

Oracle® Fusion Middleware

Installing and Configuring Oracle HTTP Server



12c (12.2.1.3.0)

E95528-04

August 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server, 12c (12.2.1.3.0)

E95528-04

Copyright © 2014, 2021, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	vii

1 About the Oracle HTTP Server Installation

About Oracle HTTP Server	1-1
Using the Standard Installation Topology as a Starting Point	1-1
About the Standard Installation Topology for Oracle HTTP Server in a WebLogic Server Domain	1-2
Understanding Elements in the Standard Installation Topology Illustration	1-3
About Full-JRF and Restricted-JRF Modes	1-4
About the Standard Installation Topology for Oracle HTTP Server in a Standalone Domain	1-5

2 Preparing to Install and Configure Oracle HTTP Server

Selecting an Oracle HTTP Server Installation Type	2-1
Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain	2-2
Roadmap for Installing and Configuring Oracle HTTP Server in a Standalone Domain	2-4
Roadmap for Verifying Your System Environment	2-5
Verifying Certification, System, and Interoperability Requirements	2-6
Selecting an Installation User	2-6
About User Permissions	2-7
About Non-Default User Permissions on UNIX Operating Systems	2-9
Verifying that the Installation User has Administrator Privileges on Windows Operating Systems	2-9
Understanding Directories for Installation and Configuration	2-10
Understanding the Recommended Directory Structure	2-10
About the Oracle Home Directory	2-11
About the Domain Home Directory	2-12
About the Application Home Directory	2-13

Installing Multiple Products in the Same Domain	2-13
Preparing for Shared Storage	2-14
About JDK Requirements for an Oracle Fusion Middleware Installation	2-14
About Database Requirements for an Oracle Fusion Middleware Installation	2-14
Obtaining the Product Distribution	2-15

3 Installing the Oracle HTTP Server Software

Starting the Installation Program	3-1
Navigating the Installation Screens	3-1
Verifying the Installation	3-2
Reviewing the Installation Log Files	3-3
Checking the Directory Structure	3-3
Viewing the Contents of the Oracle Home	3-3

4 Configuring Oracle HTTP Server in a WebLogic Server Domain

Stopping the Administration Server	4-1
Starting the Configuration Wizard	4-2
Navigating the Configuration Wizard Screens to Extend the Domain	4-2
Selecting the Domain Type and Domain Home Location	4-3
Selecting the Configuration Template for Oracle HTTP Server	4-3
Configuring High Availability Options	4-4
Specifying the Database Configuration Type	4-5
Specifying JDBC Component Schema Information	4-5
Testing the JDBC Connections	4-6
Selecting Advanced Configuration	4-6
Specifying the Oracle HTTP Server Details	4-6
Assigning System Components to Machines	4-6
Reviewing Your Configuration Specifications and Configuring the Domain	4-7
Writing Down Your Domain Home and Administration Server URL	4-7
Starting the Servers	4-7
Starting Node Manager	4-7
Starting the Administration Server	4-8
Starting the Oracle HTTP Server Instance	4-9

5 Configuring Oracle HTTP Server in a Standalone Domain

Starting the Configuration Wizard	5-1
Navigating the Configuration Wizard Screens to Configure the Domain	5-2
Selecting the Domain Type and Domain Home Location	5-2
Selecting the Configuration Template for Oracle HTTP Server	5-3

Specifying the Domain Mode and JDK	5-3
Adding System Components	5-3
Specifying the Oracle HTTP Server Details	5-4
Configuring Node Manager	5-4
Reviewing Your Configuration Specifications and Configuring the Domain	5-4
Starting the Instances	5-4
Starting Node Manager	5-5
Starting the Oracle HTTP Server Instance	5-5
Missing Libraries Might Cause HTTPD to Exit Without Notice	5-6

6 Next Steps After Configuring the Domain

Next Steps for Oracle HTTP Server in a WebLogic Server Domain	6-1
Next Steps for Oracle HTTP Server in a Standalone Domain	6-3
Preparing Your Environment for High Availability	6-4

7 Uninstalling or Reinstalling Oracle HTTP Server

Stopping Oracle Fusion Middleware	7-2
Uninstalling the Software	7-2
Starting the Uninstall Wizard	7-2
Navigating the Uninstall Wizard Screens	7-2
Verifying the Uninstallation of Oracle HTTP Server in a Standalone Domain	7-3
Removing the Oracle Home Directory Manually	7-3
Removing the Program Shortcuts on Windows Operating Systems	7-4
Removing the Domain and Application Data	7-4
Reinstalling the Software	7-5

A Oracle HTTP Server Directory Structure

Oracle HTTP Server Directory Structure in a WebLogic Server Domain	A-1
Oracle HTTP Server Directory Structure in a Standalone Domain	A-3

B Configuring Oracle HTTP Server WebGate for Oracle Access Manager

About Webgate	B-1
General Prerequisites for Configuring Oracle HTTP Server WebGate	B-1
Configuring Oracle HTTP Server WebGate	B-2
Registering the Oracle HTTP Server WebGate with Oracle Access Manager	B-3
Locating and Preparing the RREG Tool	B-3
Updating the Standard Properties in the OAM11gRequest.xml File	B-4
Running the RREG Tool	B-6

About RREG In-Band and Out-of-Band Mode	B-6
Running the RREG Tool in In-Band Mode	B-7
Running the RREG Tool in Out-Of-Band Mode	B-8
Files and Artifacts Generated by RREG	B-8
Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location	B-9
Restarting the Oracle HTTP Server Instance	B-13

C Updating the JDK After Installing and Configuring an Oracle Fusion Middleware Product

About Updating the JDK Location After Installing an Oracle Fusion Middleware Product	C-1
Updating the JDK Location in an Existing Oracle Home	C-2
Updating the JDK Location in an Existing Domain Home	C-3

Preface

This document describes how to install and configure an Oracle HTTP Server domain.

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)
Learn about the conventions used in this document.

Audience

This document is intended for system administrators or application developers who are installing the Oracle HTTP Server. It is assumed that readers are familiar with Web technologies and have a general understanding of Windows and UNIX platforms.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

Refer to the Oracle Fusion Middleware Library for additional information.

- For Oracle HTTP Server information, see Oracle HTTP Server Documentation.
- For installation information, see Fusion Middleware Installation Documentation.
- For upgrade information, see Fusion Middleware Upgrade Documentation.
- For administration-related information, see Fusion Middleware Administration Documentation.
- For release-related information, see Fusion Middleware Release Notes.

Conventions

Learn about the conventions used in this document.

This document uses the following text conventions:

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

1

About the Oracle HTTP Server Installation

The standard installation for Oracle HTTP Server described in this guide creates the standard topology, which represents a sample starting topology for this product.

- [About Oracle HTTP Server](#)
Oracle HTTP Server provides web services for Oracle Fusion Middleware.
- [Using the Standard Installation Topology as a Starting Point](#)
The standard installation topology is a flexible topology that you can use as a starting point in production environments.

About Oracle HTTP Server

Oracle HTTP Server provides web services for Oracle Fusion Middleware.

In summary, Oracle HTTP Server:

- Delivers the HTTP Listener for the Oracle WebLogic Server through the built-in Oracle WebLogic Server Proxy Plug-In
- Provides the Web Server component for Oracle Fusion Middleware
- Serves static web content, such as HTML, JavaScript, images, and dynamic web content built with CGI/FastCGI-based applications.

See Introduction to Oracle HTTP Server in *Administering Oracle HTTP Server*.

Oracle HTTP Server can be installed in either an existing Oracle home or in a standalone domain:

- In an existing Oracle home, Oracle HTTP Server is collocated with a WebLogic Server domain, so the Oracle HTTP Server instance can be managed like any other element of the WebLogic Server domain. Specifically, you can manage a Oracle HTTP Server instance from the Oracle Enterprise Manager Fusion Middleware Control or the WLST command line interface. See [About the Standard Installation Topology for Oracle HTTP Server in a WebLogic Server Domain](#).

You can configure a collocated Oracle HTTP Server to run in either Full-JRF or Restricted-JRF mode. See [About Full-JRF and Restricted-JRF Modes](#).
- In a standalone domain, the standalone Oracle HTTP Server topology can be installed and configured without Oracle Fusion Middleware Infrastructure. See [About the Standard Installation Topology for Oracle HTTP Server in a Standalone Domain](#).

Using the Standard Installation Topology as a Starting Point

The standard installation topology is a flexible topology that you can use as a starting point in production environments.

The information in this guide helps you to create a standard installation topology for Oracle HTTP Server. If required, you can later extend the standard installation topology to create a

secure and highly available production environment, see [Next Steps After Configuring the Domain](#).

The standard installation topology represents a sample topology for this product. It is not the only topology that this product supports. See [About the Standard Installation Topology](#) in *Planning an Installation of Oracle Fusion Middleware*.

- [About the Standard Installation Topology for Oracle HTTP Server in a WebLogic Server Domain](#)
Oracle HTTP Server can be installed in an existing Oracle home, collocated with a WebLogic Server domain
- [About the Standard Installation Topology for Oracle HTTP Server in a Standalone Domain](#)
Oracle HTTP Server can be installed in a standalone domain in its own Oracle home.

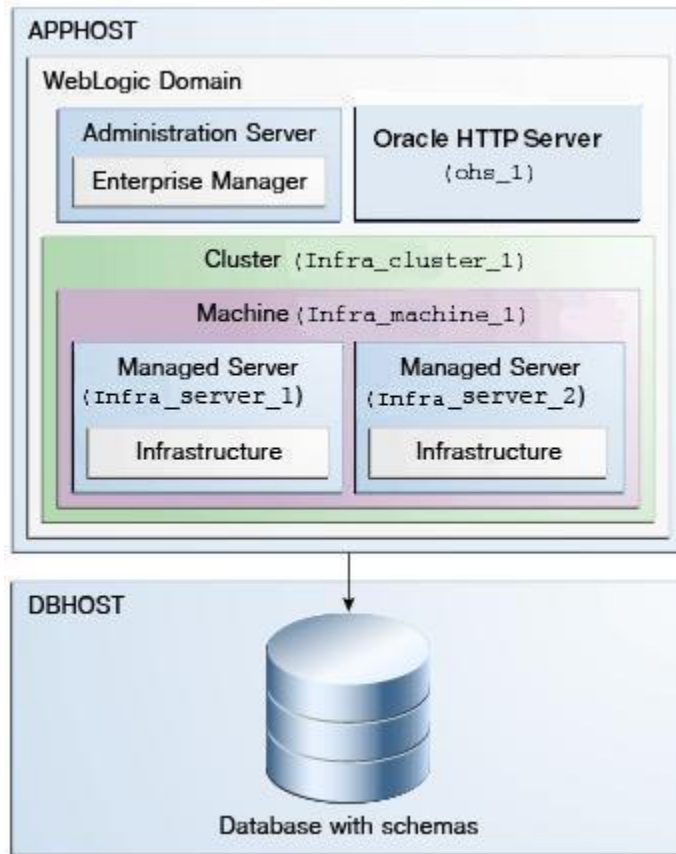
About the Standard Installation Topology for Oracle HTTP Server in a WebLogic Server Domain

Oracle HTTP Server can be installed in an existing Oracle home, collocated with a WebLogic Server domain

In this type of installation, the Oracle HTTP Server instance can be managed like any other elements of the WebLogic Server domain. Specifically, you can manage your Oracle HTTP Server instance from the Oracle Enterprise Manager Fusion Middleware Control, the WLST Command line interface, and the WebLogic Server Node Manager.

[Figure 1-1](#) shows the standard Oracle Fusion Middleware Infrastructure topology with an Oracle HTTP Server instance collocated in the same domain home. Note that you can also use this topology to scale out to a high availability environment spanning multiple machines.

Figure 1-1 Topology of an Oracle HTTP Server Installation in a WebLogic Server Domain



For configuration instructions, see [Configuring Oracle HTTP Server in a WebLogic Server Domain](#)

- [Understanding Elements in the Standard Installation Topology Illustration](#)
The standard installation topology typically includes common elements.
- [About Full-JRF and Restricted-JRF Modes](#)
You can configure a collocated Oracle HTTP Server to run in either Full-JRF or Restricted-JRF mode.

Understanding Elements in the Standard Installation Topology Illustration

The standard installation topology typically includes common elements.

[Table 1-1](#) describes all elements of the topology illustration.

Table 1-1 Description of Elements in the Standard Installation Topology for Oracle HTTP Server

Element	Description and Links to Additional Documentation
APPHOST	A standard term used in Oracle documentation to refer to the machine that hosts the application tier.

Table 1-1 (Cont.) Description of Elements in the Standard Installation Topology for Oracle HTTP Server

Element	Description and Links to Additional Documentation
DBHOST	A standard term used in Oracle documentation to refer to the machine that hosts the database.
WebLogic Domain	A logically related group of Java components (in this case, the Administration Server, Managed Servers, and other related software components). For more information, see <i>What Is an Oracle WebLogic Server Domain?</i> in <i>Understanding Oracle Fusion Middleware</i> .
Administration Server	The central control entity of a domain which maintains the configuration objects for that domain and distributes configuration changes to the Managed Servers. For more information, see <i>What Is the Administration Server?</i> in <i>Understanding Oracle Fusion Middleware</i> .
Enterprise Manager	The Oracle Enterprise Manager Fusion Middleware Control is a primary tool used to manage a domain. For more information, see <i>Oracle Enterprise Manager Fusion Middleware Control</i> in <i>Understanding Oracle Fusion Middleware</i> .
Oracle HTTP Server	The Oracle HTTP Server binaries and an Oracle HTTP Server instance.
Cluster	A collection of multiple WebLogic Server instances running simultaneously and working together. For more information, see <i>Overview of Managed Servers and Managed Server Clusters</i> in <i>Understanding Oracle Fusion Middleware</i> .
Machine	A logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are also the logical glue between the Managed Servers and the Node Manager. You must associate Managed Servers with a machine to use Node Manager to start or stop them.
Managed Server	A host for your applications, application components, web services, and their associated resources. For more information, see <i>Overview of Managed Servers and Managed Server Clusters</i> in <i>Understanding Oracle Fusion Middleware</i> .
Infrastructure	A collection of services that include: <ul style="list-style-type: none"> • Metadata repository (MDS) containing metadata for Oracle Fusion Middleware components, such as the Oracle Application Developer Framework. For more information, see <i>What Is the Metadata Repository?</i> in <i>Understanding Oracle Fusion Middleware</i>. • Oracle Application Developer Framework (Oracle ADF) • Oracle Web Services Manager (OWSM)

About Full-JRF and Restricted-JRF Modes

You can configure a collocated Oracle HTTP Server to run in either Full-JRF or Restricted-JRF mode.

Table 1-2 provides a comparison of these two modes.

Table 1-2 Comparison of Full-JRF and Restricted-JRF Modes

Operational Mode	Description	Recommended Use Case
Full Domain (or Full-JRF)	Full-JRF mode is dependent on a database connection.	This operational mode is best if your domain is accessing upper stack features, such as Oracle SOA Suite or Oracle Web Services Manager.
Restricted-JRF	Restricted-JRF mode works without a database connection. Users can still manage all products via the Oracle Enterprise Manager Fusion Middleware Control, but they cannot use cross component wiring.	This operational mode is best if your domain only accesses Oracle Traffic Director and Oracle HTTP Server.

Note that migrating from Restricted-JRF to Full-JRF is not currently supported.

For a full description of what these two operational modes offer, see Domain Types in *Administering Oracle HTTP Server*.

About the Standard Installation Topology for Oracle HTTP Server in a Standalone Domain

Oracle HTTP Server can be installed in a standalone domain in its own Oracle home.

You can create a standalone Oracle HTTP Server domain dedicated to managing system components only. A standalone configuration resides in its own Oracle home, which you create during the installation process, which means that the standalone Oracle HTTP Server topology can be installed and configured without the Oracle Fusion Middleware Infrastructure.

A standalone domain cannot be managed with Fusion Middleware Control. You can manage your standalone Oracle HTTP Server domain with the WLST command-line and other features available in a standalone domain.

[#unique_19/unique_19_Connect_42_BABEFCGE](#) shows the standard Oracle HTTP Server instance in the Web tier.

For configuration instructions, see [Configuring Oracle HTTP Server in a Standalone Domain](#)

[Table 1-3](#) describes the elements of this topology.

Table 1-3 Description of the Elements in the Oracle HTTP Server Standalone Installation Topology

Element	Description and Links to Additional Documentation
WEBHOST	This a separate host, which the sends requests to a WebLogic Server domain.
Standalone Domain	For more information, see Standalone Domain in <i>Administering Oracle HTTP Server</i> .
Oracle HTTP Server	The Oracle HTTP Server instance, which is created and managed by the management tools available in the standalone domain.

 **Note:**

For more information on standalone domains in general, see What Is a Standalone Domain? in *Understanding Oracle Fusion Middleware*.

2

Preparing to Install and Configure Oracle HTTP Server

To prepare for your Oracle HTTP Server installation, verify that your system meets the basic requirements, then obtain the correct installation software.

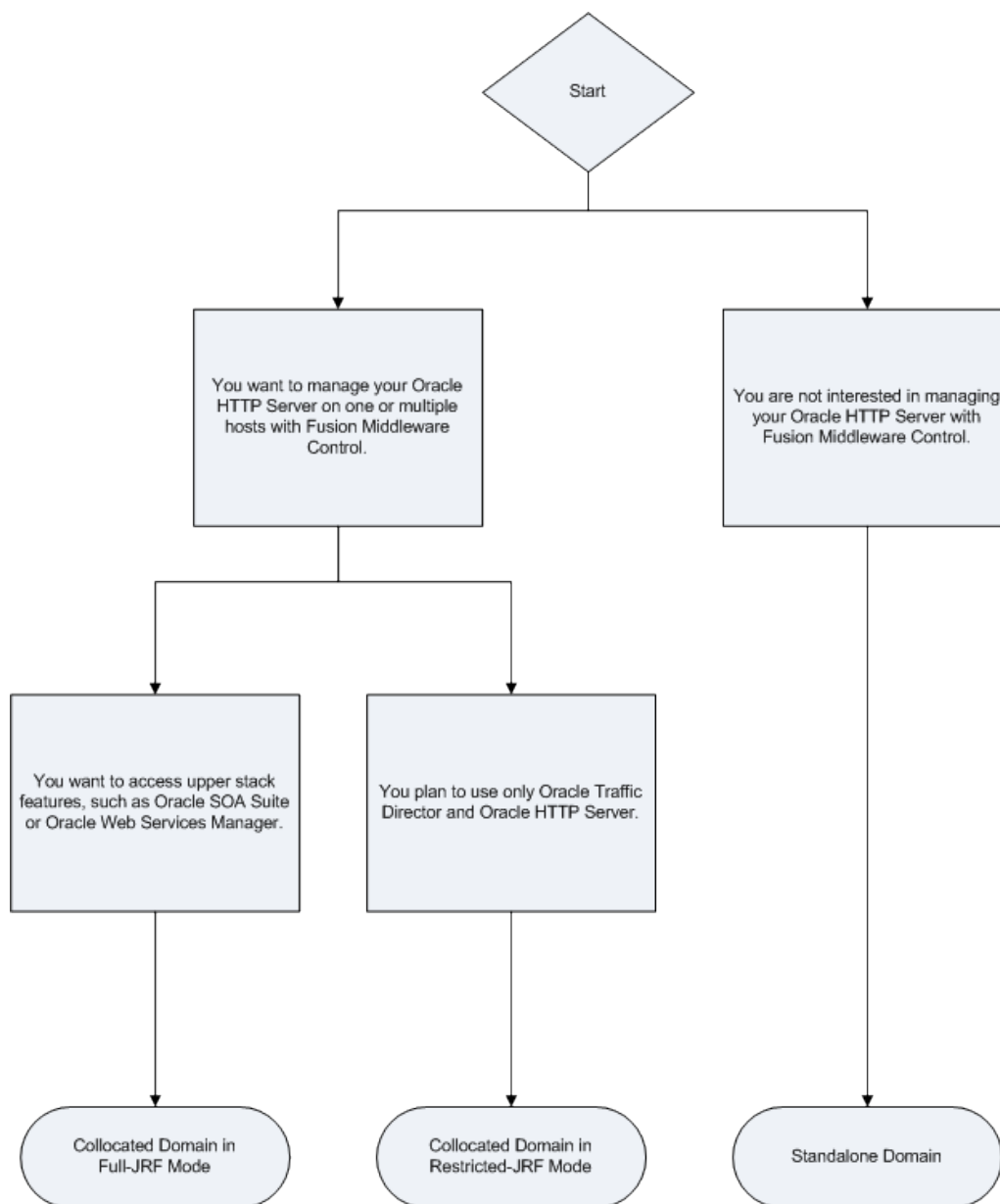
- [Selecting an Oracle HTTP Server Installation Type](#)
You must decide whether to install Oracle HTTP Server collocated with an existing WebLogic Server domain or in a standalone domain.
- [Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain](#)
This roadmap provides steps required to install and configure Oracle HTTP Server in a WebLogic Server domain.
- [Roadmap for Installing and Configuring Oracle HTTP Server in a Standalone Domain](#)
This roadmap provides the steps required to install and configure Oracle HTTP Server in a standalone domain.
- [Roadmap for Verifying Your System Environment](#)
Before you begin the installation and configuration process, you must verify your system environment.
- [Obtaining the Product Distribution](#)
The Oracle HTTP Server distribution is available as a standalone executable installation program that includes a Java Runtime Environment.

Selecting an Oracle HTTP Server Installation Type

You must decide whether to install Oracle HTTP Server collocated with an existing WebLogic Server domain or in a standalone domain.

Use the following flowchart to decide which of the two primary Oracle HTTP Server domain options to configure.

Figure 2-1 Oracle HTTP Server Domain Options



Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain

This roadmap provides steps required to install and configure Oracle HTTP Server in a WebLogic Server domain.

Table 2-1 Roadmap for Oracle HTTP Server Installation Configured in a WebLogic Server Domain

Task	Description	For More Information
Install and configure the Oracle Fusion Middleware Infrastructure standard installation topology.	Before beginning the Oracle HTTP Server installation, you must install and configure Oracle Fusion Middleware Infrastructure.	See <i>Installing the Infrastructure Software in Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure</i> .
Make sure all processes associated with the Oracle Fusion Middleware Infrastructure domain are shut down.	If your Node Manager, Administration Server, Managed Servers, or any other servers are up and running, then you must shut them down before you can begin installing and configuring Oracle HTTP Server in a WebLogic Server domain.	Stopping an Oracle Fusion Middleware Environment in <i>Administering Oracle Fusion Middleware</i> . Methods for Installing Multiple Products in the Same Domain in <i>Planning an Installation of Oracle Fusion Middleware</i> .
Verify your system environment.	Before beginning the installation, verify that the minimum system and network requirements are met.	See Roadmap for Verifying Your System Environment
Check for any mandatory patches that will be required before or after the installation.	Review the Oracle Fusion Middleware Infrastructure release notes to see if there are any mandatory patches required for the software products you are installing.	See <i>Install and Configure</i> in <i>Release Notes for Oracle Fusion Middleware Infrastructure</i> .
Obtain the appropriate distribution.	To create the topology described in this guide, obtain the Oracle HTTP Server distribution.	See Obtaining the Product Distribution .
Determine your installation directories.	Verify that the installer can access or create the required installer directories. Also, verify that the directories exist on systems that meet the minimum requirements.	See Oracle HTTP Server Directory Structure .
Install Oracle HTTP Server.	Run the installation program to install the software. On the Installation Type screen, select Collocated HTTP Server (Managed through WebLogic server) . The installer then transfers the software to your system and updates the Oracle Fusion Middleware Infrastructure Oracle home to include the Oracle HTTP Server binaries.	See Installing the Oracle HTTP Server Software .

Table 2-1 (Cont.) Roadmap for Oracle HTTP Server Installation Configured in a WebLogic Server Domain

Task	Description	For More Information
Update the existing Infrastructure domain.	<p>Use the configuration wizard to extend the Infrastructure domain to include an Oracle HTTP Server instance.</p> <p>During this step, choose whether you want the Full-JRF or Restricted-JRF operation mode. See About Full-JRF and Restricted-JRF Modes for more information about these modes.</p> <p>Note: If you have not configured Oracle Fusion Middleware Infrastructure yet, then you can configure it at the same time as Oracle HTTP Server.</p>	See Configuring Oracle HTTP Server in a WebLogic Server Domain .
Review post-configuration administration and configuration tasks.	Discover additional tools and resources to administer your domain and configure your domain to be highly available.	See Next Steps After Configuring the Domain .

Roadmap for Installing and Configuring Oracle HTTP Server in a Standalone Domain

This roadmap provides the steps required to install and configure Oracle HTTP Server in a standalone domain.

Table 2-2 Roadmap for Oracle HTTP Server Installation Configured in a Standalone Domain

Task	Description	For More Information
Verify your system environment.	Before beginning the installation, verify that the minimum system and network requirements are met.	See Roadmap for Verifying Your System Environment .
Obtain the appropriate distribution.	To create the topology described in this guide, obtain the Oracle HTTP Server distribution.	See Obtaining the Product Distribution .
Determine your installation directories.	Verify that the <i>product</i> and <i>config</i> directories are created, are accessible by the installer, and exist on systems that meet the minimum requirements.	See Oracle HTTP Server Directory Structure .
Install Oracle HTTP Server.	<p>Run the installation program to install the software. On the Installation Type screen, select Standalone HTTP Server (Managed independently of WebLogic server).</p> <p>The installer then transfers the software to your system and creates the Oracle home directory.</p>	See Installing the Oracle HTTP Server Software .

Table 2-2 (Cont.) Roadmap for Oracle HTTP Server Installation Configured in a Standalone Domain

Task	Description	For More Information
Create an Oracle HTTP Server Standalone Domain.	Use the configuration wizard to create and configure an Oracle HTTP Server in a standalone domain.	See Configuring Oracle HTTP Server in a Standalone Domain .
Review post-configuration administration and configuration tasks.	Discover additional tools and resources to administer your domain and configure your domain to be highly available.	See Next Steps After Configuring the Domain .

Roadmap for Verifying Your System Environment

Before you begin the installation and configuration process, you must verify your system environment.

[Table 2-3](#) identifies important tasks and checks to perform to ensure that your environment is prepared to install and configure Oracle HTTP Server.

Table 2-3 Roadmap for Verifying Your System Environment

Task	Description	Documentation
Verify certification and system requirements.	Verify that your operating system is certified and configured for installation and configuration.	See Verifying Certification, System, and Interoperability Requirements .
Identify a proper installation user.	Verify that the installation user has the required permissions to install and configure the software.	See Selecting an Installation User .
Select the installation and configuration directories on your system.	Verify that you can create the necessary directories to install and configure the software, according to the recommended directory structure.	See Understanding Directories for Installation and Configuration .
Install a certified JDK.	The installation program for the distribution requires a certified JDK present on your system.	See About JDK Requirements for an Oracle Fusion Middleware Installation .
Install and configure a database for mid-tier schemas.	To configure your WebLogic domain, you must have access to a certified database that is configured for the schemas required by Oracle HTTP Server.	See About Database Requirements for an Oracle Fusion Middleware Installation .

- [Verifying Certification, System, and Interoperability Requirements](#)
Oracle recommends that you use the certification matrix and system requirements documents with each other to verify that your environment meets the requirements for installation.
- [Selecting an Installation User](#)
The user who installs and configures your system must have the required permissions and privileges.

- [Understanding Directories for Installation and Configuration](#)
During the installation and domain configuration process, you must plan on providing the locations for these directories: Oracle home, Domain home, and the Application home.
- [About JDK Requirements for an Oracle Fusion Middleware Installation](#)
Most Fusion Middleware products are in .jar file format. These distributions do not include a JDK. To run a .jar distribution installer, you must have a certified JDK installed on your system.
- [About Database Requirements for an Oracle Fusion Middleware Installation](#)
Many Oracle Fusion Middleware products require database schemas prior to configuration. If you do not already have a database where you can install these schemas, you must install and configure a certified database.

Verifying Certification, System, and Interoperability Requirements

Oracle recommends that you use the certification matrix and system requirements documents with each other to verify that your environment meets the requirements for installation.

1. Verifying that your environment meets certification requirements:

Make sure that you install your product on a supported hardware and software configuration. See the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

Oracle has tested and verified the performance of your product on all certified systems and environments. Whenever new certifications are released, they are added to the certification document right away. New certifications can be released at any time. Therefore, the certification documents are kept outside the documentation libraries and are available on Oracle Technology Network.

2. Using the system requirements document to verify certification:

Oracle recommends that you use the *Oracle Fusion Middleware System Requirements and Specifications* document to verify that the certification requirements are met. For example, if the certification document indicates that your product is certified for installation on 64-Bit Oracle Linux 6.5, use this document to verify that your system meets the required minimum specifications. These include disk space, available memory, specific platform packages and patches, and other operating system-specific requirements. System requirements can change in the future. Therefore, the system requirement documents are kept outside of the documentation libraries and are available on Oracle Technology Network.

3. Verifying interoperability among multiple products:

To learn how to install and run multiple Fusion Middleware products from the same release or mixed releases with each other, see Oracle Fusion Middleware 12c Interoperability and Compatibility in *Understanding Interoperability and Compatibility*.

Selecting an Installation User

The user who installs and configures your system must have the required permissions and privileges.

- [About User Permissions](#)
The user who installs a Fusion Middleware product owns the files and has certain permissions on the files.
- [About Non-Default User Permissions on UNIX Operating Systems](#)
Changing the default permission setting reduces the security of the installation and your system. Oracle does not recommend that change the default permission settings.
- [Verifying that the Installation User has Administrator Privileges on Windows Operating Systems](#)
To update the Windows Registry, you must have administrator privileges.

About User Permissions

The user who installs a Fusion Middleware product owns the files and has certain permissions on the files.

- Read and write permissions on all non-executable files (for example, `.jar`, `.properties`, or `.xml`). All other users in the same group as the file owner have read permissions only.
- Read, write, and execute permissions on all executable files (for example, `.exe`, `.sh`, or `.cmd`). All other users in the same group as the file owner have read and execute permissions only.

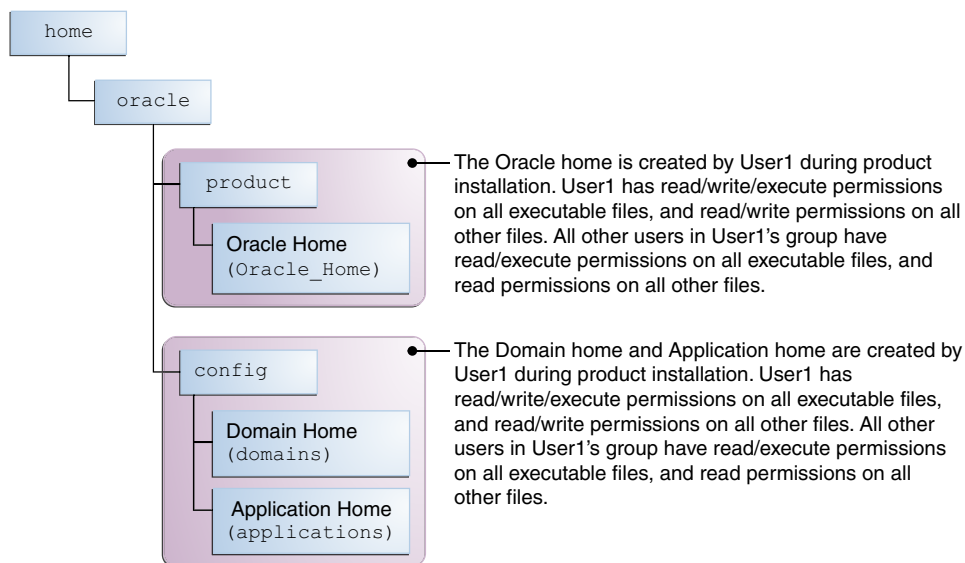
This means that someone other than the person who installs the software can use the installed binaries in the Oracle home directory to configure a domain or set of Fusion Middleware products.

During configuration, the files generated by the configuration process are owned by the user who ran the Configuration Wizard. This user has the same permissions as described above for the installation user. However, security-sensitive files are not created with group permissions. Only the user that created the domain has read and write permissions and can administer the domain.

Consider the following examples:

- **Example 1: A Single User Installs the Software and Configures the Domain**
This example explains the file permissions where the same user installs the software and configures the domain.
To ensure proper permissions and privileges for all files, Oracle recommends that the same owner perform both tasks: install the Oracle Fusion Middleware product and configure the WebLogic Server domain by using the Configuration Wizard.

Figure 2-2 Directory Structure when a Single User Installs the Software and Configures the Domain

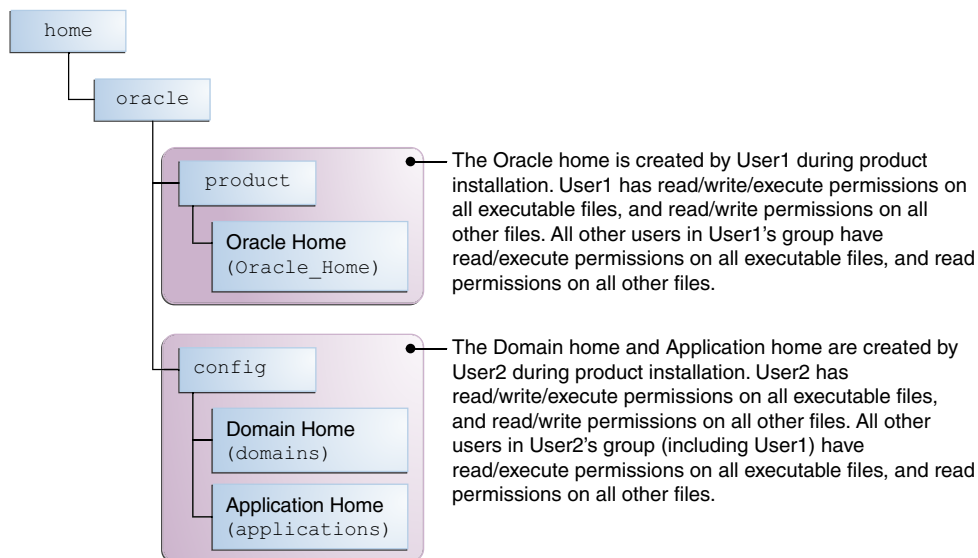


If the user who creates the domain is different than the user who installed the software, then both users must have the same privileges, as shown in the next example.

- **Example 2: The Oracle Home Directory and Domain are Created by Different Users**

This example explains the file permissions where one user creates the Oracle home and another user configures the domain.

Figure 2-3 Directory Structure when Different Users Install the Software and Configure the Domain





Note:

Certain domain files do not have group permissions. For example, `cwallet.sso`.

Consider the following points before you run the installer:

- On UNIX operating systems, Oracle recommends that you set `umask` to `027` on your system before you install the software. This ensures that the file permissions are set properly during installation. Use the following command:

```
umask 027
```

You must enter this command in the same terminal window from which you plan to run the product installer.
- On UNIX operating systems, do not run the installation program as a `root` user. If you run the installer as a root user, the startup validation may fail and you cannot continue the installation.
- When you manage a product installation (for example, applying patches or starting managed Servers), use the same user ID that you used to install the product.
- On Windows operating systems, you must have administrative privileges to install the product. See [Verifying the Installation User has Administrator Privileges on Windows Operating Systems](#).

About Non-Default User Permissions on UNIX Operating Systems

Changing the default permission setting reduces the security of the installation and your system. Oracle does not recommend that change the default permission settings.

If other users require access to a particular file or executable, use the UNIX `sudo` command or other similar commands to change the file permissions.

Refer to your UNIX operating system Administrator's Guide or contact your operating system vendor, if you need further assistance.

Verifying that the Installation User has Administrator Privileges on Windows Operating Systems

To update the Windows Registry, you must have administrator privileges.

By default, users with the administrator privilege sign in to the system with regular privileges, but can request elevated permissions to perform administrative tasks.

To perform a task with elevated privileges:

1. Find the Command Prompt icon, either from the Start menu or the Windows icon in the lower-left corner.
2. Right-click **Command Prompt** and select **Run as administrator**.

This opens a new command prompt window, and all actions performed in this window are done with administrator privileges.

 **Note:**

If you have User Access Control enabled on your system, you may see an additional window asking you to confirm this action. Confirm and continue with this procedure.

3. Perform the desired task.

For example, to start the product installer:

For a jar file, enter:

```
java -jar distribution_name.jar
```

For an executable (.exe, .bin, or .sh file), enter:

```
distribution_name.exe
```

Understanding Directories for Installation and Configuration

During the installation and domain configuration process, you must plan on providing the locations for these directories: Oracle home, Domain home, and the Application home.

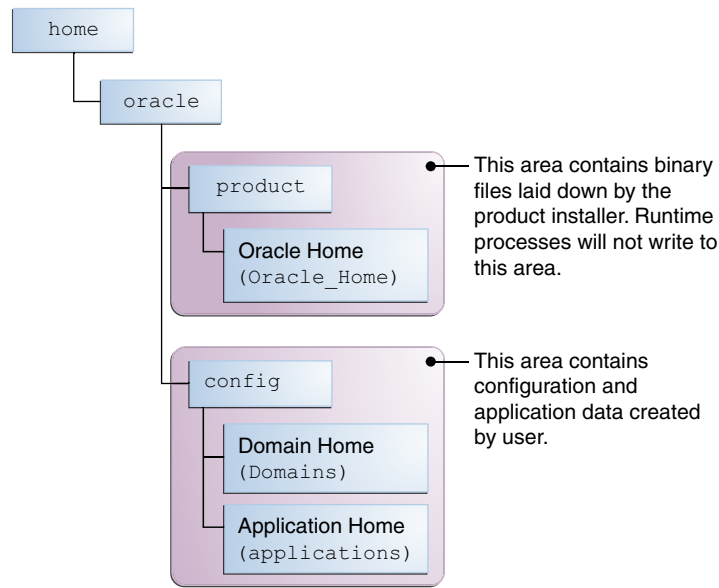
- [Understanding the Recommended Directory Structure](#)
Oracle recommends specific locations for the Oracle Home, Domain Home, and Application Home.
- [About the Oracle Home Directory](#)
When you install any Oracle Fusion Middleware product, you must use an Oracle home directory.
- [About the Domain Home Directory](#)
The Domain home is the directory where domains that you configure are created.
- [About the Application Home Directory](#)
The Application home is the directory where applications for domains you configure are created.
- [Installing Multiple Products in the Same Domain](#)
There are two methods to install and configure multiple products in one domain. This is also known as *extending* a domain.
- [Preparing for Shared Storage](#)
Oracle Fusion Middleware allows you to configure multiple WebLogic Server domains from a single Oracle home. This allows you to install the Oracle home in a single location on a shared volume and reuse the Oracle home for multiple host installations.

Understanding the Recommended Directory Structure

Oracle recommends specific locations for the Oracle Home, Domain Home, and Application Home.

Oracle recommends a directory structure similar to the one shown in [Figure 2-4](#).

Figure 2-4 Recommended Oracle Fusion Middleware Directory Structure



A base location (Oracle base) should be established on your system (for example, `/home/oracle`) and from there, two separate branches should be created. The `product` directory should contain the product binary files and all of the Oracle home directories. The `config` directory should contain your domain and application data.

Oracle recommends that you do not keep your configuration data anywhere underneath the Oracle home; if you upgrade your product to another major release, you will be required to create a new Oracle home for binaries. You must also make sure that your configuration data exist in a location to which the binaries in the Oracle home have access.

The `/home/oracle/product` (for the Oracle home) and `/home/oracle/config` (for the application and configuration data) directories are used in examples throughout the documentation; be sure to replace these directories with the actual directories on your system.

About the Oracle Home Directory

When you install any Oracle Fusion Middleware product, you must use an Oracle home directory.

This directory is a repository for common files that are used by multiple Fusion Middleware products installed on the same machine. These files ensure that Fusion Middleware operates correctly on your system. They facilitate checking of cross-product dependencies during installation. For this reason, you can consider the Oracle home directory a *central support directory* for all Oracle Fusion Middleware products installed on your system.

Fusion Middleware documentation refers to the Oracle home directory as `ORACLE_HOME`.

Oracle Home Considerations

Keep the following in mind when you create the Oracle home directory and install Fusion Middleware products:

- Do not include spaces in the name of your Oracle home directory; the installer displays an error message if your Oracle home directory path contains spaces.
- You can install only one instance of each Oracle Fusion Middleware product in a single Oracle home directory. If you need to maintain separate versions of a product on the same machine, each version must be in its own Oracle home directory.

Although you can have several different products in a single Oracle home, only one version of each product can be in the Oracle home.

Multiple Home Directories

Although in most situations, a single Oracle home directory is sufficient, it is possible to create more than one Oracle home directory. For example, you need to maintain multiple Oracle home directories in the following situations:

- You prefer to maintain separate development and production environments, with a separate product stack for each. With two directories, you can update your development environment without modifying the production environment until you are ready to do so.
- You want to maintain two different versions of a Fusion Middleware product at the same time. For example, you want to install a new version of a product while keeping your existing version intact. In this case, you must install each product version in its own Oracle home directory.
- You need to install multiple products that are not compatible with each other. See Oracle Fusion Middleware 12c (12.2.1.3.0) Interoperability and Compatibility in *Understanding Interoperability and Compatibility*.

Note:

If you create more than one Oracle home directory, you must provide non-overlapping port ranges during the configuration phase for each product.

About the Domain Home Directory

The Domain home is the directory where domains that you configure are created.

The default Domain home location is `ORACLE_HOME/user_projects/domains/domain_name`. However, Oracle strongly recommends that you do not use this default location. Put your Domain home *outside* of the Oracle home directory, for example, in `/home/oracle/config/domains`. The `config` directory should contain domain and application data. Oracle recommends a separate domain directory so that new installs, patches, and other operations update the `ORACLE_HOME` only, *not* the domain configuration.

See [Understanding the Recommended Directory Structure](#) for more on the recommended directory structure and locating your Domain home.

Fusion Middleware documentation refers to the Domain home directory as `DOMAIN_HOME` and includes all folders up to and including the domain name. For example, if you name your domain `exampledomain` and locate your domain data in the `/home/oracle/config/domains` directory, the documentation would use `DOMAIN_HOME` to refer to `/home/oracle/config/domains/exampledomain`.

About the Application Home Directory

The Application home is the directory where applications for domains you configure are created.

The default Application home location is `ORACLE_HOME/user_projects/applications/domain_name`. However, Oracle strongly recommends that you locate your Application home *outside* of the Oracle home directory; if you upgrade your product to another major release, you must create a new Oracle home for binaries.

See [Understanding the Recommended Directory Structure](#) for more on the recommended directory structure and locating your Application home..

Fusion Middleware documentation refers to the Application home directory as `APPLICATION_HOME` and includes all folders up to and including the domain name. For example, if you name your domain `exampledomain` and you locate your application data in the `/home/oracle/config/applications` directory, the documentation uses `APPLICATION_HOME` to refer to `/home/oracle/config/applications/exampledomain`.

Installing Multiple Products in the Same Domain

There are two methods to install and configure multiple products in one domain. This is also known as *extending* a domain.

- **Method 1.**

Install and configure Product A, including creating the schemas and starting all servers in the domain to verify a successful domain configuration.

This is the method used in all installation guides in the Fusion Middleware library. You can repeat this process for as many products as necessary. It allows you to validate one product at a time and add more products incrementally.

To install Product B in the same domain as Product A:

1. Stop all servers to prevent any updates to the domain while you add the new product.

See Starting and Stopping Oracle Fusion Middleware in *Administering Oracle Fusion Middleware*.

2. Follow the instructions in the installation guide for Product B, including creating the necessary schemas.

3. Run the Configuration Wizard to configure the domain.

During configuration, the Configuration Wizard automatically detects the components that have been installed and offers you the option to extend the existing Product A domain to include Product B.

- **Method 2.**

Install all of the required products, then create the schemas for all of the products. After you create the schemas, configure the domain by using the necessary product templates, then start all the servers.

This method of creating a multi-product domain may be slightly faster than Method 1; however, the installation guides in the Fusion Middleware library do not provide specific instructions for this method of domain creation.

 **See Also:**

- To update WebLogic domains, see *Updating WebLogic Domains in Creating WebLogic Domains Using the Configuration Wizard*.
- For important information regarding the ability of Oracle Fusion Middleware products to function with previous versions of other Oracle Fusion Middleware, Oracle, or third-party products, see *Oracle Fusion Middleware 12c (12.2.1.3.0) Interoperability and Compatibility in Understanding Interoperability and Compatibility*.

Preparing for Shared Storage

Oracle Fusion Middleware allows you to configure multiple WebLogic Server domains from a single Oracle home. This allows you to install the Oracle home in a single location on a shared volume and reuse the Oracle home for multiple host installations.

If you plan to use shared storage in your environment, see *Using Shared Storage in High Availability Guide* for more information.

About JDK Requirements for an Oracle Fusion Middleware Installation

Most Fusion Middleware products are in `.jar` file format. These distributions do not include a JDK. To run a `.jar` distribution installer, you must have a certified JDK installed on your system.

Make sure that the JDK is installed *outside* of the Oracle home. If you install the JDK under the Oracle home, you may encounter problems when you try to perform tasks in the future. Oracle Universal Installer validates that the Oracle home directory is empty; the install does not progress until you specify an empty directory. Oracle recommends that you locate your JDK installation in the `/home/oracle/products/jdk` directory.

Platform-specific distributions have a `.bin` (for UNIX operating systems) or `.exe` (for Windows operating systems) installer; in these cases, a platform-specific JDK is in the distribution and you do not need to install a JDK separately. However, you may need to upgrade this JDK to a more recent version, depending on the JDK versions that are certified.

Always verify the required JDK version by reviewing the certification information on the *Oracle Fusion Middleware Supported System Configurations* page. For 12c (12.2.1.3.0), the certified JDK is 1.8.0_131 and later.

To download the required JDK, navigate to the following URL and download the Java SE JDK:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

About Database Requirements for an Oracle Fusion Middleware Installation

Many Oracle Fusion Middleware products require database schemas prior to configuration. If you do not already have a database where you can install these schemas, you must install and configure a certified database.

To find a certified database for your operating system, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page on the Oracle Technology Network (OTN).

To make sure that your database is properly configured for schema creation, see *Repository Creation Utility Requirements* in the *Oracle Fusion Middleware System Requirements and Specifications* document.

After your database is properly configured, you use the Repository Creation Utility (RCU) to create product schemas in your database. This tool is available in the Oracle home for your Oracle Fusion Middleware product. See *About the Repository Creation Utility* in *Creating Schemas with the Repository Creation Utility*.

Obtaining the Product Distribution

The Oracle HTTP Server distribution is available as a standalone executable installation program that includes a Java Runtime Environment.

To locate and download the Oracle HTTP Server distribution, see *Obtaining Product Distributions* in *Planning an Installation of Oracle Fusion Middleware*.

3

Installing the Oracle HTTP Server Software

Follow the steps in this section to install the Oracle HTTP Server software. Before beginning the installation, ensure that you have verified the prerequisites and completed all steps covered in [Preparing to Install and Configure Oracle HTTP Server](#).

- [Starting the Installation Program](#)
You can start the installation program on UNIX or Windows.
- [Navigating the Installation Screens](#)
The installer shows a series of screens where you verify or enter information.
- [Verifying the Installation](#)
After you complete the installation, verify whether it was successful by completing a series of tasks.

Starting the Installation Program

You can start the installation program on UNIX or Windows.

To start the installation program:

1. Sign in to the host system.
2. Go to the directory where you downloaded the installation program.
3. Enter the following command:
 - (UNIX) `./fmw_12.2.1.3.0_ohs_linux64.bin`
 - (Windows) `fmw_12.2.1.3.0_ohs_win64.exe`

Note:

You will not be able to execute `./fmw_12.2.1.3.0_ohs_linux64.bin` if it does not have execute permission. Make sure to check and grant execute permission before running this command.

When the installation program appears, you are ready to begin the installation.

Navigating the Installation Screens

The installer shows a series of screens where you verify or enter information.

The following table lists the order in which installer screens appear. If you need additional help with an installation screen, click **Help**.

Table 3-1 Oracle HTTP Server Install Screens

Screen	Description
Installation Inventory Setup	<p>On UNIX operating systems, this screen opens if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location.</p> <p>See About the Oracle Central Inventory in <i>Installing Software with the Oracle Universal Installer</i>.</p> <p>This screen does not appear on Windows operating systems.</p>
Welcome	Review the information to make sure that you have met all the prerequisites, then click Next .
Auto Updates	Select to skip automatic updates, select patches, or search for the latest software updates, including important security updates, through your My Oracle Support account.
Installation Location	<p>Specify your Oracle home directory location.</p> <p>You can click View to verify and ensure that you are installing Oracle HTTP Server in the correct Oracle home.</p>
Installation Type	Select either Standalone HTTP Server (Managed independently of WebLogic server) or Collocated HTTP Server (Managed through WebLogic server) . See About Oracle HTTP Server for more information on these two installation types.
Prerequisite Checks	<p>This screen verifies that your system meets the minimum necessary requirements.</p> <p>To view the list of tasks that gets verified, select View Successful Tasks. To view log details, select View Log. If any prerequisite check fails, then an error message appears at the bottom of the screen. Fix the error and click Rerun to try again. To ignore the error or the warning message and continue with the installation, click Skip (not recommended).</p>
Installation Summary	<p>Use this screen to verify installation options you selected. If you want to save these options to a response file, click Save Response File and enter the response file location and name. The response file collects and stores all the information that you have entered, and enables you to perform a silent installation (from the command line) at a later time.</p> <p>Click Install to begin the installation.</p>
Installation Progress	<p>This screen shows the installation progress.</p> <p>When the progress bar reaches 100% complete, click Finish to dismiss the installer, or click Next to see a summary.</p>
Installation Complete	This screen displays the Installation Location and the Feature Sets that are installed. Review this information and click Finish to close the installer.

Verifying the Installation

After you complete the installation, verify whether it was successful by completing a series of tasks.

- [Reviewing the Installation Log Files](#)
Review the contents of the installation log files to make sure that the installer did not encounter any problems.

- [Checking the Directory Structure](#)
The contents of your installation vary based on the options you selected during the installation.
- [Viewing the Contents of the Oracle Home](#)
You can view the contents of the Oracle home directory by using the `viewInventory` script.

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that the installer did not encounter any problems.

By default, the installer writes logs files to the `Oracle_Inventory_Location/logs` (on UNIX operating systems) or `Oracle_Inventory_Location\logs` (on Windows operating systems) directory.

For a description of the log files and where to find them, see *Installation Log Files* in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

The contents of your installation vary based on the options you selected during the installation.

To verify that your installation completed successfully, you can compare your directory structure to the one shown in [Oracle HTTP Server Directory Structure](#).

For more information about the directory structure after installation, see *What are the Key Oracle Fusion Middleware Directories?* in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of the Oracle Home

You can view the contents of the Oracle home directory by using the `viewInventory` script.

See *Viewing the Contents of an Oracle Home* in *Installing Software with the Oracle Universal Installer*.

4

Configuring Oracle HTTP Server in a WebLogic Server Domain

After you have installed Oracle HTTP Server, you can extend an existing Infrastructure domain with Oracle HTTP Server.



Note:

Perform the configuration steps in this section only if you selected **Collocated HTTP Server (Managed through WebLogic server)** in the Installation Type screen. If you selected **Standalone HTTP Server (Managed independently of WebLogic server)**, see [Configuring Oracle HTTP Server in a Standalone Domain](#).

For information about Oracle HTTP Server domain configuration options, see [About Oracle HTTP Server](#).

If you have not created the Oracle Fusion Middleware Infrastructure domain, you can create it at the same time as Oracle HTTP Server. For instructions, see *Configuring the Oracle Fusion Middleware Infrastructure Domain* in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*. However, in **Selecting Advanced Configurations**, ensure that you select **System Components** too.

- [Stopping the Administration Server](#)
Before starting the configuration wizard, it is important to shut down the Administration Server. This prevents any configuration locks, saves, or activations from occurring during the configuration of the domain.
- [Starting the Configuration Wizard](#)
Start the Configuration Wizard to begin configuring a domain.
- [Navigating the Configuration Wizard Screens to Extend the Domain](#)
Use the Configuration Wizard to extend an existing Infrastructure domain with an Oracle HTTP Server instance.
- [Starting the Servers](#)
After configuration is complete, start Node Manager, then the WebLogic Administration Server and Managed Servers.

Stopping the Administration Server

Before starting the configuration wizard, it is important to shut down the Administration Server. This prevents any configuration locks, saves, or activations from occurring during the configuration of the domain.

To stop the Administration Server, go to the `DOMAIN_HOME/bin` directory.

- On UNIX operating systems:

```
./stopWebLogic.sh
```

- On Windows operating systems:

```
stopWebLogic.cmd
```

You may be prompted for the login credentials of the Administrator user you provided when you configured the Oracle Fusion Middleware Infrastructure domain.

Starting the Configuration Wizard

Start the Configuration Wizard to begin configuring a domain.

To start the Configuration Wizard:

1. Change to the following directory:

```
(UNIX) ORACLE_HOME/oracle_common/common/bin
```

```
(Windows) ORACLE_HOME\oracle_common\common\bin
```

where *ORACLE_HOME* is your 12c (12.2.1.3.0) Oracle home.

2. Enter the following command:

```
(UNIX) ./config.sh
```

```
(Windows) config.cmd
```

Navigating the Configuration Wizard Screens to Extend the Domain

Use the Configuration Wizard to extend an existing Infrastructure domain with an Oracle HTTP Server instance.



Note:

When you are adding an Oracle HTTP Server instance to an existing domain, verify that the ports you are assigning to any new components are available. The Configuration Wizard does not perform automatic port allocation.

The following sections step through the Configuration Wizard:

- [Selecting the Domain Type and Domain Home Location](#)
You must specify the domain to be updated with Oracle HTTP Server.
- [Selecting the Configuration Template for Oracle HTTP Server](#)
Use the Templates screen to select the template you require.
- [Configuring High Availability Options](#)
Use this screen to configure service migration and persistence settings that affect high availability. This screen appears for the first time when you create a cluster that uses automatic service migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

- [Specifying the Database Configuration Type](#)
Use the Database Configuration type screen to specify details about the database and database schema.
- [Specifying JDBC Component Schema Information](#)
Use the JDBC Component Schema screen to verify or specify details about the database schemas.
- [Testing the JDBC Connections](#)
Use the JDBC Component Schema Test screen to test the data source connections.
- [Selecting Advanced Configuration](#)
Use the Advanced Configuration screen to complete the domain configuration.
- [Specifying the Oracle HTTP Server Details](#)
Use the OHS Server screen to configure the Oracle HTTP Server instance in the domain.
- [Assigning System Components to Machines](#)
Use the Assign System Components to Machines screen to assign Oracle HTTP Server system components to the machine that you defined when you configured the initial Fusion Middleware Infrastructure domain.
- [Reviewing Your Configuration Specifications and Configuring the Domain](#)
The Configuration Summary screen has detailed configuration information for the domain you are about to extend.
- [Writing Down Your Domain Home and Administration Server URL](#)
The End of Configuration screen shows information about the domain you just configured.

Selecting the Domain Type and Domain Home Location

You must specify the domain to be updated with Oracle HTTP Server.

To specify the Domain type and Domain home directory:

1. On the Configuration Type screen, select **Update an existing domain**.
2. In the Domain Location field, select the Domain home location specified while configuring the Infrastructure domain.

For more information about this screen, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Selecting the Configuration Template for Oracle HTTP Server

Use the Templates screen to select the template you require.

On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select one of the following templates:

- For a Full-JRF configuration, select Oracle HTTP Server (Collocated) - 12.2.1.3.0 [ohs].
- For a Restricted-JRF configuration, select Oracle HTTP Server (Restricted JRF) - 12.2.1.3.0 [ohs].

Selecting this template automatically selects the following as dependencies:

- Oracle Enterprise Manager-Restricted JRF
- Oracle Restricted JRF

- WebLogic Coherence Cluster Extension

If you are unsure what operational mode you want, see [About Full-JRF and Restricted-JRF Modes](#) for an overview of both modes.

For more information about this screen, see Templates in *Creating WebLogic Domains Using the Configuration Wizard*

Configuring High Availability Options

Use this screen to configure service migration and persistence settings that affect high availability. This screen appears for the first time when you create a cluster that uses automatic service migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

Enable Automatic Service Migration

Select **Enable Automatic Service Migration** to enable pinned services to migrate automatically to a healthy Managed Server for failover. It configures migratable target definitions that are required for automatic service migration and the cluster leasing. Choose one of these cluster leasing options:

- Database Leasing - Managed Servers use a table on a valid JDBC System Resource for leasing. Requires that the Automatic Migration data source have a valid JDBC System Resource. If you select this option, the Migration Basis is configured to Database and the Data Source for Automatic Migration is also automatically configured by the Configuration Wizard. If you have a high availability database, such as Oracle RAC, to manage leasing information, configure the database for server migration according to steps in [High-availability Database Leasing](#).
- Consensus Leasing - Managed Servers maintain leasing information in-memory. You use Node Manager to control Managed Servers in a cluster. (All servers that are migratable, or which could host a migratable target, must have a Node Manager associated with them.) If you select this option, the Migration Basis is configured to Consensus by the Configuration Wizard.

See Leasing for more information on leasing.

See Service Migration for more information on Automatic Service Migration.

JTA Transaction Log Persistence

This section has two options: **Default Persistent Store** and **JDBC TLog Store**.

- Default Persistent Store - Configures the JTA Transaction Log store of the servers in the default file store.
- JDBC TLog Store - Configures the JTA Transaction Log store of the servers in JDBC stores.

Oracle recommends that you select **JDBC TLog Store**. When you complete the configuration, you have a cluster where JDBC persistent stores are set up for Transaction logs.

For more details on persistent and TLOG stores, see the following topics in *Developing JTA Applications for Oracle WebLogic Server*:

- Using the Default Persistent Store

- Using a JDBC TLOG Store

JMS Service Persistence

A persistent **JMS store** is a physical repository for storing persistent message data and durable subscribers. It can be either a disk-based **file store** or a JDBC-accessible database. You can use a **JMS file store** for paging of messages to disk when memory is exhausted.

- JMS File Store - Configures a component to use JMS File Stores. If you select this option, you can choose the **File Store** option in the Advanced Configuration Screen to change the settings, if required. In the File Stores screen, you can set file store names, directories, and synchronous write policies.
- JMS JDBC Store - Configures a component to use JDBC stores for all its JMS servers. When you complete the configuration, you have a cluster and JDBC persistent stores are configured for the JMS servers.

Specifying the Database Configuration Type

Use the Database Configuration type screen to specify details about the database and database schema.

On the Database Configuration type screen, all fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. Verify and ensure that credentials in all the fields are the same that you have provided while configuring Oracle Fusion Middleware Infrastructure.

Verify the database connection information, then select **Get RCU Configuration**. The following output in the Connection Result Log indicates that the operation succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK
```

Successfully Done.

For more information about the schema installed when RCU is run, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.

For more information about this screen, see Database Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Specifying JDBC Component Schema Information

Use the JDBC Component Schema screen to verify or specify details about the database schemas.

Verify that the values populated on the JDBC Component Schema screen are correct for all schemas. If you selected **RCU Data** on the previous screen, the schema table should already be populated appropriately.

For high availability environments, see the following sections in *High Availability Guide* for additional information on configuring data sources for Oracle RAC databases:

- Configuring Active GridLink Data Sources with Oracle RAC
- Configuring Multi Data Sources

See JDBC Component Schema in *Creating WebLogic Domains Using the Configuration Wizard* for more details about this screen.

Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections.

A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

By default, the schema password for each schema component is the password you specified while creating your schemas. If you want different passwords for different schema components, manually edit them in the previous screen (JDBC Component Schema) by entering the password you want in the **Schema Password** column, against each row. After specifying the passwords, select the check box corresponding to the schemas that you changed the password in and test the connection again.

For more information about this screen, see JDBC Component Schema Test in *Creating WebLogic Domains Using the Configuration Wizard*.

Selecting Advanced Configuration

Use the Advanced Configuration screen to complete the domain configuration.

On the Advanced Configuration screen, select:

- System Components

Specifying the Oracle HTTP Server Details

Use the OHS Server screen to configure the Oracle HTTP Server instance in the domain.

In the Listen Address field, specify the IP address of the host on which the Oracle HTTP Server instance will reside. Do not use `localhost`. See Specifying the Listen Address in *Creating WebLogic Domains Using the Configuration Wizard*.

All of the remaining fields are pre-populated, but you can change the values as required. For example, verify that the default port numbers are not used by other services.

For more information about this screen, see OHS Server in *Creating WebLogic Domains Using the Configuration Wizard*.

Assigning System Components to Machines

Use the Assign System Components to Machines screen to assign Oracle HTTP Server system components to the machine that you defined when you configured the initial Fusion Middleware Infrastructure domain.

On the Assign System Components to Machines screen:

1. In the **Machine** list box, select the machine (for example, `infra_machine_1`) to which you want to assign the Oracle HTTP Server instance.

2. Assign the Oracle HTTP Server instance to the selected machine in one of the following ways:

- Double-click `ohs1` in the **System Components** list box.
- Select the `ohs1` in the **System Components** list box and click the right arrow.

You can associate an Oracle HTTP Server machine to an existing machine or create a new machine. You typically create additional machines when you are planning to scale out the topology to multiple host computers. For more information, see *Scaling Out a Topology (Machine Scale Out)* in *High Availability Guide*.

Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen has detailed configuration information for the domain you are about to extend.

Review each item on the screen and verify that the information is correct. To make any changes, go back to a screen by clicking the **Back** button or selecting the screen in the navigation pane. Domain update does not start until you click **Update**.

For more information about this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Writing Down Your Domain Home and Administration Server URL

The End of Configuration screen shows information about the domain you just configured.

Make a note of the following items because you need them later:

- Domain Location
- Administration Server URL

You need the domain location to access scripts that start Node Manager and Administration Server, and you need the URL to access the Administration Server.

Click **Finish** to dismiss the Configuration Wizard.

Starting the Servers

After configuration is complete, start Node Manager, then the WebLogic Administration Server and Managed Servers.

For more information on additional tools you can use to manage your domain, see Overview of Oracle Fusion Middleware Administration Tools in *Administering Oracle Fusion Middleware*.

- [Starting Node Manager](#)
- [Starting the Administration Server](#)
- [Starting the Oracle HTTP Server Instance](#)

Starting Node Manager

To start the per-domain Node Manager:

1. Change to the following directory:

- (UNIX) `setenv JAVA_HOME /home/Oracle/Java/jdk1.8.0_131`
 - (Windows) `set JAVA_HOME=C:\home\Oracle\Java\jdk1.8.0_131`
2. Enter the following command:

- (UNIX) Using `nohup` and `nm.out` as an example output file:

```
nohup ./startNodeManager.sh > LOG_DIR/nm.out&
```

where `LOG_DIR` is the location of directory in which you want to store the log files.

- (Windows) `startNodeManager.cmd`

 **Note:**

On Windows operating systems, Oracle recommends that you configure Node Manager to run as a startup service. This allows Node Manager to start up automatically each time the system is restarted.

See *Running Node Manager as a Startup Service* in *Administering Node Manager for Oracle WebLogic Server*.

Starting the Administration Server

To start the Administration Server:

1. Change to the `DOMAIN_HOME/bin`
2. Enter the following:
 - (UNIX) `./startWebLogic.sh`
 - (Windows) `startWebLogic.cmd`

If you selected **Production Mode** on the Domain Mode and JDK screen when you created the domain, you see a prompt for the Administrator user login credentials as provided on the Administrator Account screen.

For more information about starting the Administration Server, see *Starting and Stopping Administration Servers* in *Administering Oracle Fusion Middleware*.

 **Tip:**

In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. See *Creating a Boot Identity File for an Administration Server* in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

You can verify that the Administration Server is up and running by accessing the Administration Server Console. The URL is provided on the Configuration Success screen, as described in “Writing Down Your Domain Home and Administration Server URL” in [Navigating the Configuration Wizard Screens to Configure the Domain](#)

(`http://administration_server_host:administration_server_port/console`). The default Administration Server port number is 7001.



Note:

Make sure that the database hosting your product schemas is up and running and accessible by the Administration Server.

For more information about how to use the Administration Console, see *Getting Started Using Oracle WebLogic Server Administration Console in Administering Oracle Fusion Middleware*.

Starting the Oracle HTTP Server Instance

To start the Oracle HTTP Server instance:

1. Sign in to Fusion Middleware Control:

`http://administration_server_host:administration_server_port/em`

The Administration Server host and port number were in the URL on the Configuration Success screen. The default Administration Server port number is 7001.

The login credentials were provided on the Administrator Account screen while installing.

2. Click **WebTier** in the navigation panel on the left side.
3. Select the Oracle HTTP Server instance you want to start (for example, `ohs1`) and click **Start Up** from the top panel.

For more information about managing Oracle HTTP Server using Oracle Enterprise Manager Fusion Middleware Control, see *Manage Oracle HTTP Server in Administering Oracle Fusion Middleware with Fusion Middleware Control*.

For other tools and instructions, see *Starting Oracle HTTP Server Instances in Administering Oracle HTTP Server*.

5

Configuring Oracle HTTP Server in a Standalone Domain

After you have installed Oracle HTTP Server, you can configure it in a standalone domain. For more information on other methods available for domain creation, see *Additional Tools for Creating, Extending, and Managing WebLogic Domains* in *Creating WebLogic Domains Using the Configuration Wizard*.



Note:

Perform the configuration steps in this section only if you have selected **Standalone HTTP Server (Managed independently of WebLogic server)** in the Installation Type screen. If you have selected **Collocated HTTP Server (Managed through WebLogic server)**, see [Configuring Oracle HTTP Server in a WebLogic Server Domain](#).

For information about Oracle HTTP Server domain configuration options, see [About Oracle HTTP Server](#).

The following sections step through the Configuration Wizard:

- [Starting the Configuration Wizard](#)
Start the Configuration Wizard to begin configuring a domain.
- [Navigating the Configuration Wizard Screens to Configure the Domain](#)
Use the Configuration Wizard to configure Oracle HTTP Server in a standalone domain.
- [Starting the Instances](#)
After configuration is complete, you can use tools to manage your domain.

Starting the Configuration Wizard

Start the Configuration Wizard to begin configuring a domain.

To start the Configuration Wizard:

1. Change to the following directory:
(UNIX) `ORACLE_HOME/oracle_common/common/bin`
(Windows) `ORACLE_HOME\oracle_common\common\bin`
where `ORACLE_HOME` is your 12c (12.2.1.3.0) Oracle home.
2. Enter the following command:
(UNIX) `./config.sh`
(Windows) `config.cmd`

Navigating the Configuration Wizard Screens to Configure the Domain

Use the Configuration Wizard to configure Oracle HTTP Server in a standalone domain.

Note:

When you are creating an Oracle HTTP Server instance for a new domain, verify that the ports you are assigning to your components are available. The Configuration Wizard does not perform automatic port allocation.

- [Selecting the Domain Type and Domain Home Location](#)
Use the Configuration Type screen to select a Domain home directory location, optimally outside the Oracle home directory.
- [Selecting the Configuration Template for Oracle HTTP Server](#)
Use the Templates screen to select the template you require.
- [Specifying the Domain Mode and JDK](#)
Use the Domain Mode and JDK screen to specify the domain mode and Java Development Kit (JDK).
- [Adding System Components](#)
Use the System Components screen to specify details about the new Oracle HTTP Server instance.
- [Specifying the Oracle HTTP Server Details](#)
Use the OHS Server screen to configure the Oracle HTTP Server instance in the domain.
- [Configuring Node Manager](#)
Use the Node Manager screen to select the type of Node Manager you want to configure, along with the Node Manager credentials.
- [Reviewing Your Configuration Specifications and Configuring the Domain](#)
The Configuration Summary screen shows detailed configuration information for the domain you are about to create.

Selecting the Domain Type and Domain Home Location

Use the Configuration Type screen to select a Domain home directory location, optimally outside the Oracle home directory.

Oracle recommends that you locate your Domain home in accordance with the directory structure in *What Are the Key Oracle Fusion Middleware Directories?* in *Understanding Oracle Fusion Middleware*, where the Domain home is located outside the Oracle home directory. This directory structure helps avoid issues when you need to upgrade or reinstall software.

To specify the Domain type and Domain home directory:

1. On the Configuration Type screen, select **Create a new domain**.

2. In the Domain Location field, specify your Domain home directory.

For more details about this screen, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Selecting the Configuration Template for Oracle HTTP Server

Use the Templates screen to select the template you require.

On the Templates screen, make sure **Create Domain Using Product Templates** is selected, then select the following template:

- Oracle HTTP Server (Standalone) - 12.2.1.3.0 [ohs]

For more information about this screen, see Templates in *Creating WebLogic Domains Using the Configuration Wizard*

Specifying the Domain Mode and JDK

Use the Domain Mode and JDK screen to specify the domain mode and Java Development Kit (JDK).

On the Domain Mode and JDK screen:

- Select **Production** in the **Domain Mode** field.
- Select the **Oracle HotSpot JDK** in the **JDK** field.

For more information about this screen, see Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.

Adding System Components

Use the System Components screen to specify details about the new Oracle HTTP Server instance.

On the System Components screen:

1. Specify `ohs1` in the **System Component** field.
2. Specify `OHS` in the **Component Type** field.
3. **Restart Interval Seconds**: Specify the number of seconds to wait before attempting a restart if an application is not responding.
4. **Restart Delay Seconds**: Specify the number of seconds to wait between restart attempts.

Note:

You can rename the Oracle HTTP Server system component name (for example, `ohs_1`), when you rerun the configuration wizard and select **Update an existing domain** in the Configuration Type screen. But when you do this, the original Oracle HTTP Server component name and all customized values, like port designation, gets deleted and a new system component with default values get created. So, avoid renaming the Oracle HTTP Server component, unless you want the default settings.

For more information about this screen, see System Components in *Creating WebLogic Domains Using the Configuration Wizard*.

Specifying the Oracle HTTP Server Details

Use the OHS Server screen to configure the Oracle HTTP Server instance in the domain.

In the Listen Address field, specify the IP address of the host on which the Oracle HTTP Server instance will reside. Do not use `localhost`. See Specifying the Listen Address in *Creating WebLogic Domains Using the Configuration Wizard*.

All of the remaining fields are pre-populated, but you can change the values as required. For example, verify that the default port numbers are not used by other services.

For more information about this screen, see OHS Server in *Creating WebLogic Domains Using the Configuration Wizard*.

Configuring Node Manager

Use the Node Manager screen to select the type of Node Manager you want to configure, along with the Node Manager credentials.

Select **Per Domain Default Location** as the Node Manager type, then specify Node Manager credentials.

For more information about this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.

For more about Node Manager types, see Node Manager Overview in *Administering Node Manager for Oracle WebLogic Server*.

Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen shows detailed configuration information for the domain you are about to create.

Review each item on the screen and verify that the information is correct. To make any changes, go back to a screen by clicking the **Back** button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

For more details about options on this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Starting the Instances

After configuration is complete, you can use tools to manage your domain.

- For more information on additional tools you can use to manage your domain, see Overview of Oracle Fusion Middleware Administration Tools in *Administering Oracle Fusion Middleware*.
- If you are attempting to create an Oracle HTTP Server instance that uses a TCP port in the reserved range (typically less than 1024), then you must perform some

extra configuration to allow the server to bind to privileged ports. For more information, see *Starting Oracle HTTP Server Instances on a Privileged Port (Unix Only)* in *Administering Oracle HTTP Server*.

- [Starting Node Manager](#)
- [Starting the Oracle HTTP Server Instance](#)

Starting Node Manager

To start the per-domain Node Manager:

1. Change to the following directory:
 - (UNIX) `setenv JAVA_HOME /home/Oracle/Java/jdk1.8.0_131`
 - (Windows) `set JAVA_HOME=C:\home\Oracle\Java\jdk1.8.0_131`
2. Enter the following command:

- (UNIX) Using `nohup` and `nm.out` as an example output file:

```
nohup ./startNodeManager.sh > LOG_DIR/nm.out&
```

where `LOG_DIR` is the location of directory in which you want to store the log files.

- (Windows) `startNodeManager.cmd`

Note:

On Windows operating systems, Oracle recommends that you configure Node Manager to run as a startup service. This allows Node Manager to start up automatically each time the system is restarted.

See *Running Node Manager as a Startup Service* in *Administering Node Manager for Oracle WebLogic Server*.

Starting the Oracle HTTP Server Instance

To start the Oracle HTTP Server instance:

- **On UNIX**
 1. Go to the `Domain_Home/bin` directory. `Domain_Home` is the complete path to the location where you have configured Oracle HTTP Server.

For example:

```
cd /user_projects/domains/base_domain/bin
```

2. Run the following command:

```
./startComponent.sh system_component_name
```

where `system_component_name` is the Oracle HTTP Server system component name you provided while configuring the domain.

For example:

```
./startComponent.sh ohs1
```

- **On Windows**

1. Go to the `Domain_Home\bin` directory. `Domain_Home` is the complete path to the location where you have configured Oracle HTTP Server.

For example:

```
cd \user_projects\domains\base_domain\bin
```

2. Run the following command:

```
startComponent.cmd system_component_name
```

where `system_component_name` is the Oracle HTTP Server system component name you provided while configuring the domain.

For example:

```
startComponent.cmd ohs1
```

See Starting Oracle HTTP Server Instances from the Command Line in *Administering Oracle HTTP Server*.

Missing Libraries Might Cause HTTPD to Exit Without Notice

On the Windows platform, Oracle HTTP Server requires certain Microsoft software to be installed in order to function properly.

If the following software is not installed, Oracle HTTP Server may exit with no output or generate an error message, depending on registry settings.

- Microsoft Visual C++ 2010 Feature Pack Redistributable Package (x64)
- Microsoft Visual C++ 2012 Feature Pack Redistributable Package (x64)
- Any other packages recommended by Microsoft

If Oracle HTTP Server fails to start and no error messages are logged by Oracle HTTP Server in the server error log or in Node Manager logs, follow this procedure to diagnose the problem:

1. Verify that the following runtime libraries are installed: Microsoft Visual C++ 2010 and 2012 Feature Pack Redistributable Package (x64)
 - a. If either runtime library is missing, download it from <http://www.microsoft.com>
 - b. After installing the package, install any service packs or other fixes to this package that are recommended by Microsoft.
 - c. Try to start Oracle HTTP Server again after installing the runtime library and associated service packs or fixes..
2. Attempt to start Oracle HTTP Server from a command prompt.

 **Note:**

Oracle HTTP Server does not function properly when started from a command prompt in this manner. This mechanism for starting Oracle HTTP Server is for diagnostic purposes only.

- a. Create the `DOMAIN_HOME/servers/ohs-instancename/logs` directory if it has not yet been created by Oracle HTTP Server.

- b. Try to start Oracle HTTP Server using Fusion Middleware Control or WLST.
 - c. Copy the file `DOMAIN_HOME/servers/ohs-instancename/logs/lastinvocation.log` to a new file `startohs.cmd`.
 - d. Open a new command prompt and run `startohs.cmd` in that new command prompt.
 - e. If it generates error messages after the script runs, use those messages to diagnose the startup failure.
 - f. If it does not generate error messages after the script runs, proceed to the next step.
3. Collect Windows error information when attempting to start Oracle HTTP Server.

Error reporting for Oracle HTTP Server library errors may be inhibited on your system. Refer to the following Microsoft article for information on verifying or correcting error reporting:

"HOWTO: How To Change Hard Error Popup Handling in Windows NT" at <http://support.microsoft.com/kb/128642>

- a. Change `ErrorMode` to the desired value.
 - The default mode is 1, which means library errors are not reported for Oracle HTTP Server.
 - Change the mode to 2 to enable reporting to the Windows event log. This is the recommended setting for this task.
 - Change the mode to 0 to enable hard error pop-ups. If this is changed to 0, set it back to the previous value immediately after collecting diagnostic information as it can result in operational problems if errors occur when the server is unattended.
- b. Try to start Oracle HTTP Server using Fusion Middleware Control or WLST or the `startohs.cmd` script which was created in the previous step.
- c. Check for diagnostic information in a hard error popup or in the Windows event log, depending on the setting of `ErrorMode`.

To check the Windows event log, open Windows Event Viewer and view the logs for Event Viewer (Local) / Windows Logs / Application

Find the event log entries written at the time of the attempt to start Oracle HTTP Server.

Errors locating runtime libraries might be reported as `SideBySide` errors with event ids 33 or 35, which text describing the Oracle library attempting to load the library and the specific library being loaded. Example:

```
level=Error
Source=SideBySide
Event Id = 33
Activation context generation failed for
"C:\work\Oracle\MD1212\ORACLE~1\bin\oraiau.dll".
Dependent Assembly Microsoft.VC90.CRT,processorArchitecture="amd64",
publicKeyToken="1fc8b3b9a1e18e3b",type="win32",version="9.0.21022.8"
could not be found.
Please use sxstrace.exe for detailed diagnosis.
```

The indicated library must be installed. This example shows `VC90`, which is the Microsoft Visual C++ 2010 library described in Step 1 above.

- d. Restore `ErrorMode` to the previous setting.

6

Next Steps After Configuring the Domain

After you configure a product domain, there are additional tasks that you may want to perform.

- [Next Steps for Oracle HTTP Server in a WebLogic Server Domain](#)
After configuring Oracle HTTP Server in a WebLogic Server domain, there are additional tasks that you may want to perform.
- [Next Steps for Oracle HTTP Server in a Standalone Domain](#)
After configuring Oracle HTTP Server in a standalone domain, there are additional tasks that you may want to perform.
- [Preparing Your Environment for High Availability](#)
Scaling out for high availability requires additional steps.

Next Steps for Oracle HTTP Server in a WebLogic Server Domain

After configuring Oracle HTTP Server in a WebLogic Server domain, there are additional tasks that you may want to perform.



Note:

A WebLogic Server domain can be scaled out to multiple physical machines and be centrally managed by the administration server. See Oracle HTTP Server Release Topologies in *Administering Oracle HTTP Server*.

Table 6-1 Basic Administration Tasks for Oracle HTTP Server in a WebLogic Server Domain

Task	Description	More Information
Getting familiar with Fusion Middleware administration tools	Get familiar with the various tools available, which you can use to manage your environment.	See: Oracle HTTP Server Directory Structure in a WebLogic Server Domain Oracle HTTP Server Directory Structure in a Standalone Domain Overview of Oracle Fusion Middleware Administration Tools in <i>Administering Oracle Fusion Middleware</i> .
Starting and stopping products and servers	Learn how to start and stop Oracle Fusion Middleware, including the Administration Server, Managed Servers, and components.	See Starting and Stopping Oracle Fusion Middleware in <i>Administering Oracle Fusion Middleware</i> .

Table 6-1 (Cont.) Basic Administration Tasks for Oracle HTTP Server in a WebLogic Server Domain

Task	Description	More Information
Configuring Secure Sockets Layer (SSL)	Learn how to set up secure communications among between Oracle Fusion Middleware components using SSL.	See Configuring SSL in Oracle Fusion Middleware in <i>Administering Oracle Fusion Middleware</i> .
Deploying Applications	Learn how to deploy your applications to Oracle Fusion Middleware.	See Deploying Applications in <i>Administering Oracle Fusion Middleware</i> .
Monitoring Oracle Fusion Middleware	Learn how to keep track of the status of Oracle Fusion Middleware components.	See Monitoring Oracle Fusion Middleware in <i>Administering Oracle Fusion Middleware</i> .
Understanding Backup and Recovery Procedures	Learn the recommended backup and recovery procedures for Oracle Fusion Middleware.	See Introducing Backup and Recovery in <i>Administering Oracle Fusion Middleware</i> .
Tuning and configuring Coherence for your topology.	The standard installation topology includes a Coherence cluster that contains storage-enabled Managed Coherence Servers. This configuration is a good starting point for using Coherence, but depending upon your specific requirements, consider tuning and reconfiguring Coherence to improve performance in a production environment.	See Configuring and Managing Coherence Clusters in <i>Administering Clusters for Oracle WebLogic Server</i> . For information on tuning Coherence, see Performance Tuning in <i>Administering Oracle Coherence</i> . For information on storing HTTP session data in Coherence, see Using Coherence Web with WebLogic Server in <i>Administering HTTP Session Management with Oracle Coherence*Web</i> . For more about creating and deploying Coherence applications, see Getting Started in <i>Developing Oracle Coherence Applications for Oracle WebLogic Server</i> .

Table 6-1 (Cont.) Basic Administration Tasks for Oracle HTTP Server in a WebLogic Server Domain

Task	Description	More Information
Wiring Oracle HTTP Server to the Administration Server.	Learn how to connect Oracle HTTP Server to the Administration Server so that you can access Fusion Middleware Control Console and to the Administration Console through the Oracle HTTP Server.	See Wiring Oracle HTTP Server to the Administration Server in <i>Administering Oracle Fusion Middleware</i> .
Routing Applications Through Oracle HTTP Server to Oracle WebLogic Server.	Perform these steps to connect Oracle HTTP Server so that requests are routed through Oracle HTTP Server to Oracle WebLogic Server.	See Routing Applications Through Oracle HTTP Server to Oracle WebLogic Server in <i>Administering Oracle Fusion Middleware</i> .
Routing requests to managed servers in a WebLogic Server domain	Learn how to configure Oracle WebLogic Server Proxy Plug-In to proxy requests from Oracle HTTP Server to Oracle WebLogic Server.	See Configuring the Plug-In for Oracle HTTP Server.

Next Steps for Oracle HTTP Server in a Standalone Domain

After configuring Oracle HTTP Server in a standalone domain, there are additional tasks that you may want to perform.

Table 6-2 Basic Administration Tasks for Oracle HTTP Server in a Standalone Domain

Task	Description	More Information
Getting familiar with Oracle HTTP Server management tools	Get familiar with the various tools available, which you can use to manage your environment.	See Understanding Oracle HTTP Server Management Tools in <i>Administering Oracle HTTP Server</i> .
Starting and stopping Oracle HTTP Server instances	Learn how to start and stop Oracle HTTP Server instances in a standalone domain.	See Performing Basic Oracle HTTP Server Tasks in <i>Administering Oracle HTTP Server</i> .
Routing requests to managed servers in a WebLogic Server domain	Learn how to configure Oracle WebLogic Server Proxy Plug-In to proxy requests from Oracle HTTP Server to Oracle WebLogic Server.	See Configuring the Oracle WebLogic Server Proxy Plug-In Manually.

Preparing Your Environment for High Availability

Scaling out for high availability requires additional steps.

[Table 6-3](#) provides a list of tasks to perform if you want to scale out your standard installation environment for high availability.

Table 6-3 Tasks Required to Prepare Your Environment for High Availability

Task	Description	More Information
Scaling out to multiple host computers	To enable high availability, it is important to provide failover capabilities to another host computer. That way, if one computer goes down, your environment can continue to serve the consumers of your deployed applications.	See <i>Scaling Out a Topology (Machine Scale Out)</i> in the <i>High Availability Guide</i> .
Configuring high availability for your Web Tier components.	If you have added a Web tier front-end, then you must configure the Web Tier for high availability, as well as the WebLogic Server software.	See <i>Configuring High Availability for Web Tier Components</i> in <i>High Availability Guide</i> .
Setting up a front-end load balancer	A load balancer can be used to distribute requests across servers more evenly.	See <i>Server Load Balancing in a High Availability Environment and Configuring Load Balancer Virtual Server Names and Ports</i> in <i>High Availability Guide</i> .
Configuring Node Manager	Node Manager enables you to start, shut down, and restart the Administration Server and Managed Server instances from a remote location. This document assumes you have configured a per-domain Node Manager. Review the Node Manager documentation, for information on advanced Node Manager configuration options and features.	See <i>Advanced Node Manager Configuration</i> in <i>Administering Node Manager for Oracle WebLogic Server</i> .

7

Uninstalling or Reinstalling Oracle HTTP Server

Follow the instructions in this section to uninstall or reinstall Oracle HTTP Server.



Note:

This task applies only to *standalone* Oracle HTTP Server. You cannot uninstall Oracle HTTP Server when it is collocated in a WebLogic Server domain, as partial deconfiguration of a configured domain is not supported.

Oracle recommends that you always use the instructions in this section to remove the software. If you try to remove the software manually, you may encounter problems when you try to reinstall the software again at a later time. Following the procedures in this section ensures that the software is properly removed.

- [Stopping Oracle Fusion Middleware](#)
Before running the Uninstall Wizard, Oracle recommends that you stop all servers and processes associated with the Oracle home you are going to remove.
- [Uninstalling the Software](#)
Follow the instructions in this section to start the Uninstall Wizard and remove the software.
- [Verifying the Uninstallation of Oracle HTTP Server in a Standalone Domain](#)
When Oracle HTTP Server is installed in a standalone domain, you may need to manually delete directories and files.
- [Removing the Oracle Home Directory Manually](#)
After you uninstall the software, you must manually remove your Oracle home directory and any existing subdirectories that the Uninstall Wizard did not remove.
- [Removing the Program Shortcuts on Windows Operating Systems](#)
On Windows operating systems, you must also manually remove the program shortcuts; the Deinstallation Wizard does not remove them for you.
- [Removing the Domain and Application Data](#)
After you uninstall the software, you must remove the domain and application data.
- [Reinstalling the Software](#)
You can reinstall your software into the same Oracle home as a previous installation only if you uninstalled the software by following the instructions in this section, including manually removing the Oracle home directory.

Stopping Oracle Fusion Middleware

Before running the Uninstall Wizard, Oracle recommends that you stop all servers and processes associated with the Oracle home you are going to remove.

See [Stopping an Oracle Fusion Middleware Environment](#) in *Administering Oracle Fusion Middleware*.

Uninstalling the Software

Follow the instructions in this section to start the Uninstall Wizard and remove the software.

Note:

This task applies only to *standalone* Oracle HTTP Server. You cannot uninstall Oracle HTTP Server when it is collocated in a WebLogic Server domain, as partial deconfiguration of a configured domain is not supported.

If you want to uninstall the product in a silent (command-line) mode, see [Running the Oracle Universal Installer for Silent Uninstallation](#) in *Installing Software with the Oracle Universal Installer*.

- [Starting the Uninstall Wizard](#)
- [Navigating the Uninstall Wizard Screens](#)

Starting the Uninstall Wizard

To start the Uninstall Wizard:

1. Change to the following directory:
(UNIX) `ORACLE_HOME/oui/bin`
(Windows) `ORACLE_HOME\oui\bin`
2. Enter the following command:
(UNIX) `./deinstall.sh`
(Windows) `deinstall.cmd`

Navigating the Uninstall Wizard Screens

The Uninstall Wizard shows a series of screens to confirm the removal of the software.

If you need help on screen listed in [Table 7-1](#), click **Help** on the screen.

Table 7-1 Uninstall Wizard Screens and Descriptions

Screen	Description
Welcome	Introduces you to the product Uninstall Wizard.

Table 7-1 (Cont.) Uninstall Wizard Screens and Descriptions

Screen	Description
Uninstall Summary	Shows the Oracle home directory and its contents that are uninstalled. Verify that this is the correct directory. If you want to save these options to a response file, click Save Response File and enter the response file location and name. You can use the response file later to uninstall the product in silent (command-line) mode. See Running the Oracle Universal Installer for Silent Uninstall in <i>Installing Software with the Oracle Universal Installer</i> . Click Deinstall , to begin removing the software.
Uninstall Progress	Shows the uninstallation progress.
Uninstall Complete	Appears when the uninstallation is complete. Review the information on this screen, then click Finish to close the Uninstall Wizard.

Verifying the Uninstallation of Oracle HTTP Server in a Standalone Domain

When Oracle HTTP Server is installed in a standalone domain, you may need to manually delete directories and files.

To verify the uninstallation of Oracle HTTP Server in a standalone domain:

1. Go to the `ORACLE_HOME` directory.
2. Enter `ls` to list the directories and files in `ORACLE_HOME`.
3. If any files or directories exist in `ORACLE_HOME`, delete them by entering the following command:

```
rm -rf
```

Removing the Oracle Home Directory Manually

After you uninstall the software, you must manually remove your Oracle home directory and any existing subdirectories that the Uninstall Wizard did not remove.

For example, if your Oracle home directory is `/home/Oracle/product/ORACLE_HOME` on a UNIX operating system, enter the following commands:

```
cd /home/Oracle/product
rm -rf ORACLE_HOME
```

On a Windows operating system, if your Oracle home directory is `C:\Oracle\Product\ORACLE_HOME`, use a file manager window and navigate to the `C:\Oracle\Product` directory. Right-click on the `ORACLE_HOME` folder and select **Delete**.

Removing the Program Shortcuts on Windows Operating Systems

On Windows operating systems, you must also manually remove the program shortcuts; the Deinstallation Wizard does not remove them for you.

To remove the program shortcuts on Windows:

1. Change to the following directory:
`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Oracle\ORACLE_HOME\Product`
2. If you only have one product installed in your Oracle home, delete the `ORACLE_HOME` directory. If you have multiple products installed in your Oracle home, delete all products before you delete the `ORACLE_HOME` directory.

Removing the Domain and Application Data

After you uninstall the software, you must remove the domain and application data.

To remove the domain and application data:

1. Manually remove your Domain home directory. For example:

On a UNIX operating system, if your Domain home directory is `/home/Oracle/config/domains/ohs_domain`, enter the following command:

```
cd /home/Oracle/config/domains
rm -rf ohs_domain
```

On a Windows operating system, if your Domain home directory is `C:\Oracle\Config\domains\ohs_domain`, use a file manager window and navigate to the `C:\Oracle\Config\domains` directory. Right-click on the `ohs_domain` folder and select **Delete**.

2. Manually remove your Application home directory. For example:

On a UNIX operating system, if your Application home directory is `/home/Oracle/config/applications/ohs_domain`, enter the following commands:

```
cd /home/Oracle/config/applications
rm -rf ohs_domain
```

On a Windows operating system, if your Application home directory is `C:\Oracle\Config\applications\ohs_domain`, use a file manager window and navigate to the `C:\Oracle\Config\applications` directory. Right-click on the `ohs_domain` folder and select **Delete**.

3. Back up the `domain_registry.xml` file in your Oracle home, then edit the file and remove the line associated with the domain that you are removing. For example, to remove the `ohs_domain`, find the following line and remove it:

```
<domain location="/home/Oracle/config/domains/ohs_domain"/>
```

Save and exit the file when you are finished.

Reinstalling the Software

You can reinstall your software into the same Oracle home as a previous installation only if you uninstalled the software by following the instructions in this section, including manually removing the Oracle home directory.

When you reinstall, you can then specify the same Oracle home as your previous installation.

If ODI is installed again in the same location where it was previously deleted, delete the entire Oracle Home where it was previously installed.

Consider the following cases where the Oracle home is not empty:

- Installing in an existing Oracle home that contains the same feature sets.

The installer warns you that the Oracle home that you specified during installation already contains the same software you are trying to install.

- Installing in an existing, non-empty Oracle home.

For example, suppose you chose to create your Domain home or Application home somewhere inside your existing Oracle home. This data is not removed when you uninstall a product, so if you try to reinstall into the same Oracle home, the installer does not allow it. Your options are:

- Uninstall your software from the Oracle home (as this section describes) and then remove the Oracle home directory. After you uninstall the software and remove the Oracle home directory, you can reinstall and reuse the same Oracle home location. Any domain or application data that was in the Oracle home must be re-created.
- Select a different Oracle home directory.

A

Oracle HTTP Server Directory Structure

A directory structure is created on your system at the end of your Oracle HTTP Server installation and domain configuration.

- [Oracle HTTP Server Directory Structure in a WebLogic Server Domain](#)
- [Oracle HTTP Server Directory Structure in a Standalone Domain](#)

Oracle HTTP Server Directory Structure in a WebLogic Server Domain

[Figure A-1](#) shows an example of the basic directory structure after installation and domain configuration of Oracle HTTP Server in a WebLogic Server domain is complete. Note that figure shows only important directories. There are other directories and files created too.

This example shows the recommended directory structure, as described in *Understanding the Recommended Directory Structure in [Planning an Installation of Oracle Fusion Middleware](#)*. Your directory structure may differ slightly based on where you choose to create these directories on your system.

Figure A-1 Sample Oracle HTTP Server Directory Structure in a WebLogic Server Domain

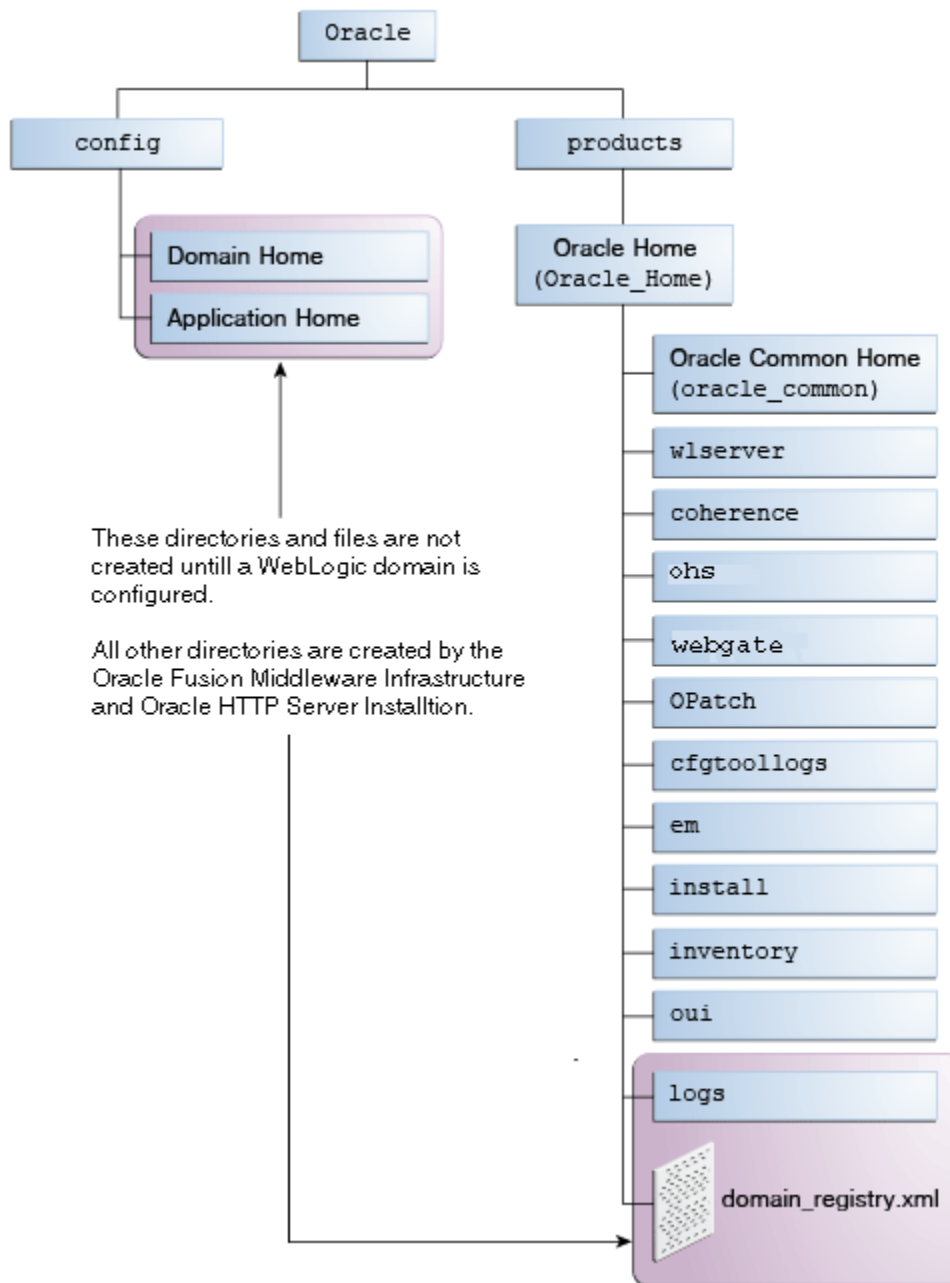


Table A-1 describes these directories in more details.

Table A-1 Directory Description for Oracle HTTP Server in a WebLogic Server Domain

Directory or file	Description
cfgtoollogs	This directory contains the log files for your installation and configuration transactions.

Table A-1 (Cont.) Directory Description for Oracle HTTP Server in a WebLogic Server Domain

Directory or file	Description
coherence	This directory contains the Coherence product files.
install	This directory contains install-related files and scripts.
inventory	This directory contains information about the components, feature sets, and patches installed in this Oracle home directory.
ohs	This directory contains the Oracle HTTP Server binaries.
OPatch	This directory contains OPatch and supported files. OPatch is a tool used to patch Oracle Fusion Middleware software.
oracle_common	This directory contains the binary and library files required for Oracle Enterprise Manager Fusion Middleware Control, Java Required Files (JRF), and Oracle HTTP Server.
orainst.loc	Text file containing the location of your Oracle inventory, and also the group ID of the user creating the inventory.
oui	This directory contains files used by the Oracle Universal Installer, including the uninstallation program. If you need to run the Oracle Universal Installer again after the product is installed, you should do so from this directory.
webgate	This directory contains the Oracle HTTP Server Webgate binaries for Oracle Access Manager (OAM).
wlserver	This directory contains a small portion of webLogic Server binaries that are used for management functionality. This directory is also referred to as the WebLogic home directory, or <i>WL_HOME</i> .
domain-registry.xml	This registry file contains the location of all domains currently registered with this WebLogic Server installation. Whenever you add a new domain, it is registered in this file.

Oracle HTTP Server Directory Structure in a Standalone Domain

Figure A-2 shows an example of the basic directory structure after installation and domain configuration of Oracle HTTP Server in a standalone domain is complete. Note that figure shows only important directories. There are other directories and files created too.

This example shows the recommended directory structure, as described in Understanding the Recommended Directory Structure in *Planning an Installation of Oracle Fusion Middleware*. Your directory structure may differ slightly based on where you choose to create these directories on your system.

Figure A-2 Sample Oracle HTTP Server Directory Structure in a Standalone Domain

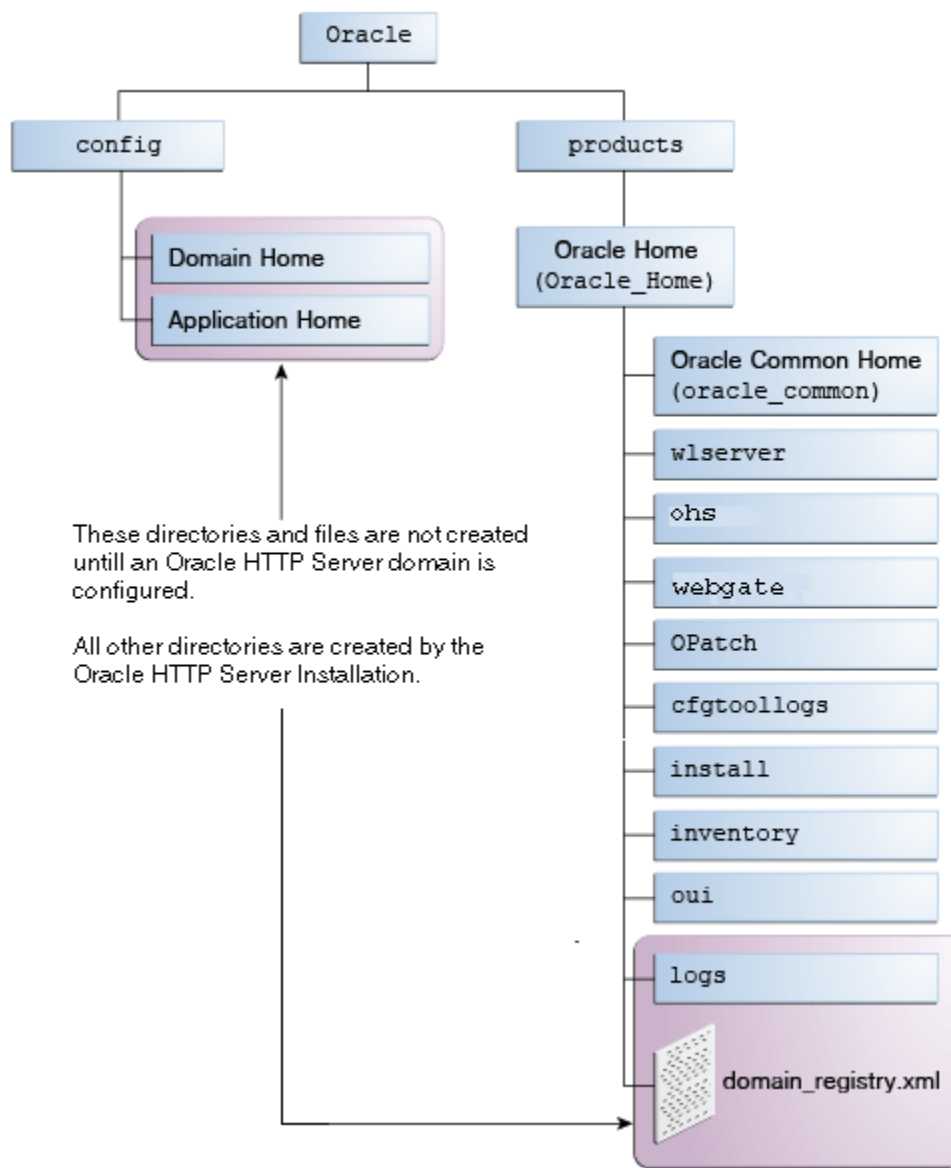


Table A-2 describes these directories in more details.

Table A-2 Directory Description for Oracle HTTP Server in a Standalone Domain

Directory or file	Description
cfgtoollogs	This directory contains the log files for your installation and configuration transactions.
install	This directory contains install-related files and scripts.
inventory	This directory contains information about the components, feature sets, and patches installed in this Oracle home directory.

Table A-2 (Cont.) Directory Description for Oracle HTTP Server in a Standalone Domain

Directory or file	Description
ohs	This directory contains the Oracle HTTP Server binaries.
OPatch	This directory contains OPatch and supported files. OPatch is a tool used to patch Oracle Fusion Middleware software.
oracle_common	This directory contains the binary and library files required for Oracle HTTP Server and WebGates.
orainst.loc	Text file containing the location of your Oracle inventory, and also the group ID of the user creating the inventory.
webgate	This directory contains the Oracle HTTP Server Webgate binaries for Oracle Access Manager (OAM).
oui	This directory contains files used by the Oracle Universal Installer, including the uninstallation program. If you need to run the Oracle Universal Installer again after the product is installed, you should do so from this directory.
wlserver	This directory contains the WebLogic Server product files. This directory is also referred to as the WebLogic home directory, or <i>WL_HOME</i> .
domain-registry.xml	This registry file contains the locations of all domains created using the templates available in this Oracle home. Whenever you add a new domain, it is registered in this file.

B

Configuring Oracle HTTP Server WebGate for Oracle Access Manager

You can configure Oracle HTTP Server WebGate that acts as a web server plug-in to intercept and forward HTTP requests to an existing Oracle Access Manager instance for authentication and authorization.

- [About Webgate](#)
WebGate is a web server plug-in that intercepts HTTP requests and forwards them to an existing Oracle Access Manager instance for authentication and authorization.
- [General Prerequisites for Configuring Oracle HTTP Server WebGate](#)
Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.
- [Configuring Oracle HTTP Server WebGate](#)
Configuring Oracle HTTP Server WebGate for Oracle Access Manager requires several steps.
- [Registering the Oracle HTTP Server WebGate with Oracle Access Manager](#)
You can register the WebGate agent with Oracle Access Manager by using the Oracle Access Manager Administration console.

About Webgate

WebGate is a web server plug-in that intercepts HTTP requests and forwards them to an existing Oracle Access Manager instance for authentication and authorization.

For Oracle Fusion Middleware 12c, the Oracle WebGate software is installed as part of the Oracle HTTP Server 12c software installation. See Registering and Managing OAM 11g Agents in *Administrator's Guide for Oracle Access Management*.

General Prerequisites for Configuring Oracle HTTP Server WebGate

Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.

For the most up-to-date information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

For WebGate certification matrix, click and open <http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/oam-webgates-2147084.html>, then click the *Certification Matrix for 12c Access Management WebGates* link to download the certification matrix spreadsheet.

 **Note:**

For production environments, it is highly recommended that you install Oracle Access Manager in its own environment and not on the machines that are hosting the enterprise deployment.

For more information about Oracle Access Manager, see the latest Oracle Identity and Access Management documentation, which you can find in the **Middleware** documentation on the [Oracle Help Center](#).

Configuring Oracle HTTP Server WebGate

Configuring Oracle HTTP Server WebGate for Oracle Access Manager requires several steps.

In the following examples:

- Replace `OHS_ORACLE_HOME` with the complete path to the Oracle home where you installed the Oracle HTTP Server software.
- Replace `OHS_Configuration_Staging_Directory` with the path to the following location in the Oracle HTTP Server domain home:

```
OHS_Configuration_Staging_Directory/config/fmwconfig/components/OHS/  
componentName
```

1. Change to `deployWebGate` directory in the Oracle HTTP Server.

```
(UNIX)OHS_ORACLE_HOME/webgate/ohs/tools/deployWebGate
```

```
(Windows)OHS_ORACLE_HOME\webgate\ohs\tools\deployWebGate
```

2. To create the WebGate Instance directory and enable WebGate logging on OHS Instance, enter the following command::

```
(UNIX) ./deployWebGateInstance.sh -w DOMAIN_HOME -oh OHS_ORACLE_HOME
```

```
(Windows) deployWebGateInstance.bat -w DOMAIN_HOME -oh OHS_ORACLE_HOME
```

3. Verify that a `webgate` directory and subdirectories was created by the `deployWebGateInstance` command:

For example, on UNIX:

```
ls -lart DOMAIN_HOME/webgate/  
total 6  
drwxr-x---+ 8 orcl oinstall 20 Oct  2 07:14 ..  
drwxr-xr-x+ 4 orcl oinstall  4 Oct  2 07:14 .  
drwxr-xr-x+ 3 orcl oinstall  3 Oct  2 07:14 tools  
drwxr-xr-x+ 3 orcl oinstall  4 Oct  2 07:14 config
```

4. To change set the `PATH` environment variable, enter the following command::

```
(UNIX) export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:OHS_ORACLE_HOME/lib
```

```
(Windows) set PATH=%PATH%;OHS_ORACLE_HOME\bin
```

5. Change to the `EditHttpConf` directory:

```
(UNIX) OHS_ORACLE_HOME/webgate/ohs/tools/EditHttpConf
```

```
(Windows) OHS_ORACLE_HOME\webgate\ohs\tools>EditHttpConf
```

6. Enter the following command from the `InstallTools` directory:

```
(UNIX) ./EditHttpConf -w DOMAIN_HOME [-oh OHS_ORACLE_HOME] [-o  
output_file_name] [-dcc custom_dcc_scripts/pages_location]
```

```
(Windows) EditHttpConf -w DOMAIN_HOME [-oh OHS_ORACLE_HOME] [-o  
output_file_name] [-dcc custom_dcc_scripts\pages_location]
```

This command does the following:

- Copies the `apache_webgate.template` file from the Oracle HTTP Server Oracle home to a new `webgate.conf` file in the Oracle HTTP Server configuration directory.
- Updates the `httpd.conf` file to add one line, so it includes the `webgate.conf`.
- Generates a WebGate configuration file. The default name of the file is `webgate.conf`, but you can use a custom name by using the `output_file` argument to the command.

If you want to customize Detached Credential Collector (DCC) scripts or pages, such as the `oamssso/logout.html`, `oamssso-bin/login.p`, or `logout.pll` scripts), then you can copy these scripts from the following location to the custom location identified by the `-dcc` parameter to `EditHttpConf` utility:

```
OHS_ORACLE_HOME/webgate/ohs/
```

Registering the Oracle HTTP Server WebGate with Oracle Access Manager

You can register the WebGate agent with Oracle Access Manager by using the Oracle Access Manager Administration console.

For more information about OAM registration, see *Registering an OAM Agent Using the Console* in *Administrator's Guide for Oracle Access Management*.

- [Locating and Preparing the RREG Tool](#)
- [Updating the Standard Properties in the OAM11gRequest.xml File](#)
- [Running the RREG Tool](#)
- [Files and Artifacts Generated by RREG](#)
- [Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location](#)
- [Restarting the Oracle HTTP Server Instance](#)

Locating and Preparing the RREG Tool

To set up the RREG tool, complete the following steps:

1. Sign in to one of the Oracle Access Manager hosts in the Application tier.
2. Change to the following directory in the Oracle Access Manager Oracle home:

```
OAM_ORACLE_HOME/oam/server/rreg/client
```

 **Note:**

The location is required only for the out-of-band mode.

In this example, `OAM_ORACLE_HOME` refers to the Oracle home on the system where the Oracle Access Manager software was installed.

 **Note:**

If the Oracle Enterprise Deployment Guide for IDM was used, `OAM_ORACLE_HOME` may be `/u01/oracle/products/access/iam`.

 **Note:**

If you do not have privileges or access to the Oracle Access Manager server, then you can use out-of-band mode to generate the required files and register the WebGate with Oracle Access Manager. See [About RREG In-Band and Out-of-Band Mode](#).

3. Decompress the `RREG.tar.gz` file to the required directory.
4. From the unzipped directory, open the `oamreg.sh` file and set the following environment variables in the file, as follows:
 - Set `OAM_REG_HOME` to the absolute path to the directory in which you extracted the contents of RREG archive.

Set `JAVA_HOME` to the absolute path of the directory in which a supported JDK is installed on your machine.

Updating the Standard Properties in the OAM11gRequest.xml File

Before you can register the Webgate agent with Oracle Access Manager, you must update some required properties in the `OAM11gRequest.xml` file.

 **Note:**

If you plan to use the default values for most of the parameters in the provided XML file, then you can use the shorter version (`OAM11gRequest_short.xml`, in which all non-listed fields will take a default value).

 **Note:**

In the primary server list, the default names are mentioned as `OAM_SERVER1` and `OAM_SERVER2` for OAM servers. Rename these names in the list if the server names are changed in your environment.

To perform this task:

1. If you are using in-band mode, then change directory to the following location in the directory:

```
OAM_ORACLE_HOME/oam/server/rreg/input
```

If you are using out-of-band mode, then change directory to the location where you unpacked the RREG archive.

2. Make a copy of the `OAM11GRequest.xml` file template.
3. Review the properties listed in the file, and then update your copy of the `OAM11GRequest.xml` file to make sure the properties reference the host names and other values specific to your environment.

OAM11gRequest.xml Property	Set to...
serverAddress	The host and the port of the Administration Server for the Oracle Access Manager domain.
agentName	Any custom name for the agent. Typically, you use a name that identifies the Fusion Middleware product you are configuring for single sign-on.
applicationDomain	A value that identifies the Web tier host and the FMW component you are configuring for single sign-on.
security	<p>The security mode of the Oracle Access Manager server, which can be open, simple, or certificate mode.</p> <p>For an enterprise deployment, Oracle recommends simple mode, unless additional requirements exist to implement custom security certificates for the encryption of authentication and authorization traffic.</p> <p>In most cases, avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted.</p> <p>For more information using certificate mode or about Oracle Access Manager supported security modes in general, see <i>Securing Communication Between OAM Servers and WebGates</i> in the <i>Administrator's Guide for Oracle Access Management</i>.</p>
cachePragmaHeader	private
cacheControlHeader	private
ipValidation	0
	<ipValidation>0</ipValidation>
ipValidationExceptions	<p>The IP address of the front-end load balancer. For example:</p> <pre><ipValidationExceptions> <ipAddress>130.35.165.42</ipAddress> </ipValidationExceptions></pre>
agentBaseUrl	The host and the port of the machine on which Oracle HTTP Server 12c WebGate is installed.

Running the RREG Tool

The following topics provide information about running the RREG tool to register your Oracle HTTP Server Webgate with Oracle Access Manager.

- [About RREG In-Band and Out-of-Band Mode](#)
- [Running the RREG Tool in In-Band Mode](#)
- [Running the RREG Tool in Out-Of-Band Mode](#)

About RREG In-Band and Out-of-Band Mode

You can run the RREG Tool in one of two modes: in-band and out-of-band.

Use **in-band** mode when you have the privileges to access the Oracle Access Manager server and run the RREG tool yourself from the Oracle Access Manager Oracle home. You can then copy the generated artifacts and files to the Web server configuration directory after you run the RREG Tool.

Use **out-of-band** mode if you do *not* have privileges or access to the Oracle Access Manager server. For example, in some organizations, only the Oracle Access Manager server administrators have privileges access the server directories and perform administration tasks on the server. In out-of-band mode, the process can work as follows:

1. The Oracle Access Manager server administrator provides you with a copy of the RREG archive file (RREG.tar.gz).
2. Untar the RREG.tar.gz file that was provided to you by the server administrator.

For example:

```
gunzip RREG.tar.gz
tar -xvf RREG.tar
```

After you unpack the RREG archive, you can find the tool for registering the agent in the following location:

```
RREG_HOME/bin/oamreg.sh
```

In this example, *RREG_Home* is the directory in which you extracted the contents of RREG archive.

3. Use the instructions in [Updating the Standard Properties in the OAM11gRequest.xml File](#) to update the `OAM11gRequest.xml` file, and send the completed `OAM11gRequest.xml` file to the Oracle Access Manager server administrator.
4. The Oracle Access Manager server administrator then uses the instructions in [Running the RREG Tool in Out-Of-Band Mode](#) to run the RREG Tool and generate the `AgentID_response.xml` file.
5. The Oracle Access Manager server administrator sends the `AgentID_response.xml` file to you.
6. Use the instructions in [Running the RREG Tool in Out-Of-Band Mode](#) to run the RREG Tool with the `AgentID_response.xml` file and generate the required artifacts and files on the client system.

Running the RREG Tool in In-Band Mode

To run the RREG Tool in in-band mode:

1. Change to the RREG home directory.

If you are using in-band mode, the RREG directory is inside the Oracle Access Manager Oracle home:

```
OAM_ORACLE_HOME/oam/server/rreg
```

If you are using out-of-band mode, then the RREG home directory is the location where you unpacked the RREG archive.

2. Change to the following directory:

- (UNIX) `RREG_HOME/bin`
- (Windows) `RREG_HOME\bin`

```
cd RREG_HOME/bin/
```

3. Set the permissions of the `oamreg.sh` command so that you can execute the file:

```
chmod +x oamreg.sh
```

4. Enter the following command:

```
./oamreg.sh inband RREG_HOME/input/OAM11GRequest_edg.xml
```

In this example:

- It is assumed that the edited `OAM11GRequest.xml` file is located in the `RREG_HOME/input` directory.
- The output from this command is saved to the following directory:

```
RREG_HOME/output/
```

The following example shows a sample RREG session:

```
Welcome to OAM Remote Registration Tool!
Parameters passed to the registration tool are:
Mode: inband
Filename: /u01/oracle/products/fmw/iam_home/oam/server/rreg/client/rreg/
input/OAM11GRequest_edg.xml
Enter admin username:weblogic_idm
Username: weblogic_iam
Enter admin password:
Do you want to enter a Webgate password?(y/n):
n
Do you want to import an URIs file?(y/n):
n

-----
Request summary:
OAM11G Agent Name:SOA12213_EDG_AGENT
Base URL: https://soa.example.com:443
URL String:null
Registering in Mode:inband
Your registration request is being sent to the Admin server at: http://
```

```
host1.example.com:7001
```

```
-----  
Jul 08, 2015 7:18:13 PM oracle.security.jps.util.JpsUtil disableAudit  
INFO: JpsUtil: isAuditDisabled set to true  
Jul 08, 2015 7:18:14 PM oracle.security.jps.util.JpsUtil disableAudit  
INFO: JpsUtil: isAuditDisabled set to true  
Inband registration process completed successfully! Output artifacts  
are created in the output folder.
```

Running the RREG Tool in Out-Of-Band Mode

To run the RREG Tool in out-of-band mode on the WEBHOST server, the administrator uses the following command:

```
RREG_HOME/bin/oamreg.sh outofband input/OAM11GRequest.xml
```

In this example:

- Replace *RREG_HOME* with the location where the RREG archive file was unpacked on the server.
- The edited *OAM11GRequest.xml* file is located in the *RREG_HOME/input* directory.
- The RREG Tool saves the output from this command (the *AgentID_response.xml* file) to the following directory:

```
RREG_HOME/output/
```

The Oracle Access Manager server administrator can then send the *AgentID_response.xml* to the user who provided the *OAM11GRequest.xml* file.

To run the RREG Tool in out-of-band mode on the Web server client machine, use the following command:

```
RREG_HOME/bin/oamreg.sh outofband input/AgentID_response.xml
```

In this example:

- Replace *RREG_HOME* with the location where you unpacked the RREG archive file on the client system.
- The *AgentID_response.xml* file, which was provided by the Oracle Access Manager server administrator, is located in the *RREG_HOME/input* directory.
- The RREG Tool saves the output from this command (the artifacts and files required to register the Webgate software) to the following directory on the client machine:

```
RREG_HOME/output/
```

Files and Artifacts Generated by RREG

The files that get generated by the RREG Tool vary, depending on the security level you are using for communications between the WebGate and the Oracle Access Manager server. For more information about the supported security levels, see *Securing Communication Between OAM Servers and WebGates* in *Administrator's Guide for Oracle Access Management*.

Note that in this topic any references to `RREG_HOME` should be replaced with the path to the directory where you ran the RREG tool. This is typically the following directory on the Oracle Access Manager server, or (if you are using out-of-band mode) the directory where you unpacked the RREG archive:

```
OAM_ORACLE_HOME/oam/server/rreg/client
```

The following table lists the artifacts that are always generated by the RREG Tool, regardless of the Oracle Access Manager security level.

File	Location
<code>cwallet.sso</code>	<code>RREG_HOME/output/Agent_ID/</code>
<code>ObAccessClient.xml</code>	<code>RREG_HOME/output/Agent_ID/</code>

The following table lists the additional files that are created if you are using the SIMPLE or CERT security level for Oracle Access Manager:

File	Location
<code>aaa_key.pem</code>	<code>RREG_HOME/output/Agent_ID/</code>
<code>aaa_cert.pem</code>	<code>RREG_HOME/output/Agent_ID/</code>
<code>password.xml</code>	<code>RREG_HOME/output/Agent_ID/</code>

Note that the `password.xml` file contains the obfuscated global passphrase to encrypt the private key used in SSL. This passphrase can be different than the passphrase used on the server.

You can use the files generated by RREG to generate a certificate request and get it signed by a third-party Certification Authority. To install an existing certificate, you must use the existing `aaa_cert.pem` and `aaa_chain.pem` files along with `password.xml` and `aaa_key.pem`.

Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location

After the RREG Tool generates the required artifacts, manually copy the artifacts from the `RREG_Home/output/agent_ID` directory to the Oracle HTTP Server configuration directory on the Web tier host.

The location of the files in the Oracle HTTP Server configuration directory depends upon the Oracle Access Manager security mode setting (OPEN, SIMPLE, or CERT).

The following table lists the required location of each generated artifact in the Oracle HTTP Server configuration directory, based on the security mode setting for Oracle Access Manager. In some cases, you might have to create the directories if they do not exist already. For example, the wallet directory might not exist in the configuration directory.


 **Note:**

For an enterprise deployment, Oracle recommends simple mode, unless additional requirements exist to implement custom security certificates for the encryption of authentication and authorization traffic. The information about using open or certification mode is provided here as a convenience.

Avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted.

For more information using certificate mode or about Oracle Access Manager supported security modes in general, see *Securing Communication Between OAM Servers and WebGates* in *Administrator's Guide for Oracle Access Management*.

File	Location When Using OPEN Mode	Location When Using SIMPLE Mode	Location When Using CERT Mode
wallet/cwallet.sso	<i>OHS_CONFIG_DIR</i> / webgate/config/wallet	<i>OHS_CONFIG_DIR</i> / webgate/config/ wallet/	<i>OHS_CONFIG_DIR</i> / webgate/config/ wallet/

 **N**
o
t
e
:
 B
 y
 d
 e
 f
 a
 u
 l
 t
 t
 h
 e
 w
 a
 l
 l
 e
 t
 f
 o
 l
 d
 e
 r
 i
 s
 n
 o
 t
 a
 v
 a
 i
 l
 a
 b
 l
 e
 .
C

File	Location When Using OPEN Mode	Location When Using SIMPLE Mode	Location When Using CERT Mode
			r e a t e t h e w a l l e t f o l d e r u n d e r O H S - C O N F I G - D I R / w e b g a t e / c o n f i

File	Location When Using OPEN Mode	Location When Using SIMPLE Mode	Location When Using CERT Mode
			g / .
ObAccessClient.xml	<i>OHS_CONFIG_DIR</i> / webgate/config	<i>OHS_CONFIG_DIR</i> / webgate/config/	<i>OHS_CONFIG_DIR</i> / webgate/config/
password.xml	N/A	<i>OHS_CONFIG_DIR</i> / webgate/config/	<i>OHS_CONFIG_DIR</i> / webgate/config/
aaa_key.pem	N/A	<i>OHS_CONFIG_DIR</i> / webgate/config/ simple/	<i>OHS_CONFIG_DIR</i> / webgate/config/
aaa_cert.pem	N/A	<i>OHS_CONFIG_DIR</i> / webgate/config/ simple/	<i>OHS_CONFIG_DIR</i> / webgate/config/

 **Note:**

If you need to redeploy the `ObAccessClient.xml` to `WEBHOST1` and `WEBHOST2`, delete the cached copy of `ObAccessClient.xml` from the servers. The cache location on `WEBHOST1` is:

```
OHS_DOMAIN_HOME/servers/ohs1/cache/
```

And you must perform the similar step for the second Oracle HTTP Server instance on `WEBHOST2`:

```
OHS_DOMAIN_HOME/servers/ohs2/cache/
```

Restarting the Oracle HTTP Server Instance

For information about restarting the Oracle HTTP Server instance, see *Restarting Oracle HTTP Server Instances by Using WLST* in *Administering Oracle HTTP Server*.

If you have configured Oracle HTTP Server in a WebLogic Server domain, you can also use Oracle Fusion Middleware Control to restart the Oracle HTTP Server instances. For more information, see *Restarting Oracle HTTP Server Instances by Using Fusion Middleware Control* in *Administering Oracle HTTP Server*.

C

Updating the JDK After Installing and Configuring an Oracle Fusion Middleware Product

Consider that you have a JDK version `jdk1.8.0_121` installed on your machine. When you install and configure an Oracle Fusion Middleware product, the utilities, such as Configuration Wizard (`config.sh|exe`), OPatch, or RCU point to a default JDK, for example, `jdk1.8.0_121`. After some time, Oracle releases a new version of the JDK, say `jdk1.8.0_131` that carries security enhancements and bug fixes. From 12c (12.2.1.3.0) onwards, you can upgrade the existing JDK to a newer version, and can have the complete product stack point to the newer version of the JDK.

You can maintain multiple versions of JDK and switch to the required version on need basis.

- [About Updating the JDK Location After Installing an Oracle Fusion Middleware Product](#)
The binaries and other metadata and utility scripts in the Oracle home and Domain home, such as RCU or Configuration Wizard, use a JDK version that was used while installing the software and continue to refer to the same version of the JDK. The JDK path is stored in a variable called `JAVA_HOME` which is centrally located in `.globalEnv.properties` file inside the `ORACLE_HOME/oui` directory.

About Updating the JDK Location After Installing an Oracle Fusion Middleware Product

The binaries and other metadata and utility scripts in the Oracle home and Domain home, such as RCU or Configuration Wizard, use a JDK version that was used while installing the software and continue to refer to the same version of the JDK. The JDK path is stored in a variable called `JAVA_HOME` which is centrally located in `.globalEnv.properties` file inside the `ORACLE_HOME/oui` directory.

The utility scripts such as `config.sh|cmd`, `launch.sh`, or `opatch` reside in the `ORACLE_HOME`, and when you invoke them, they refer to the `JAVA_HOME` variable located in `.globalEnv.properties` file. To point these scripts and utilities to the newer version of JDK, you must update the value of the `JAVA_HOME` variable in the `.globalEnv.properties` file by following the directions listed in [Updating the JDK Location in an Existing Oracle Home](#).

To make the scripts and files in your Domain home directory point to the newer version of the JDK, you can follow one of the following approaches:

- Specify the path to the newer JDK on the Domain Mode and JDK screen while running the Configuration Wizard.

For example, consider that you installed Oracle Fusion Middleware Infrastructure with the JDK version `8u121`. So while configuring the WebLogic domain with the Configuration Assistant, you can select the path to the newer JDK on the Domain Mode and JDK screen of the Configuration Wizard. Example: `/scratch/jdk/jdk1.8.0_131`.

- Manually locate the files that have references to the JDK using `grep` (UNIX) or `findstr` (Windows) commands and update each reference. See [Updating the JDK Location in an Existing Domain Home](#).

**Note:**

If you install the newer version of the JDK in the same location as the existing JDK by overwriting the files, then you don't need to take any action.

- [Updating the JDK Location in an Existing Oracle Home](#)
The `getProperty.sh|cmd` script displays the value of a variable, such as `JAVA_HOME`, from the `.globalEnv.properties` file. The `setProperty.sh|cmd` script is used to set the value of variables, such as `OLD_JAVA_HOME` or `JAVA_HOME` that contain the locations of old and new JDKs in the `.globalEnv.properties` file.
- [Updating the JDK Location in an Existing Domain Home](#)
You must search the references to the current JDK, for example `jdk1.8.0_121` manually, and replace those instances with the location of the new JDK.

Updating the JDK Location in an Existing Oracle Home

The `getProperty.sh|cmd` script displays the value of a variable, such as `JAVA_HOME`, from the `.globalEnv.properties` file. The `setProperty.sh|cmd` script is used to set the value of variables, such as `OLD_JAVA_HOME` or `JAVA_HOME` that contain the locations of old and new JDKs in the `.globalEnv.properties` file.

The `getProperty.sh|cmd` and `setProperty.sh|cmd` scripts are located in the following location:

(UNIX) `ORACLE_HOME/oui/bin`

(Windows) `ORACLE_HOME\oui\bin`

Where, `ORACLE_HOME` is the directory that contains the products using the current version of the JDK, such as `jdk1.8.0_121`.

To update the JDK location in the `.globalEnv.properties` file:

- Use the `getProperty.sh|cmd` script to display the path of the current JDK from the `JAVA_HOME` variable. For example:

(UNIX) `ORACLE_HOME/oui/bin/getProperty.sh JAVA_HOME`

(Windows) `ORACLE_HOME\oui\bin\getProperty.cmd JAVA_HOME`

`echo JAVA_HOME`

Where `JAVA_HOME` is the variable in the `.globalEnv.properties` file that contains the location of the JDK.

- Back up the path of the current JDK to another variable such as `OLD_JAVA_HOME` in the `.globalEnv.properties` file by entering the following commands:

(UNIX) `ORACLE_HOME/oui/bin/setProperty.sh -name OLD_JAVA_HOME -value specify_the_path_of_current_JDK`

(Windows) `ORACLE_HOME\oui\bin\setProperty.cmd -name OLD_JAVA_HOME -value specify_the_path_of_current_JDK`

This command creates a new variable called `OLD_JAVA_HOME` in the `.globalEnv.properties` file, with a value that you have specified.

3. Set the new location of the JDK in the `JAVA_HOME` variable of the `.globalEnv.properties` file, by entering the following commands:

```
(UNIX) ORACLE_HOME/oui/bin/setProperty.sh -name JAVA_HOME -value  
specify_the_location_of_new_JDK
```

```
(Windows) ORACLE_HOME\oui\bin\setProperty.cmd -name JAVA_HOME -  
value specify_the_location_of_new_JDK
```

After you run this command, the `JAVA_HOME` variable in the `.globalEnv.properties` file now contains the path to the new JDK, such as `jdk1.8.0_131`.

Updating the JDK Location in an Existing Domain Home

You must search the references to the current JDK, for example `jdk1.8.0_121` manually, and replace those instances with the location of the new JDK.

You can use the `grep` (UNIX) or `findstr` (Windows) commands to search for the `jdk`-related references.

You'll likely be required to update the location of JDK in the following three files:

```
(UNIX) DOMAIN_HOME/bin/setNMJavaHome.sh
```

```
(Windows) DOMAIN_HOME\bin\setNMJavaHome.cmd
```

```
(UNIX) DOMAIN_HOME/nodemanager/nodemanager.properties
```

```
(Windows) DOMAIN_HOME\nodemanager\nodemanager.properties
```

```
(UNIX) DOMAIN_HOME/bin/setDomainEnv.sh
```

```
(Windows) DOMAIN_HOME\bin\setDomainEnv.cmd
```