 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 19.

Table 19: `war` File Locations for Tomcat (Clustered)

Application Server Instance	Location of war Files	Sample Deployment Location
<cosA_M1>	<cos_install_dir>/deploy/management/cos.war	<cosA_M1>/webapps/cos
<cosB_P1>	<cos_install_dir>/deploy/production/cos.war <cos_install_dir>/deploy/production/cas.war	<cosB_P1>/webapps/cos <cosB_P1>/webapps/cas
<cosA_M2>	<cos_install_dir>/deploy/management/cos.war	<cosA_M2>/webapps/cos
<cosB_P2>	<cos_install_dir>/deploy/production/cos.war <cos_install_dir>/deploy/production/cas.war	<cosB_P2>/webapps/cos <cosB_P2>/webapps/cas

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 2, "Prerequisites."
4. Start the production deployment servers before starting the management deployment servers.

   For example, in Linux:

   On <ServerB>:

```
cd <cosB_P1>/bin
./catalina.sh run
```

```
cd <cosB_P2>/bin
./catalina.sh run
```

On <ServerA>:
```
cd <cosA_M1>/bin
./catalina.sh run

cd <cosA_M2>/bin
./catalina.sh run
```

5. Continue to "Verifying and Registering Oracle WebCenter Sites: Community," on page 77.

Deploying on Oracle WebLogic Application Server (Clustered)

In this section, you will deploy a clustered Community application on Oracle WebLogic Application Server.

Note

- 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 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 application, ensure that each application server's classpath points to the respective Community configuration directories (created in section "B. Copying Installer-Generated Files," on page 61):

   For reference, this guide uses the following configurations (see Table 20):

   Table 20: Classpath Configuration for Managed Servers (Clustered WebLogic)

Managed Server	Server CLASSPATH
<cosA_M1>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosA_M1>:${CLASSPATH}"
<cosB_P1>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosB_P1>:${CLASSPATH}"
<cosA_M2>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosA_M2>:${CLASSPATH}"
<cosB_P2>	CLASSPATH="/<cos_install_dir>/<cos-standalone-configs>/<cosB_P2>:${CLASSPATH}"

2. The Community installer creates `war` and `ear` files in the `<cos_install_dir>/deploy/management` directory and the `<cos_install_dir>/deploy/production` directory. 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 21.

Table 21: `war` File Locations for WebLogic (Clustered)

Target Servers	Location of war Files	Sample Deployment Location

All cluster members:
`<cosA_M1>`
`<cosA_M2>`
	`<cos_install_dir>/deploy/management/cos.war`	`<WL_HOME>/user_projects/domains/<cosMgmt>/applications/cos`

All cluster members:
`<cosB_P1>`
`<cosB_P2>`
	`<cos_install_dir>/deploy/production/cos.war`
	`<cos_install_dir>/deploy/production/cas.war`
		`<WL_HOME>/user_projects/domains/<cosProd>/applications/cos`
		`<WL_HOME>/user_projects/domains/<cosProd>/applications/cas`

3. Before starting the managed servers, ensure that you have WebCenter Sites running and configured as described in Chapter 2, "Prerequisites."
4. Copy the following files from the Community 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 `<JDK>/jre/lib/endorsed` folder (if the `endorsed` folder does not exist, create it).
5. Start the production deployment servers before starting the management deployment servers.

   For example, in Linux:

   On `<ServerB>`:

   ```
   cd <WL_HOME>/user_projects/domains/<cosProd>/bin
   ./startManagedServer <cosB_P1> http://<ServerB>:7001

   cd <WL_HOME>/user_projects/domains/<cosProd>/bin
   ./startManagedServer <cosB_P2> http://<ServerB>:7001
   ```

   On `<ServerA>`:

   ```
   cd <WL_HOME>/user_projects/domains/<cosMgmt>/bin
   ./startManagedServer <cosA_M1> http://<ServerA>:7001

   cd <WL_HOME>/user_projects/domains/<cosMgmt>/bin
   ./startManagedServer <cosA_M2> http://<ServerA>:7001
   ```

6. Continue to "Verifying and Registering Oracle WebCenter Sites: Community," on page 77.

Deploying on IBM WebSphere Application Server (Clustered)

In this section, you will deploy a clustered Community 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 application, ensure that each application server's classpath points to the respective Community configuration directories (created in section "B. Copying Installer-Generated Files," on page 61). 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** > **Process Definition** > **Java Virtual Machine** and edit the `CLASSPATH` field.

      For reference, this guide uses the following configurations (see Table 22):

Table 22: Classpath Configuration for WebSphere (Clustered)

Managed Server	Server CLASSPATH
`<cosA_M1>`	`/<cos_install_dir>/<cos-standalone-configs>/<cosA_M1>`
`<cosB_P1>`	`/<cos_install_dir>/<cos-standalone-configs>/<cosB_P1>`
`<cosA_M2>`	`/<cos_install_dir>/<cos-standalone-configs>/<cosA_M2>`
`<cosB_P2>`	`/<cos_install_dir>/<cos-standalone-configs>/<cosB_P2>`

2. The Community installater creates war and ear files in the `<cos_install_dir>/deploy/management` directory and the `<cos_install_dir>/deploy/production` directory. Locations of the `ear` files are shown in Table 23.

Table 23: ear File Locations for WebSphere (Clustered)

Target Server	Location of ear Files
`<wscos_M> cluster: <cosA_M1>,<cosA_M2>`	`<cos_install_dir>/deploy/management/Community.ear`
`<wscos_P> cluster: <cosB_P1>,<cosB_P2>`	`<cos_install_dir>/deploy/production/Community.ear` `<cos_install_dir>/deploy/production/cas.ear`

3. Complete the following steps in the WebSphere Admin Console for each Community application and the visitor CAS application:
   a. Select **Applications** > **Application Types** > **WebSphere enterprise applications** > ***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 application is `/cos`, and for the visitor 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 application to its application server's JRE: `jaxb-api-2.2.2.jar` and `jaxb-impl-2.2.3-1.jar`. The files must be placed into the `<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 2, "Prerequisites."
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 servers before starting the management servers.

   For example, in Linux:

   On `<ServerB>`:
   ```
   ./startServer.sh cosB_P1 -profileName <AppSrv02>
   ./startServer.sh cosB_P2 -profileName <AppSrv02>
   ```

   On `<ServerA>`:
   ```
   ./startServer.sh cosA_M1 -profileName <AppSrv01>
   ./startServer.sh cosA_M2 -profileName <AppSrv01>
   ```
8. Continue to "Verifying and Registering Oracle WebCenter Sites: Community," on page 77.

Chapter 6

Verifying and Registering Oracle WebCenter Sites: Community

Having completed the WebCenter Sites: Community installation and deployed the web applications, you will verify the installation, register the management Community application so that it can be recognized by WebCenter Sites, and finally authorize users to work with the Community application.

This chapter contains the following sections:

- Verifying Community URLs
- Registering the Community Application
- Authorizing Users

Verifying Community URLs

In this section, you will verify that your newly installed WebCenter Sites: Community application is operating correctly.

Note

The URLs in this section are used only to verify that the Community application is correctly installed. The URLs cannot be used to invoke the Community application interface for management operations (an error will be displayed).

For security, after completing the installation 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 find it before starting a number of servers.

   a. Start the WebCenter Sites application servers (production and management).
   b. Start a single Community (production) application server instance.
   c. Start the remaining Community (production) application server instances.
   d. Start a single Community (management) application server instance.
   e. Start the remaining Community (management) application server instances.
   f. Start load balancers (when required).
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 visitor CAS:
   a. Verify that the visitor CAS application login form is displayed. **You cannot yet log in:**

      `http://{production_cos_ip}:{production_cos_port}/cas/login`

      For example:

      `http://10.120.19.122:8080/cas/login`
   b. Verify the status of the Community application for each node of both production and management. It is best to start with production, then move to management:

      `http://<cos node>:<cos node port>/cos/status`

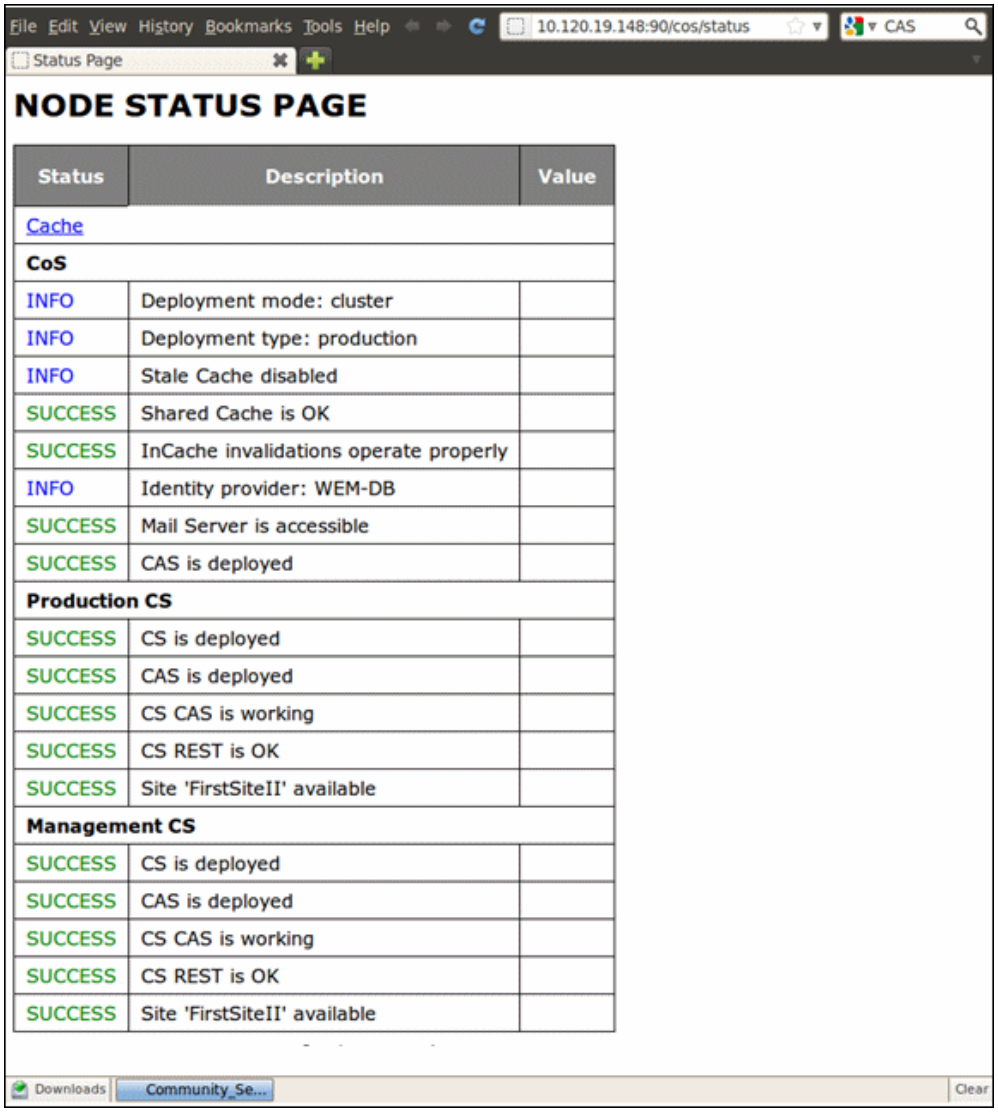
      For example:

      Node1: `http://10.120.19.122:8080/cos/status`
      Node2: `http://10.120.19.123:8080/cos/status`

A screen similar to the following should be displayed with all the options showing either `SUCCESS` or `Info`. If any option displays `Failed` or the page does not load within 90 seconds, then you cannot continue.

Note

If any failures are seen, review the Community application's log file and the application server's logs for details on what has failed.



4. Log in to the Community application as a general administrator on each Management node (the default user is `fwadmin`. If you entered the credentials of a different administrator during the installation process, use those credentials):

```
http://<Cos Node Name>:<Cos Node Port>/cos/admin/
 start.jsp?siteid=<Site used to register the Community
 application>
```

For example:

`http://10.120.19.148:90/cos/admin/start.jsp?siteid=FirstSiteII`

Once logged in, you should see a message stating that you did not access the Community interface through the WEM Framework.

Note

If the Community 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 application's log file and the application server's logs for details on what has failed.

Registering the Community Application

In this section, you will register the management Community application in order for it to be recognized by WebCenter Sites. Registration displays the Community application icon on the WebCenter Sites login page and applications bar; lists the application on the **Apps** page of the WEM Admin interface, and enables the WEM Framework to render the application's interface.

Follow these steps to register the Community application:

1. On the management system, log in to WebCenter Sites as a general administrator (the default credentials are `fwadmin/xceladmin`).
   a. Use the following URL:

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

      where `<server>` is the host name or IP address of the server running WebCenter Sites, and `<context>` is the name of the WebCenter Sites web application that was deployed on `<server>`. Depending on how the system was set up, you may also need to include the port number.
   b. Access **AdminSite**.
2. Open the WebCenter Sites Admin interface and register the Community application:
   a. On the **Admin** tab, select **Site** > **AdminSite**.
   b. Create an asset of `type FW_View` with the following field values:
      - **Name**: `CoSView` (or a name of your own choice)
      - **Parent Node**: `frame1`
      - **View Type**: `Iframe`
      - **Source Url**: `http://{management_cos_ip}:{management_cos_port}/cos/admin/WemContext.action`
   c. Create an asset of type `FW_Application` with the following field values:
      - **Name**: `CoS` (or a name of your own choice)
      - **Icon URL:** `wemresources/images/icons/apps/CoS.png`

- **Active Icon URL:** `wemresources/images/icons/apps/CoSActive.png`
- **Hover Icon URL:** `wemresources/images/icons/apps/CoS.png`
- **Click Icon URL:** `wemresources/images/icons/apps/CoS.png`
- **Layout Type:** `Layout Renderer`
- **Layout URL:** `wemresources/layout/admin.html`

d. Add the newly created asset of type `FW_View` as an associated view by selecting it from the **History** tab and clicking **Add Selected Items**. Click **Save Changes**.

3. Verify that the Community application is listed in the WEM Admin interface.
   a. Open the WEM Admin interface (on AdminSite).
   b. Click **Apps** in the menu bar and verify that the Community application is listed on the "Apps" page. Keep this page open.
4. Continue to "Authorizing Users" to authorize access to the Community application.

Authorizing Users

In this section, you will enable users to work with the management Community application. Detailed information about user authorization is available in the *Oracle WebCenter Sites Administrator's Guide for the WEM Framework*. For a quick start, follow the steps below.

Note

Throughout this section, you will be working in the WEM Admin interface. You will assign the application and the users to the same site (`{cs_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.

To authorize users, complete the following steps:

1. Assign the management Community application to `{cs_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 site to which you are assigning the management Community application.
   d. Select at least the following roles for the Community application: **SiteAdmin**, **Moderator**, and **Designer**
2. Authorize users to access the management Community application:
   a. In the menu bar, select **Sites**.
   b. Point to the site to which the Community 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:

- 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**.

1) In the "Assign Roles to User" screen, select at least one of the following roles: **SiteAdmin**, **Designer**, **Moderator** (for permissions associated with each role, see the *Oracle WebCenter Sites User's Guide for the Community Application*).

2) Click **Save and Close**.

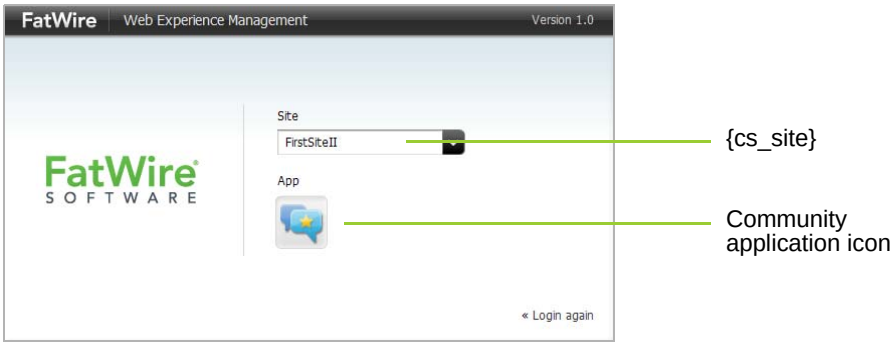
Note

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 *Oracle WebCenter Sites Administrator's Guide for the Web Experience Management Framework.*

3. Verify each user's access to the management Community application on `{cs_site}` as follows:

a. Log in to WebCenter Sites as a Community user that you authorized in the steps above.

b. Select `{cs_site}` and click the Community application icon:



The "All Comments" screen opens:

The user authorization process on `{cs_site}` is complete.

4. If you wish to assign the management Community 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.

You are now ready to use the Community application. For more information, see the *Oracle WebCenter Sites User's Guide for the Community Application*.