

Oracle® Fusion Middleware

Upgrading Oracle HTTP Server



12c (12.2.1.4.0)

E95754-07

April 2023

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 Upgrading Oracle HTTP Server, 12c (12.2.1.4.0)

E95754-07

Copyright © 2014, 2023, 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, software documentation, data (as defined in the Federal Acquisition Regulation), 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," "commercial computer software documentation," or "limited rights data" 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	viii
Documentation Accessibility	viii
Diversity and Inclusion	viii
Related Documents	ix
Conventions	ix

What's New in This Guide

New and Changed Features for 12c	x
Other Significant Changes in this Document for 12c	x

1 Introduction to Upgrading Oracle HTTP Server to 12c (12.2.1.4.0)

About the Starting Points for an Oracle HTTP Server Upgrade	1-1
About the Oracle HTTP Server Standard Topologies	1-1
Fusion Middleware Infrastructure Upgrade Topology with Oracle HTTP Server	1-2
Oracle HTTP Server Standard Upgrade Topology for Standalone Agents not Associated with a WebLogic Domain	1-5
Differences between Oracle HTTP Server 11g and 12c	1-6
About Upgrade Restrictions	1-7
How to use this Guide	1-7
Determining whether Oracle HTTP Server is Standalone or Managed (Collocated)	1-8

2 Preparing to Upgrade Oracle HTTP Server

Oracle Fusion Middleware Pre-Upgrade Checklist	2-2
Creating a Complete Backup	2-4
Backing Up the Schema Version Registry Table	2-4
Maintaining Customized Domain and Environment Settings	2-4
Cloning Your Production Environment for Testing	2-5
Verifying Certification and System Requirements	2-6
Verify Your Environment Meets Certification Requirements	2-7

Verify System Requirements and Specifications	2-7
Migrating from a 32-Bit to a 64-Bit Operating System	2-8
Verify That the Database Hosting Oracle Fusion Middleware is Supported	2-11
Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware	2-11
Updating Policy Files when Using Enhanced Encryption (AES 256)	2-12
Purging Unused Data	2-12
Creating an Edition on the Server for Edition-Based Redefinition	2-12
Creating a Non-SYSDBA User to Run the Upgrade Assistant	2-13

Part I Upgrading Oracle HTTP Server from 11g to 12c

3 Upgrading a Standalone Oracle HTTP Server from 11g to 12c

About the Standalone Oracle HTTP Server Upgrade Process from 11g to 12c	3-1
Important Pre-Upgrade Considerations	3-3
Oracle Web Cache 11g Users	3-3
WebGate 11g Users	3-4
Updating the WebGate Log file-name	3-4
Application Artifacts from 11g	3-5
Migrating 11g Application Artifacts	3-5
Installing the Standalone Oracle HTTP Server	3-5
Stopping Servers and Processes	3-7
Upgrading Domain Component Configurations	3-7
Starting the Upgrade Assistant	3-8
Upgrade Assistant Parameters	3-8
Upgrading the 11g Domain Components Using the Upgrade Assistant	3-10
Verifying the Domain-Specific-Component Configurations Upgrade	3-13
Starting Servers and Processes	3-14
Starting the Standalone Oracle HTTP Server	3-14
Verifying the Upgrade	3-15
Post-Upgrade Task	3-15

4 Upgrading a Managed Oracle HTTP Server from 11g to 12c

About the Managed Oracle HTTP Server Upgrade Process from 11g to 12c	4-2
Installing the Product Distributions	4-4
Running a Pre-Upgrade Readiness Check	4-6
About Running a Pre-Upgrade Readiness Check	4-6
Starting the Upgrade Assistant in Readiness Mode	4-7
Upgrade Assistant Parameters	4-7
Performing a Readiness Check with the Upgrade Assistant	4-9

Understanding the Readiness Report	4-11
Stopping Servers and Processes	4-15
Creating the Required 12c Schemas with the RCU	4-16
About Reconfiguring the Domain	4-20
Backing Up the Domain	4-21
Starting the Reconfiguration Wizard	4-22
Reconfiguring the Domain with the Reconfiguration Wizard	4-22
Upgrading Domain Component Configurations	4-27
Starting the Upgrade Assistant	4-28
Upgrade Assistant Parameters	4-28
Upgrading Domain Components Using the Upgrade Assistant	4-30
Verifying the Domain-Specific-Component Configurations Upgrade	4-32
Starting Servers and Processes	4-33
Starting the Oracle HTTP Server	4-34
Verifying the Upgrade	4-35
Importing Wallets to the KSS Database after an Upgrade Using WLST	4-35
Post-Upgrade Task	4-35

Part II Upgrading Oracle HTTP Server from a Previous 12c Release

5 Upgrading a Standalone Oracle HTTP Server from a Previous 12c Release

About the Standalone Oracle HTTP Server Upgrade Process from a Previous 12c Release	5-2
Installing the Standalone Oracle HTTP Server	5-3
Stopping Servers and Processes	5-5
About Reconfiguring the Domain	5-6
Backing Up the Domain	5-7
Starting the Reconfiguration Wizard	5-8
Reconfiguring the Existing Domain with the Reconfiguration Wizard	5-8
Upgrading Standalone System Component Configurations	5-12
Starting the Upgrade Assistant	5-12
Upgrade Assistant Parameters	5-13
Upgrading Standalone System Component Configurations Using the Upgrade Assistant	5-14
Verifying the Domain-Specific-Component Configurations Upgrade	5-17
Verifying the Upgrade	5-17
Starting the Node Manager	5-18
Starting the Standalone Oracle HTTP Server	5-18
Starting Servers and Processes	5-18

6 Upgrading a Managed Oracle HTTP Server from a Previous 12c Release

About the Managed Oracle HTTP Server Upgrade Process from a Previous 12c Release	6-2
Installing the Product Distributions	6-4
Running a Pre-Upgrade Readiness Check	6-6
About Running a Pre-Upgrade Readiness Check	6-6
Starting the Upgrade Assistant in Readiness Mode	6-7
Upgrade Assistant Parameters	6-7
Performing a Readiness Check with the Upgrade Assistant	6-9
Understanding the Readiness Report	6-11
Stopping Servers and Processes	6-15
Upgrading Product Schemas (Applicable for Full-JRF Only)	6-16
Identifying Existing Schemas Available for Upgrade	6-17
Starting the Upgrade Assistant	6-18
Upgrade Assistant Parameters	6-19
Upgrading the Product Schemas Using the Upgrade Assistant	6-20
Verifying the Schema Upgrade	6-23
About Reconfiguring the Domain	6-24
Backing Up the Domain	6-25
Starting the Reconfiguration Wizard	6-25
Reconfiguring the Domain with the Reconfiguration Wizard	6-26
Upgrading Domain Component Configurations	6-31
Starting the Upgrade Assistant	6-31
Upgrade Assistant Parameters	6-32
Upgrading Domain Components Using the Upgrade Assistant	6-34
Verifying the Domain-Specific-Component Configurations Upgrade	6-36
Starting Servers and Processes	6-37
Starting the Oracle HTTP Server	6-38
Verifying the Upgrade	6-39
Post-Upgrade Task	6-39

A Replacing Certificate Signed Using MD5 Algorithm with Certificate Signed Using SHA-2 Algorithm

How to Check whether Certificate Signed with MD5 Algorithm is Present in the Wallet?	A-1
Removing Certificate Signed with MD5 Algorithm from the Wallet	A-3
Adding Certificate Signed with SHA-2 Algorithm to the Wallet	A-4

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

Updating the JDK Location in an Existing Oracle Home

B-1

C About the Reduced Downtime Upgrade Process

Performing a Reduced Downtime Upgrade

C-4

 Performing the Upgrade on Host 1

C-5

 Stopping Components, Servers and Processes on Host 1

C-6

 Uninstalling the Software

C-7

 Installing the Product Distributions

C-8

 Restarting Node Manager, Administration Server, Managed Servers and
 Components on Host 1

C-10

 Performing the Upgrade on Host 2

C-11

 Stopping the Components, Servers and Processes on Host 2

C-12

 Uninstalling the Software

C-13

 Installing the Software into the Existing Middleware Home

C-13

 Restarting the Managed Servers and Processes

C-13

 Validating the Upgrade

C-14

Preface

This preface contains the following sections:

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Documents](#)

Upgrade documentation is organized by tasks in the 12c documentation library. The task-specific pages provide direct links to common upgrade procedures and related documentation.

- [Conventions](#)

Audience

This manual is intended for Oracle Fusion Middleware system administrators who are responsible for upgrading Oracle Fusion Middleware. It is assumed that the readers of this manual have knowledge of the following:

- Oracle Fusion Middleware 11g or Oracle Fusion Middleware 12c system administration and configuration information for the existing deployment
- The configuration and expected behavior of the system or systems being upgraded

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://support.oracle.com/portal/> or visit [Oracle Accessibility Learning and Support](#) if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our

products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

Upgrade documentation is organized by tasks in the 12c documentation library. The task-specific pages provide direct links to common upgrade procedures and related documentation.

You can refer the Oracle Fusion Middleware Library for additional information.

- For installation information, see Fusion Middleware Installation Documentation.
- For upgrade information, see Fusion Middleware 12c Upgrade Documentation.
- For administration-related information, see Fusion Middleware 12c Administration Documentation.
- For release-related information, see Fusion Middleware 12c Release Notes.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<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.

What's New in This Guide

The following topics introduce the new and changed features of the standalone Oracle HTTP Server and provides pointers to additional information.

- [New and Changed Features for 12c](#)
- [Other Significant Changes in this Document for 12c](#)

New and Changed Features for 12c

Before you begin the upgrade process, review the following list of changes for standalone Oracle HTTP Servers in 12c:

- The installer no longer creates a separate instance directory for Oracle HTTP Server. Instance information for the standalone OHS is stored in a new **standalone domain** home. See [Oracle HTTP Server Standard Upgrade Topology for Standalone Agents not Associated with a WebLogic Domain](#).
- The OPMN agent and `opmnctl` command-line utilities are no longer used in Oracle Fusion Middleware. Instead, system components are managed by NodeManager and WLST commands.

Other Significant Changes in this Document for 12c

Support for Certificates signed with MD5 algorithm has been removed. See [Replacing Certificate Signed Using MD5 Algorithm with Certificate Signed Using SHA-2 Algorithm](#).

1

Introduction to Upgrading Oracle HTTP Server to 12c (12.2.1.4.0)

Before you begin, review all introductory information to understand the standard upgrade topologies and upgrade paths for Oracle HTTP Server 12c (12.2.1.4.0)

- [About the Starting Points for an Oracle HTTP Server Upgrade](#)
You can upgrade to Oracle HTTP Server 12c (12.2.1.4.0) from supported 11g and 12c releases.
- [About the Oracle HTTP Server Standard Topologies](#)
The steps to upgrade Oracle HTTP Server to 12c (12.2.1.4.0) depend on the existing production topology (either 11g or a previous 12c release).
- [Differences between Oracle HTTP Server 11g and 12c](#)
There are a few key differences to be aware of before you upgrade your Oracle HTTP Server to this release of Oracle Fusion Middleware.
- [About Upgrade Restrictions](#)
If you are using two or more Oracle Fusion Middleware products of the same or different versions in a single, supported, Oracle Fusion Middleware configuration, you must consider the interoperability and compatibility factors before planning the upgrade.
- [How to use this Guide](#)
The procedure to upgrade Oracle HTTP Server is determined by whether it's a standalone deployment or a managed deployment.
- [Determining whether Oracle HTTP Server is Standalone or Managed \(Collocated\)](#)

About the Starting Points for an Oracle HTTP Server Upgrade

You can upgrade to Oracle HTTP Server 12c (12.2.1.4.0) from supported 11g and 12c releases.

Supported release starting points are:

- Oracle HTTP Server 11g (11.1.1.9)
- Oracle HTTP Server 12c (12.1.3.0, 12.2.1.2 and 12.2.1.3)

The upgrade procedures in this guide explain how to upgrade an existing Oracle HTTP Server 11g domain and an existing 12c domain to Oracle Fusion Middleware 12c (12.2.1.4.0).

About the Oracle HTTP Server Standard Topologies

The steps to upgrade Oracle HTTP Server to 12c (12.2.1.4.0) depend on the existing production topology (either 11g or a previous 12c release).

As a result, it is difficult to provide exact upgrade instructions for every possible Oracle HTTP Server installation. Therefore, this upgrade documentation provides instructions for upgrading several typical Oracle HTTP Server topologies. These typical topologies are referred to as *standard upgrade topologies*.

Your actual topology may vary, but the topologies described here provide an example that can be used as a guide to upgrade other similar Oracle HTTP Server topologies.

 **Note:**

For additional information about the upgrade process and planning resources to ensure your upgrade is successful, see *Preparing to Upgrade in Planning an Upgrade of Oracle Fusion Middleware*.

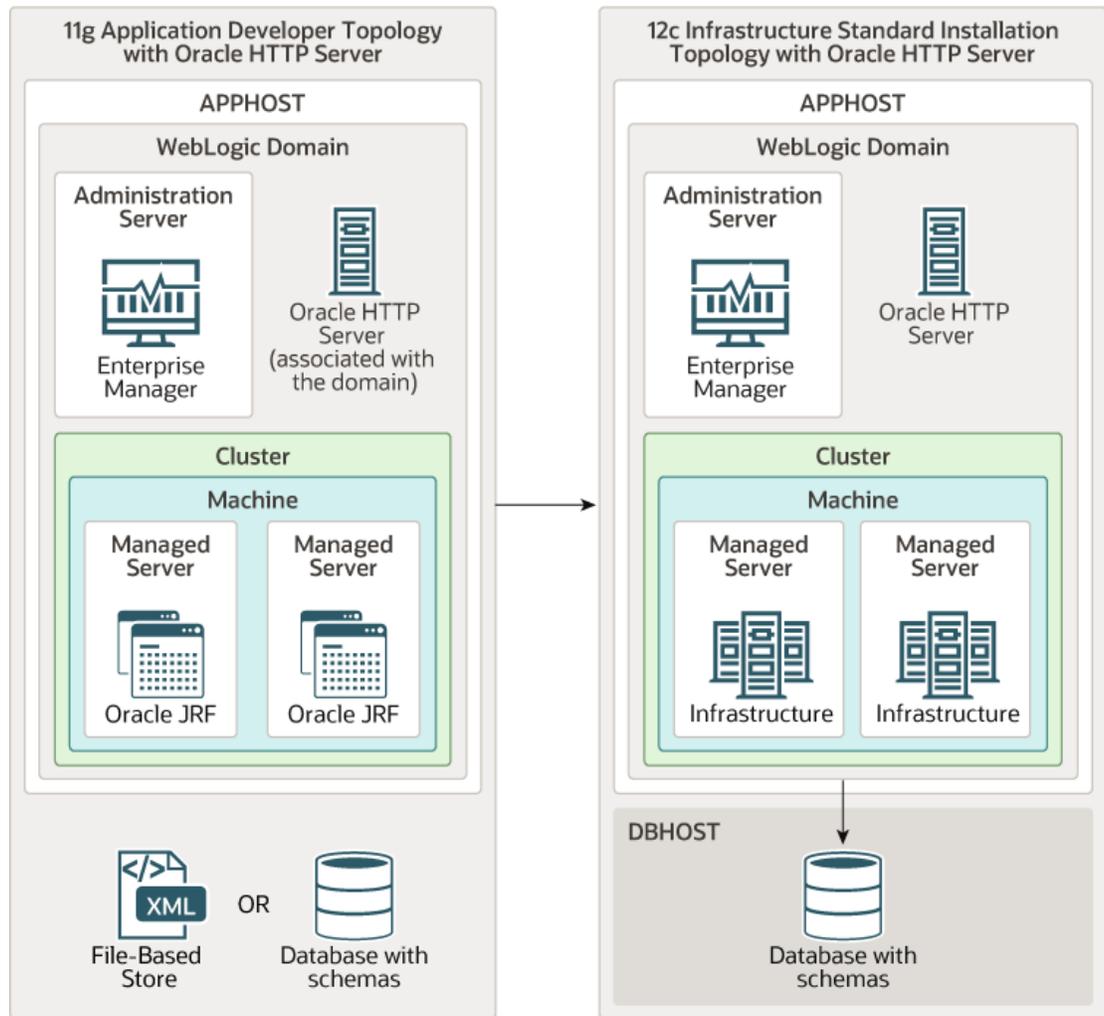
If you are upgrading from a previous 12c release, the standard topology remains unchanged. If you are upgrading from 11g, refer to the following sections to upgrade the following specific topologies:

- [Fusion Middleware Infrastructure Upgrade Topology with Oracle HTTP Server](#)
This topic contains the Oracle Fusion Middleware 11g Application Developer standard upgrade topology with Oracle HTTP Server and the resulting Oracle Fusion Middleware 12c Infrastructure topology as it appears after you complete the upgrade procedures in this guide.

Fusion Middleware Infrastructure Upgrade Topology with Oracle HTTP Server

This topic contains the Oracle Fusion Middleware 11g Application Developer standard upgrade topology with Oracle HTTP Server and the resulting Oracle Fusion Middleware 12c Infrastructure topology as it appears after you complete the upgrade procedures in this guide.

Figure 1-1 Infrastructure Standard Upgrade Topology with Oracle HTTP Server



Most of the elements in [Figure 1-1](#) are described in [Table 1-1](#).

Table 1-1 Description of the Elements in the Infrastructure Standard Upgrade Topology with Oracle HTTP Server

Element	Description and Links to Additional Documentation
11g Application Developer Topology with Oracle HTTP Server	<p>This is the label for the left side of Figure 1-1. It shows a typical single-host topology created using the Oracle Fusion Middleware 11g Application Developer installer. It consists of a single domain that contains a cluster of two managed servers and the Administration Server. It also has an optional file-based store or database with schemas.</p> <p>Figure 1-1 also shows an Oracle HTTP Server instance as part of the 11g domain.</p> <p>This document describes, step-by-step, how to upgrade this topology to an equivalent topology created by using the Oracle Fusion Middleware 12c Infrastructure distribution.</p>
12c Infrastructure Standard Installation Topology with Oracle HTTP Server	<p>This is the label for the right side of the figure. It shows a typical single-host topology created by using the Oracle Fusion Middleware 12c Infrastructure distribution.</p> <p>Like the Application Developer 11g topology, it also consists of a single domain that contains a cluster of two managed servers and the Administration Server.</p> <p>Figure 1-1 also shows an Oracle HTTP Server instance as part of the 12c domain.</p>
Oracle HTTP Server <i>associated with the domain</i>	<p>An Oracle HTTP Server 11g instance that is configured to be <i>associated with</i> the Oracle WebLogic Server domain. In Oracle Fusion Middleware 11g, system component instance, such as Oracle HTTP Server, are configured with an Oracle Universal Installer-based configuration wizard and are managed using Oracle Process Manager and Notification Server.</p>
Oracle HTTP Server	<p>Unlike the Oracle HTTP Server 11g instance in the left side of the diagram, the Oracle HTTP Server 12c instance shown in the 12c topology is configured as part of the domain using the Oracle Fusion Middleware Configuration Wizard. It is managed using Oracle Enterprise Manager Fusion Middleware Control, the Oracle WebLogic Scripting Tool (WLST), and the Oracle WebLogic Server Node Manager software.</p>

- [Oracle HTTP Server Standard Upgrade Topology for Standalone Agents not Associated with a WebLogic Domain](#)

In 12c, a **Standalone Oracle HTTP Server** is not managed by or associated with an Oracle WebLogic Server domain. A standalone Oracle HTTP Server 12c topology is installed and configured without the Oracle Fusion Middleware Infrastructure. A **Managed Oracle HTTP Server**, however, is associated with an existing Oracle WebLogic Server domain. For the standalone scenario, you install the Oracle HTTP Server software in its own Oracle home, and you configure the Oracle HTTP Server instance in its own **standalone domain**.

Oracle HTTP Server Standard Upgrade Topology for Standalone Agents not Associated with a WebLogic Domain

In 12c, a **Standalone Oracle HTTP Server** is not managed by or associated with an Oracle WebLogic Server domain. A standalone Oracle HTTP Server 12c topology is installed and configured without the Oracle Fusion Middleware Infrastructure. A **Managed Oracle HTTP Server**, however, is associated with an existing Oracle WebLogic Server domain. For the standalone scenario, you install the Oracle HTTP Server software in its own Oracle home, and you configure the Oracle HTTP Server instance in its own **standalone domain**.

 **Note:**

- For more information on installing and configuring a standalone Oracle HTTP Server, see About Oracle HTTP Server Installation in *Installing and Configuring Oracle HTTP Server*.
- For more information on the latest 12c standalone domain, see What Is a Standalone Domain? in *Understanding Oracle Fusion Middleware*.
- For more information on the administration scenarios and key features of the Oracle HTTP Server, see Introduction to Oracle HTTP Server *Administrator's Guide for Oracle HTTP Server*.
- For more information on upgrading a managed Oracle HTTP Server, see Performing the Infrastructure Upgrade in *Upgrading to the Oracle Fusion Middleware Infrastructure*.

Figure 1-2 shows a standalone Oracle HTTP Server topology.

Figure 1-2 Standalone Oracle HTTP Server Upgrade Topology

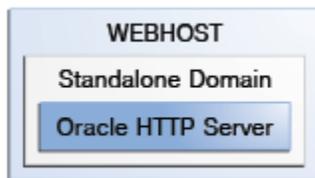


Table 1-2 describe the elements of this topology

Table 1-2 Description of the Elements in the Oracle Fusion Middleware Standalone Oracle HTTP Server Upgrade Topology

Element	Description and Links to Additional Documentation
WEBHOST	Standard term used in Oracle documentation that refers to the computer that hosts the Web tier.

Table 1-2 (Cont.) Description of the Elements in the Oracle Fusion Middleware Standalone Oracle HTTP Server Upgrade Topology

Element	Description and Links to Additional Documentation
Standalone Domain	<p>A standalone domain is only created if you are upgrading from 11g to 12c. If you are upgrading from 12.1.2.0.0, 12.1.3.0.0, 12.2.1.0.0, 12.2.1.1.0, 12.2.1.2.0, and 12.2.1.3.0, then the standalone domain already exists and there is no need to create the standalone domain.</p> <p>The standalone domain has a directory structure similar to an Oracle WebLogic domain, but it does not contain an Administration Server or Managed Servers. The Oracle WebLogic Server Node Manager and other tools allow you to manage the standalone Oracle HTTP Server instance.</p> <p>See <i>What Is a Standalone Domain?</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>

Differences between Oracle HTTP Server 11g and 12c

There are a few key differences to be aware of before you upgrade your Oracle HTTP Server to this release of Oracle Fusion Middleware.

[Table 1-3](#) lists the key differences between Oracle HTTP Server 11g and 12c.

Table 1-3 Differences between Oracle HTTP Server 11g and 12c

In Oracle HTTP Server 11g:	In Oracle HTTP Server 12c:
<p>Oracle HTTP Server instances are typically configured in a separate Oracle instance directory outside the 11g Middleware home.</p> <p>Oracle HTTP Server instances are managed using the Oracle Process Manager and the Notification Server (OPMN) management software. Optionally, the Oracle HTTP Server instances can be associated with the WebLogic domain.</p> <p><code>mod_ossso</code> is supported and included with Oracle HTTP Server 11g. If you use <code>mod_ossso</code> in 11g, it will be disabled after you upgrade, as it is not supported in 12c.</p>	<p>Oracle HTTP Server instances can be configured as part of an Oracle WebLogic Server domain, using the Oracle Fusion Middleware Configuration Wizard.</p> <p>When configured as part of an Oracle Application Server Infrastructure domain, Oracle HTTP Server instances can be managed using Oracle Enterprise Manager Fusion Middleware Control and the Oracle WebLogic Scripting Tool (WLST).</p> <p>In Oracle Fusion Middleware 12c, the Node Manager agent is responsible for delegating and executing management requests to Oracle HTTP Server instances.</p> <p><code>mod_ossso</code> is not supported or included with Oracle HTTP Server 12c.</p> <p>Oracle WebGate is the recommended replacement. You can install WebGate with Oracle HTTP Server. For more information on configuring WebGate with Oracle HTTP Server, see <i>Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server</i>.</p>

For more information about the changes to the way system components, such as Oracle HTTP Server, are configured and managed in Oracle Fusion Middleware 12c, as well as other key changes for Oracle Fusion Middleware 12c, see the following:

- [New and Changed Features for 12c](#)
- [New and Deprecated Terminology for 12c](#)
- [What is the WebLogic Management Framework?](#)

About Upgrade Restrictions

If you are using two or more Oracle Fusion Middleware products of the same or different versions in a single, supported, Oracle Fusion Middleware configuration, you must consider the interoperability and compatibility factors before planning the upgrade.

Interoperability

In the context of Oracle Fusion Middleware products, interoperability is defined as the ability of two Oracle Fusion Middleware products or components of the same version (or release) to work together (interoperate) in a supported Oracle Fusion Middleware configuration. Specifically, interoperability applies when the first 4 digits of the release or version number are the same. For example, Oracle Fusion Middleware 12c (12.2.1.0) components are generally interoperable with other 12c (12.2.1.4.0) components.

Compatibility

In the context of Oracle Fusion Middleware products, compatibility is defined as the ability of two Oracle Fusion Middleware components of different versions (or releases) to interoperate.

For a list of products and features available in Oracle Fusion Middleware Release 12.2.1.4.0, see [Products and Features Available in Oracle Fusion Middleware in *Understanding Interoperability and Compatibility*](#).

When performing the upgrade of your hardware or software, verify that your Oracle Fusion Middleware software is certified to support the new operating system or computer hardware. For more information, refer to the following resources:

- [Oracle Fusion Middleware Supported System Configurations](#)
- [Oracle® Fusion Middleware System Requirements and Specifications](#)

How to use this Guide

The procedure to upgrade Oracle HTTP Server is determined by whether it's a standalone deployment or a managed deployment.

Depending upon the type of your deployment, navigate to one of the following parts of this guide:

- [Upgrading Oracle HTTP Server from 11g to 12c](#)
 - [Upgrading a Standalone Oracle HTTP Server from 11g to 12c](#)
 - [Upgrading a Managed Oracle HTTP Server from 11g to 12c](#)
- [Upgrading Oracle HTTP Server from a Previous 12c Release](#)
 - [Upgrading a Standalone Oracle HTTP Server from a Previous 12c Release](#)
 - [Upgrading a Managed Oracle HTTP Server from a Previous 12c Release](#)

**Note:**

Before you start your upgrade, make sure that you review the introductory topics and complete the required pre-upgrade tasks.

Determining whether Oracle HTTP Server is Standalone or Managed (Collocated)

Oracle HTTP Server is the web server component for Oracle Fusion Middleware. It provides a listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the Web. If you configure Oracle HTTP Server in a WebLogic Server domain, it is the Managed Oracle HTTP Server because you can manage the Oracle HTTP Server instances like any other elements of the WebLogic Server domain using Enterprise Manager Fusion Middleware Control, or the WLST command-line interface, or the Node Manager. If you install the Oracle HTTP software in a separate Oracle home without installing the Oracle Fusion Middleware Infrastructure, it is the standalone mode.

To determine whether you are upgrading a managed or a standalone Oracle HTTP Server:

If you are an 11g user: Check for the `registered` property in the file `ORACLE_INSTANCE/config/OPMN/opmn/instance.properties`. If it is set to `true`, then the instance is registered. A managed Oracle HTTP Server is registered, if it is not registered then it is Standalone Oracle HTTP Server.

If you are a 12c user: Check the element `extention-template-ref` and its attribute `name` in the file `DOMAIN_HOME/init-info/domain-info.xml`. If you find an element with the name **Oracle HTTP Server (Standalone)**, then it is a standalone Oracle HTTP Server. And if you find an element with name **Oracle HTTP Server (Collocated)**, then it is collocated.

2

Preparing to Upgrade Oracle HTTP Server

Upgrade is performed while the servers are down. The pre-upgrade tasks are often time-consuming. Oracle recommends that you plan and prepare your environment for upgrade by completing these pre-upgrade tasks, so that you have a successful upgrade and a limited downtime.



Note:

If you are installing Oracle HTTP Server on your certified Windows operating system, you should verify that the `msvcr90.dll` and `linkinfo.dll` library files exist on your system; these are required by Oracle HTTP Server. See *Missing Libraries Might Cause HTTPD to Exit Without Notice* in *Release Notes for Oracle HTTP Server*. If you are unable to obtain the required libraries on your system, contact Microsoft support for further assistance.

Use the following checklist to make sure you complete the pre-upgrade tasks:

- [Oracle Fusion Middleware Pre-Upgrade Checklist](#)
Perform the tasks in this checklist before you begin any upgrade to ensure you have a successful upgrade and limited downtime.
- [Creating a Complete Backup](#)
Before you start an upgrade, back up all system-critical files, including the databases that host your Oracle Fusion Middleware schemas.
- [Cloning Your Production Environment for Testing](#)
Create a copy of your actual production environment, upgrade the cloned environment, verify that the upgraded components work as expected, and then (and only then) upgrade your production environment.
- [Verifying Certification and System Requirements](#)
Review the certification matrix and system requirements documents to verify that your environment meets the necessary requirements for installation.
- [Updating Policy Files when Using Enhanced Encryption \(AES 256\)](#)
If you plan to use enhanced encryption, such as Advanced Encryption Standard (AES 256), in your upgraded environment, Oracle recommends that you apply the latest required policy files to the JDK before you upgrade.
- [Purging Unused Data](#)
Purging unused data and maintaining a purging methodology before an upgrade can optimize the upgrade process.
- [Creating an Edition on the Server for Edition-Based Redefinition](#)
Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c (12.2.1.4.0).

- [Creating a Non-SYSDBA User to Run the Upgrade Assistant](#)
Oracle recommends that you create a non-SYSDBA user called `FMW` to run the Upgrade Assistant. This user has the privileges required to modify schemas, but does not have full administrator privileges.

Oracle Fusion Middleware Pre-Upgrade Checklist

Perform the tasks in this checklist before you begin any upgrade to ensure you have a successful upgrade and limited downtime.

Upgrades are performed while the servers are down. This checklist identifies important and often time-consuming pre-upgrade tasks that you can perform before the upgrade to limit your downtime. The more preparation you can do before you begin the upgrade process, the less time you will spend offline.



Note:

The pre-upgrade procedures you perform will depend on the configuration of your existing system, the components you are upgrading, and the environment you want to create at the end of the upgrade and configuration process. Complete only those tasks that apply to your configurations or use cases.

Table 2-1 Tasks to Perform Before You Upgrade to Oracle Fusion Middleware 12c

Task	Description
<p>Required Create a complete backup of your existing environment.</p>	<p>Back up all system-critical files and database(s) that contain any schemas that are to be upgraded. If the upgrade fails, you must restore your pre-upgrade environment and begin the upgrade again.</p> <p>See Creating a Complete Backup.</p> <ul style="list-style-type: none"> • Make sure that your backup includes the schema version registry table. See Backing Up the Schema Version Registry Table. • If you modified any of the startup scripts in your existing domain, you will need to copy them to temporary directory location (outside of the existing domain) during the upgrade and redeploy them after the upgrade. See Maintaining Customized Domain and Environment Settings.
<p>Optional Clone your production environment to use as an upgrade testing platform.</p>	<p>In addition to creating a complete backup of your system files, Oracle strongly recommends that you clone your production environment. This environment can be used to test the upgrade.</p> <p>See Cloning Your Production Environment for Testing.</p>

Table 2-1 (Cont.) Tasks to Perform Before You Upgrade to Oracle Fusion Middleware 12c

Task	Description
<p>Required Verify that you are installing and upgrading your product on a supported hardware and software configuration.</p> <div data-bbox="367 495 727 932" style="border-left: 2px solid #e6b800; border-right: 2px solid #e6b800; padding: 10px; margin: 10px 0;"> <p> Caution:</p> <p>Do not attempt an upgrade if you are unable to use the latest supported operating system. As with all supported configurations, failure to comply with these requirements may cause your upgrade to fail.</p> </div>	<p>Verify that your hardware and software configurations (including operating systems) are supported by the latest certifications and requirements documents. Also make sure to use a supported JDK version before you install the 12c product distributions.</p> <p>Oracle recommends that you verify this information right before you start the upgrade as the certification requirements are frequently updated.</p> <div data-bbox="943 613 1375 852" style="border-left: 2px solid #4f81bd; border-right: 2px solid #4f81bd; padding: 10px; margin: 10px 0;"> <p> Note:</p> <p>Make sure that you have applied the latest patches to your components before you upgrade.</p> </div> <p>See Verifying Certification and System Requirements.</p>
<p>Required for 32-bit Operating Systems Only Migrate to a 64-bit operating system before you can upgrade.</p>	<p>This is required only if you are currently running an unsupported 32-bit operating system.</p> <p>See Migrating from a 32-Bit to a 64-Bit Operating System.</p>
<p>Optional Update security policy files if you are using enhanced encryption (AES 256).</p>	<p>Some of the security algorithms used in Fusion Middleware 12c require additional policy files for the JDK.</p> <p>If you plan to use enhanced encryption, such as AES 256, Oracle recommends that you apply the latest required policy files to the JDK before you upgrade.</p> <p>See Updating Policy Files when Using Enhanced Encryption (AES 256).</p>
<p>Optional Purge any outdated or unused data before you upgrade.</p>	<p>To optimize performance, Oracle strongly recommends that you purge data and objects that will not be used in the upgraded environment.</p> <p>See Purging Unused Data.</p>
<p>Required for Oracle Database Users Only Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c (12.2.1.4.0).</p>	<p>If you are using an Edition-Based Redefinition (EBR) database, you must create the edition before starting the upgrade.</p> <p>See Creating an Edition on the Server for Edition-Based Redefinition.</p>
<p>Optional Create a Non-SYSDBA user to run the Upgrade Assistant.</p>	<p>Oracle recommends that you create the FMW user to run Upgrade Assistant. User FMW can run the Upgrade Assistant without system administration privileges.</p> <p>See Creating a Non-SYSDBA User to Run the Upgrade Assistant</p>

Creating a Complete Backup

Before you start an upgrade, back up all system-critical files, including the databases that host your Oracle Fusion Middleware schemas.

The backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table so that you can restore the contents back to its pre-upgrade state if the upgrade fails.

The Upgrade Assistant Prerequisites screen prompts you to acknowledge that backups have been performed before you proceed with the actual upgrade. However, note that the Upgrade Assistant does not verify that a backup has been created.

See:

- [Backing Up Your Environment in *Administering Oracle Fusion Middleware*](#)
- [Upgrading and Preparing Your Oracle Databases for 12c \(12.2.1.4.0\) in *Planning an Upgrade of Oracle Fusion Middleware*](#)
- [Backing Up the Schema Version Registry Table](#)
Your system backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table or the `FMWREGISTRY.SCHEMA_VERSION_REGISTRY$` table.
- [Maintaining Customized Domain and Environment Settings](#)
If you have modified any domain-generated, server startup scripts, or configuration files in your pre-upgrade environment, it is important to note that these changes are overwritten during the installation, domain upgrade, and reconfiguration operations. Save your customized files to a shared library location so that you can continue to use them after the upgrade.

Backing Up the Schema Version Registry Table

Your system backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table or the `FMWREGISTRY.SCHEMA_VERSION_REGISTRY$` table.

Each Fusion Middleware schema has a row in the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table. If you run the Upgrade Assistant to update an existing schema and it does not succeed, you must restore the original schema before you can try again. Before you run the Upgrade Assistant, make sure you back up your existing database schemas and the schema version registry.

Note:

Before you upgrade a schema using the Upgrade Assistant, you must perform a complete database backup. During the upgrade, you are required to acknowledge that backups have been performed.

Maintaining Customized Domain and Environment Settings

If you have modified any domain-generated, server startup scripts, or configuration files in your pre-upgrade environment, it is important to note that these changes are overwritten during the installation, domain upgrade, and reconfiguration operations.

Save your customized files to a shared library location so that you can continue to use them after the upgrade.

Every domain installation includes dynamically-generated domain and server startup scripts, such as `setDomainEnv`. These files are replaced by newer versions during the installation and upgrade process. To maintain your custom domain-level environment settings, Oracle recommends that you create a separate file to store the custom domain information before you upgrade, instead of modifying the scripts directly.

For example, if you want to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverrides.cmd` (Windows) or `setUserOverrides.sh` (UNIX) and configure it to add custom libraries to the WebLogic Server classpath, specify additional command-line options for running the servers, or specify additional environment variables. When using the `pack` and `unpack` commands, any custom settings that you add to this file are preserved during the domain upgrade operation and are carried over to the remote servers.

The following example illustrates startup customizations in a `setUserOverrides` file:

```
# add custom libraries to the WebLogic Server system classpath
if [ "${POST_CLASSPATH}" != "" ] ; then
    POST_CLASSPATH="${POST_CLASSPATH}${CLASSPATHSEP}${HOME}/foo/fooBar.jar"
    export POST_CLASSPATH
else
    POST_CLASSPATH="${HOME}/foo/fooBar.jar"
    export POST_CLASSPATH
fi

# specify additional java command-line options for servers
JAVA_OPTIONS="${JAVA_OPTIONS} -Dcustom.property.key=custom.value"
```

If the `setUserOverrides` file exists during a server startup, the file is included in the startup sequence and any overrides contained within this file take effect. You must store the `setUserOverrides` file in the `EXISTING_DOMAIN_HOME/bin` directory.

 **Note:**

If you are unable to create the `setUserOverrides` script before an upgrade, you need to reapply your settings as described in *Re-apply Customizations to Startup Scripts in Upgrading Oracle WebLogic Server*.

Cloning Your Production Environment for Testing

Create a copy of your actual production environment, upgrade the cloned environment, verify that the upgraded components work as expected, and then (and only then) upgrade your production environment.

Cloning your production environment for testing is recommended, but not required.

Upgrades cannot be reversed. In most cases, if an error occurs, you must stop the upgrade and restore the entire environment from backup and begin the upgrade process from the

beginning. Identifying potential upgrade issues in a development environment can eliminate unnecessary downtime.

 **Note:**

It is beyond the scope of this document to describe the cloning procedures for all components and operating systems. Cloning procedures are component and operating system-specific. At a high level, you install the pre-upgrade version of your component domain on a test machine, create the required schemas using the Repository Creation Utility (RCU), and perform the upgrade.

Additional benefits of running an upgrade in a cloned production environment include the following:

- Uncover and correct any upgrade issues.
- Practice completing an end-to-end upgrade.
- Understand the upgrade performance and how purge scripts can help.
- Understand the time required to complete the upgrade.
- Understand the database resource usage (such as temporary tablespace; PGA, and so on).

 **Note:**

You can run the pre-upgrade Readiness Check on the cloned production environment to help identify potential upgrade issues with your data, but you must perform a complete test upgrade on a cloned environment to ensure a successful upgrade.

Verifying Certification and System Requirements

Review the certification matrix and system requirements documents to verify that your environment meets the necessary requirements for installation.

 **Note:**

When checking the certification, system requirements, and interoperability information, be sure to check specifically for any 32-bit or 64-bit system requirements. It is important for you to download software specifically designed for the 32-bit or 64-bit environment, explicitly.

 **WARNING:**

Make sure that your current environment has been patched to the latest patch set *before* you begin the upgrade. Certifications are based on fully patched environments, unless stated otherwise.

- [Verify Your Environment Meets Certification Requirements](#)
Oracle has tested and verified the performance of your product on all certified systems and environments. Make sure that you are installing your product on a supported hardware or software configuration.
- [Verify System Requirements and Specifications](#)
It is important to verify that the system requirements such as disk space, available memory, specific platform packages and patches, and other operating system-specific items are met.
- [Verify That the Database Hosting Oracle Fusion Middleware is Supported](#)
You must have a supported Oracle database configured with the required schemas before you run Oracle Fusion Middleware 12c (12.2.1.4.0).
- [Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware](#)
At the time this document was published, the certified JDK for 12c (12.2.1.4.0) was 1.8.0_211.

Verify Your Environment Meets Certification Requirements

Oracle has tested and verified the performance of your product on all certified systems and environments. Make sure that you are installing your product on a supported hardware or software configuration.

Whenever new certifications occur, they are added to the appropriate certification document right away. New certifications can occur at any time, and for this reason the certification documents are kept outside of the documentation libraries and are available on Oracle Technology Network. See the Certification Matrix for 12c (12.2.1.4.0).

Verify System Requirements and Specifications

It is important to verify that the system requirements such as disk space, available memory, specific platform packages and patches, and other operating system-specific items are met.

Use the *Oracle Fusion Middleware System Requirements and Specifications* document to verify that the requirements of the certification are met. For example, if the Certification Matrix for 12c (12.2.1.4.0) indicates that your product is certified for installation on 64-Bit Oracle Linux 7, the System Requirements and Specifications document should be used to verify that your Oracle Linux 7 system has met the required minimum specifications such as disk space, available memory, specific platform packages and patches, and other operating system-specific items. This document is updated as needed and resides outside of the documentation libraries on the Oracle Technology Network (OTN).

 **Note:**

When you install the Oracle Fusion Middleware Release 12c software in preparation for upgrade, you should use the same user account that you used to install and configure the existing, pre-upgrade Oracle Fusion Middleware software. On UNIX operating systems, this ensures that the proper owner and group is applied to new Oracle Fusion Middleware 12c files and directories.

If you are running a 32-bit environment, you will need to perform an additional set of steps:

- [Migrating from a 32-Bit to a 64-Bit Operating System](#)
If you have a 32-bit operating system, then you must migrate your 32-bit environment to a 64-bit software environment before you upgrade.

Migrating from a 32-Bit to a 64-Bit Operating System

If you have a 32-bit operating system, then you must migrate your 32-bit environment to a 64-bit software environment before you upgrade.

Make sure to validate the migration to ensure all your Oracle Fusion Middleware 12c (12.2.1.3.0) software is working properly on the 64-bit machine, and only then perform the upgrade to Oracle Fusion Middleware 12c (12.2.1.4.0).

In these tasks, *host* refers to the 32-bit source machine and *target* refers to the new 64-bit target machine.

 **Note:**

These steps assume that your database is located on a separate host and will not be moved.

Upgrading an operating system typically involves the following:

 **Caution:**

These steps are provided as an example of the operating system upgrade process and may or may not include all of the procedures you must perform to update your specific operating system. Consult your operating system's upgrade documentation for more information.

- [Procure the Hardware That Supports the Upgrade's 64-bit Software Requirement](#)
Make sure that you have supported target hardware in place before you begin the upgrade process.
- [Stop All Processes](#)
Before upgrading, you must stop all processes, including Managed Servers, the Administration Server, and Node Manager, if they are started on the host.

- **Back Up All Files from the 32-bit Host Machine**
Make sure that you have created a complete backup of your entire 12c (12.2.1.3.0) deployment before you begin the upgrade process. These files can be used if there is an issue during the migration and you have to restart the process.
- **Set Up the Target 64-bit Machine with the 12c (12.2.1.3.0) Host Name and IP Address**
The host name and IP address of the target machine must be made identical to the host. This requires you to change the IP address and name of the source machine or decommission the source machine to avoid conflicts in the network.
- **Restore the 12c (12.2.1.3.0) Backup from 32-bit Host to 64-bit Host**
Restore the files you backed from the 32-bit host using the same directory structure that was used in 12c (12.2.1.3.0). The directory structure on the target machine must be identical to the structure of the host machine.
- **Install the 12c (12.2.1.4.0) Product Distributions on the Target Machine**
Oracle recommends an Out-of-Place approach for upgrade. Therefore, you must install the 12c (12.2.1.4.0) product distributions in a new Oracle home on the target machine.
- **Upgrade the Target 64-bit Environment Using the Standard Upgrade Procedure**
After installing the product on the target machine, you must upgrade each product component individually using an Upgrade Utility specified in the component-specific upgrade guide and complete any post-upgrade tasks.

Procure the Hardware That Supports the Upgrade's 64-bit Software Requirement

Make sure that you have supported target hardware in place before you begin the upgrade process.

Stop All Processes

Before upgrading, you must stop all processes, including Managed Servers, the Administration Server, and Node Manager, if they are started on the host.

Stop the Managed Servers

To stop a WebLogic Server Managed Server, use the `stopManagedWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopManagedWebLogic.sh managed_server_name admin_url`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopManagedWebLogic.cmd managed_server_name admin_url`

When prompted, enter your user name and password.

Stop the Administration Server

When you stop the Administration Server, you also stop the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To stop the Administration Server, use the `stopWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopWebLogic.sh`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Stop Node Manager

To stop Node Manager, close the command shell in which it is running.

Alternatively, after having set the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down. See `stopNodeManager` in *WLST Command Reference for WebLogic Server*.

Back Up All Files from the 32-bit Host Machine

Make sure that you have created a complete backup of your entire 12c (12.2.1.3.0) deployment before you begin the upgrade process. These files can be used if there is an issue during the migration and you have to restart the process.

Note:

If the upgrade from 32-bit to 64-bit takes place on the same machine, there is a risk of corrupting the source environment if the upgrade fails.

See *Backing Up Your Environment* in *Administering Oracle Fusion Middleware*.

During the upgrade you must have access to the contents of the following:

- `12c_DOMAIN_HOME`
- `12c/nodemanager` directory located in `12c_ORACLE_HOME/wlserver/common/`

Some of the backup and recovery procedures described in *Backing Up Your Environment* in *Administering Oracle Fusion Middleware* are product-specific. Do not proceed with the upgrade until you have a complete backup.

Set Up the Target 64-bit Machine with the 12c (12.2.1.3.0) Host Name and IP Address

The host name and IP address of the target machine must be made identical to the host. This requires you to change the IP address and name of the source machine or decommission the source machine to avoid conflicts in the network.

The process of changing an IP address and host name vary by operating system. Consult your operating system's administration documentation for more information.

Restore the 12c (12.2.1.3.0) Backup from 32-bit Host to 64-bit Host

Restore the files you backed from the 32-bit host using the same directory structure that was used in 12c (12.2.1.3.0). The directory structure on the target machine must be identical to the structure of the host machine.

See [Recovering Your Environment](#) in *Oracle Fusion Middleware Administrator's Guide*.

Install the 12c (12.2.1.4.0) Product Distributions on the Target Machine

Oracle recommends an Out-of-Place approach for upgrade. Therefore, you must install the 12c (12.2.1.4.0) product distributions in a new Oracle home on the target machine.

Refer to the component-specific installation guides for the component(s) you are installing.

Upgrade the Target 64-bit Environment Using the Standard Upgrade Procedure

After installing the product on the target machine, you must upgrade each product component individually using an Upgrade Utility specified in the component-specific upgrade guide and complete any post-upgrade tasks.

If you are upgrading additional components, see the component-specific upgrade guide.

Note:

The Node Manager upgrade procedure requires access to the original Node Manager files. Use the 12c (12.2.1.3.0) Node Manager files that you backed up from the 32-bit source machine as part of [Back Up All Files from the 32-bit Host Machine](#).

Verify That the Database Hosting Oracle Fusion Middleware is Supported

You must have a supported Oracle database configured with the required schemas before you run Oracle Fusion Middleware 12c (12.2.1.4.0).

Review the Fusion Middleware database requirements before starting the upgrade to ensure that the database hosting Oracle Fusion Middleware is supported and has sufficient space to perform an upgrade. See the Certification Matrix for 12c (12.2.1.4.0).

Note:

If your database version is no longer supported, you must upgrade to a supported version before starting an upgrade. See *Upgrading and Preparing Your Oracle Databases for 12c (12.2.1.4.0)* in *Planning an Upgrade of Oracle Fusion Middleware*.

Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware

At the time this document was published, the certified JDK for 12c (12.2.1.4.0) was 1.8.0_211.

Refer to the Oracle Fusion Middleware Supported System Configurations information on the Oracle Technology Network (OTN) to verify that the JDK you are using is supported.

If your JDK is not supported, or you do not have a JDK installed, you must download the required Java SE JDK, from the following website:

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

Make sure that the JDK is installed outside of the Oracle home. The Oracle Universal Installer validates that the designated Oracle home directory is empty, and the install does not progress until an empty directory is specified. If you install JDK under Oracle home, you may experience issues in future operations. Therefore, Oracle recommends that you use install the JDK in the following directory: `/home/oracle/products/jdk`.

For more information on the difference between generic and platform-specific installers, see *Understanding the Difference Between Generic and Platform-Specific Distributions* in the *Oracle Fusion Middleware Download, Installation, and Configuration Readme Files*.

Updating Policy Files when Using Enhanced Encryption (AES 256)

If you plan to use enhanced encryption, such as Advanced Encryption Standard (AES 256), in your upgraded environment, Oracle recommends that you apply the latest required policy files to the JDK before you upgrade.

The Java platform defines a set of APIs spanning major security areas, including cryptography, public key infrastructure, authentication, secure communication, and access control. These APIs allow developers to easily integrate security mechanisms into their application code.

Some of the security algorithms used in Fusion Middleware 12c (12.2.1.4.0) require additional policy files for the JDK. See [Java Cryptography Architecture Oracle Providers Documentation](#).

Note:

If you attempt to use enhanced encryption without applying these policy files to the JDK before you begin the upgrade, the upgrade can fail and you must restore the entire pre-upgrade environment and start the upgrade from the beginning.

Purging Unused Data

Purging unused data and maintaining a purging methodology before an upgrade can optimize the upgrade process.

Some components have automated purge scripts. If you are using purge scripts, wait until the purge is complete before starting the upgrade process. The upgrade may fail if the purge scripts are running while using the Upgrade Assistant to upgrade your schemas.

Creating an Edition on the Server for Edition-Based Redefinition

Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c (12.2.1.4.0).

Edition-based redefinition enables you to upgrade an application's database objects while the application is in use, thus minimizing or eliminating downtime. This is accomplished by changing (redefining) database objects in a private environment known as an edition. Only when all the changes have been made and tested, you make the new version of the application available to users.

**Note:**

This task must be completed by an Oracle Database User with DBA privileges.

Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c (12.2.1.4.0). The new edition for 12c (12.2.1.3.0) must be a child of your existing 12c or 12c (12.2.1.4.0) edition.

To create an edition on the database server, sign in as an SYS user (or another Oracle user that has DBA privileges) and enter the following command:

```
create edition Oracle_FMW_12_2_1_4_0 as child of
Oracle_FMW_12_2_1_3_0;
```

where `Oracle_FMW_12_2_1_3_0` is an example of the edition name you specified in RCU 12.2.1.3 when the 12.2.1.3 schemas were created. Be sure to provide the actual name used when creating the edition.

The following message notifies you that the edition is created successfully:

```
Edition created.
```

During the upgrade, you are prompted to launch the Reconfiguration Wizard to reconfigure your existing domain. Before running the Reconfiguration Wizard, you must specify the database default edition. Use the following SQL command to manually set up the default edition name for the database, for example:

```
ALTER DATABASE DEFAULT EDITION = Oracle_FMW_12_2_1_4_0;
```

Creating a Non-SYSDBA User to Run the Upgrade Assistant

Oracle recommends that you create a non-SYSDBA user called `FMW` to run the Upgrade Assistant. This user has the privileges required to modify schemas, but does not have full administrator privileges.

SYSDBA is an administrative privilege that is required to perform high-level administrative operations such as creating, starting up, shutting down, backing up, or recovering the database. The SYSDBA system privilege is for a fully empowered database administrator. When you connect with the SYSDBA privilege, you connect with a default schema and not with the schema that is generally associated with your user name. For SYSDBA, this schema is SYS. Access to a default schema can be a very powerful privilege. For example, when you connect as user SYS, you have unlimited privileges on data dictionary tables. Therefore, Oracle recommends that you create a non-SYSDBA user to upgrade the schemas. The privileges listed below must be granted to user FMW before starting the Upgrade Assistant.

 **Notes:**

The non-SYSDBA user FMW is created solely for the purpose of running the Upgrade Assistant. After this step is complete, drop the FMW user. Note that privileges required for running the Upgrade Assistant may change from release to release.

By default, the `v$xatrans$` table does not exist. You must run the `XAVIEW.SQL` script to create this table before creating the user. Moreover, the `grant select` privilege on the `v$xatrans$` table is required only by Oracle Identity Governance. If you do not require Oracle Identity Governance for configuration, or if you do not have the `v$xatrans$` table, then remove the following line from the script:

```
grant select on v$xatrans$ to FMW with grant option;
```

In the example below, `password` is the password that you set for the FMW user. When granting privileges, make sure that you specify your actual password.

```
create user FMW identified by password;
grant dba to FMW;
grant execute on DBMS_LOB to FMW with grant option;
grant execute on DBMS_OUTPUT to FMW with grant option;
grant execute on DBMS_STATS to FMW with grant option;
grant execute on sys.dbms_aqadm to FMW with grant option;
grant execute on sys.dbms_aqin to FMW with grant option;
grant execute on sys.dbms_aqjms to FMW with grant option;
grant execute on sys.dbms_aq to FMW with grant option;
grant execute on utl_file to FMW with grant option;
grant execute on dbms_lock to FMW with grant option;
grant select on sys.V_$INSTANCE to FMW with grant option;
grant select on sys.GV_$INSTANCE to FMW with grant option;
grant select on sys.V_$SESSION to FMW with grant option;
grant select on sys.GV_$SESSION to FMW with grant option;
grant select on dba_scheduler_jobs to FMW with grant option;
grant select on dba_scheduler_job_run_details to FMW with grant option;
grant select on dba_scheduler_running_jobs to FMW with grant option;
grant select on dba_aq_agents to FMW with grant option;
grant execute on sys.DBMS_SHARED_POOL to FMW with grant option;
grant select on dba_2pc_pending to FMW with grant option;
grant select on dba_pending_transactions to FMW with grant option;
grant execute on DBMS_FLASHBACK to FMW with grant option;
grant execute on dbms_crypto to FMW with grant option;
grant execute on DBMS_REPUTIL to FMW with grant option;
grant execute on dbms_job to FMW with grant option;
grant select on pending_trans$ to FMW with grant option;
grant select on dba_scheduler_job_classes to FMW with grant option;
grant select on sys.DBA_TABLESPACE_USAGE_METRICS to FMW with grant option;
grant select on SYS.DBA_DATA_FILES to FMW with grant option;
grant select on SYS.V_$ASM_DISKGROUP to FMW with grant option;
grant select on v$xatrans$ to FMW with grant option;
```

```

grant execute on sys.dbms_system to FMW with grant option;
grant execute on DBMS_SCHEDULER to FMW with grant option;
grant select on dba_data_files to FMW with grant option;
grant execute on UTL_RAW to FMW with grant option;
grant execute on DBMS_XMLDOM to FMW with grant option;
grant execute on DBMS_APPLICATION_INFO to FMW with grant option;
grant execute on DBMS_UTILITY to FMW with grant option;
grant execute on DBMS_SESSION to FMW with grant option;
grant execute on DBMS_METADATA to FMW with grant option;
grant execute on DBMS_XMLGEN to FMW with grant option;
grant execute on DBMS_DATAPUMP to FMW with grant option;
grant execute on DBMS_MVIEW to FMW with grant option;
grant select on ALL_ENCRYPTED_COLUMNS to FMW with grant option;
grant select on dba_queue_subscribers to FMW with grant option;
grant execute on SYS.DBMS_ASSERT to FMW with grant option;
grant select on dba_subscr_registrations to FMW with grant option;
grant manage scheduler to FMW;

```

If you are upgrading Oracle Identity Manager (OIM) schema, ensure that the FMW user has the following additional privileges:

```

grant execute on SYS.DBMS_FLASHBACK to fmw with grant option;
grant execute on sys.DBMS_SHARED_POOL to fmw with grant option;
grant execute on SYS.DBMS_XMLGEN to FMW with grant option;
grant execute on SYS.DBMS_DB_VERSION to FMW with grant option;
grant execute on SYS.DBMS_SCHEDULER to FMW with grant option;
grant execute on SYS.DBMS_SQL to FMW with grant option;
grant execute on SYS.DBMS_UTILITY to FMW with grant option;
grant ctxapp to FMW with admin option;
grant execute on SYS.DBMS_FLASHBACK TO FMW with grant option;
grant create MATERIALIZED VIEW to FMW with admin option;
grant all on SCHEMA_VERSION_REGISTRY TO FMW with grant option;
grant create SYNONYM to FMW with admin option;
grant execute on CTXSYS.CTX_ADM to FMW with grant option;
grant execute on CTXSYS.CTX_CLS TO FMW with grant option;
grant execute on CTXSYS.CTX_DDL TO FMW with grant option;
grant execute on CTXSYS.CTX_DOC TO FMW with grant option;
grant execute on CTXSYS.CTX_OUTPUT TO FMW with grant option;
grant execute on CTXSYS.CTX_QUERY TO FMW with grant option;
grant execute on CTXSYS.CTX_REPORT TO FMW with grant option;
grant execute on CTXSYS.CTX_THES TO FMW with grant option;
grant execute on CTXSYS.CTX_ULEXER TO FMW with grant option;
grant create JOB to FMW with admin option;

```

Part I

Upgrading Oracle HTTP Server from 11g to 12c

Part I contains the following chapters:

- [Upgrading a Standalone Oracle HTTP Server from 11g to 12c](#)
- [Upgrading a Managed Oracle HTTP Server from 11g to 12c](#)
This chapter describes the procedure to upgrade a managed Oracle HTTP Server from 11g to 12c.

3

Upgrading a Standalone Oracle HTTP Server from 11g to 12c

This chapter describes the procedures to upgrade the 11g standalone Oracle HTTP Server to 12c; one that is not managed by or associated with an existing Oracle WebLogic Server (WLS) domain.

This chapter contains the following sections:

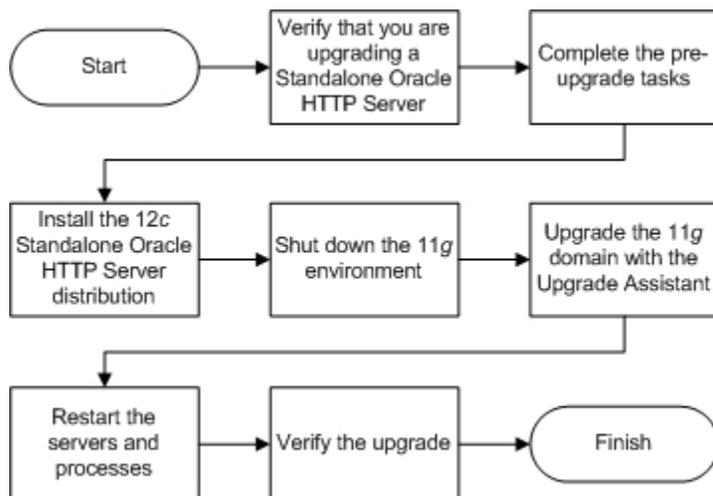
- [About the Standalone Oracle HTTP Server Upgrade Process from 11g to 12c](#)
Review the flowchart and roadmap for an overview of the upgrade process for Oracle HTTP Server.
- [Important Pre-Upgrade Considerations](#)
Before you begin the upgrade, it is important to make sure that your existing setup is not impacted during or after the upgrade. If you are using Oracle Web Cache or WebGate, or if you have Application-specific artifacts in your 11g domain, review the topics under this section carefully to prevent impact to your existing setup.
- [Installing the Standalone Oracle HTTP Server](#)
Before you begin your upgrade, download the Oracle HTTP Server 12c (12.2.1.4.0) distribution on the target system and install it by using the Oracle Universal Installer.
- [Stopping Servers and Processes](#)
A standalone Web Tier consists of Webcache and OHS as server components, which can be managed by the Oracle Process Manager and Notification (OPMN) server. Before you run the Upgrade Assistant to upgrade domain configurations, you must shut down the OPMN server.
- [Upgrading Domain Component Configurations](#)
After reconfiguring the domain, use the Upgrade Assistant to upgrade the domain *component* configurations inside the domain to match the updated domain configuration.
- [Starting Servers and Processes](#)
After a successful upgrade, start the Oracle Process Manager and Notification (OPMN) server and the Web Tier OHS server components.
- [Verifying the Upgrade](#)
You can verify that the upgrade is successful if you are able to start the Node Manager and the Standalone Oracle HTTP Server properly.
- [Post-Upgrade Task](#)
Perform the following update after upgrading the Oracle HTTP Server.

About the Standalone Oracle HTTP Server Upgrade Process from 11g to 12c

Review the flowchart and roadmap for an overview of the upgrade process for Oracle HTTP Server.

[Figure 3-1](#) shows the high-level procedures associated with a standalone Oracle HTTP Server upgrade when the starting point is 11g. The tools used for each step are also listed.

Figure 3-1 Upgrade Process Flowchart for Standalone Oracle HTTP Server from 11g to 12c



The following table describes the tasks that must be completed to upgrade a standalone Oracle HTTP Server from 11g to 12c.

Table 3-1 Tasks for Upgrading Standalone Oracle HTTP Server from 11g to 12c

Task	Description
Required Verify that you are upgrading a standalone Oracle HTTP Server.	To determine which Oracle HTTP Server you have in your existing environment, see Determining whether Oracle HTTP Server is Standalone or Managed (Collocated) .
Required Complete the pre-upgrade tasks.	The pre-upgrade tasks include cloning your production environment, verifying system requirements and certifications, purging unused data, and creating a non-SYSDBA user. For a complete list of pre-upgrade tasks, see Preparing to Upgrade Oracle HTTP Server . Additionally, also complete the tasks listed in: Important Pre-Upgrade Considerations .
Required Install the Standalone Oracle HTTP Server.	Run the installation program to install the software. Select the installation type Standalone Oracle HTTP Server (managed independently of WebLogic server) . This transfers the software to your system and creates a new Oracle home directory. See Installing the Standalone Oracle HTTP Server .
Required Shut down the 11g environment.	See Stopping Servers and Processes .
Required Upgrade the 11g standalone system component configurations.	See Upgrading Domain Component Configurations .
Required Restart the servers and processes.	See Starting Servers and Processes .

Table 3-1 (Cont.) Tasks for Upgrading Standalone Oracle HTTP Server from 11g to 12c

Task	Description
Required Verify the upgrade.	Your Oracle HTTP Server should continue to function as expected. If you have post-upgrade issues, you must troubleshoot the installation and retry the upgrade. See Troubleshooting Oracle HTTP Server in <i>Administering Oracle HTTP Server</i> .

Important Pre-Upgrade Considerations

Before you begin the upgrade, it is important to make sure that your existing setup is not impacted during or after the upgrade. If you are using Oracle Web Cache or WebGate, or if you have Application-specific artifacts in your 11g domain, review the topics under this section carefully to prevent impact to your existing setup.

- [Oracle Web Cache 11g Users](#)
Oracle Web Cache is a secure reverse proxy cache and a compression engine deployed between a browser and the HTTP server or a browser and the Content Management server to improve the performance of the websites by caching frequently accessed content. Oracle has released the last version of Web Cache in 11g. Web Cache is not available in 12c.
- [WebGate 11g Users](#)
A WebGate is a web-server plug-in for Oracle Access Manager (OAM) that intercepts HTTP requests and forwards them to the Access Server for authentication and authorization. WebGate is included as part of the Oracle HTTP Server 12c installation and is upgraded as part of the Oracle HTTP Server upgrade process through Upgrade Assistant.
- [Updating the WebGate Log file-name](#)
In Oracle Fusion Middleware 12c, the WebGate log file is renamed from `oblog.log` to `webgate.log`. After you upgrade to the latest release of Oracle HTTP Server, you must manually update the `oblog_config_wg.xml` file and replace every occurrence of `oblog.log` with `webgate.log`.
- [Application Artifacts from 11g](#)
Application artifacts include all your web resources: JSP files, images, stylesheets, Javascript, static HTML pages in addition to your Java classes and source files and web application configuration files. The integrated development environment (IDE) uses all these resources, and refers to them as web application artifacts.
- [Migrating 11g Application Artifacts](#)
You need to manually migrate any 11g application artifacts that reside within the Oracle instance, including any combination of static files such as HTML or images, CGI or FastCGI scripts or applications, or third-party modules. Application artifacts that were referred to by the 11g configuration, but were stored in directories outside of the Oracle instance, will continue to be referenced by the migrated configuration used by Oracle HTTP Server 12c.

Oracle Web Cache 11g Users

Oracle Web Cache is a secure reverse proxy cache and a compression engine deployed between a browser and the HTTP server or a browser and the Content Management server

to improve the performance of the websites by caching frequently accessed content. Oracle has released the last version of Web Cache in 11g. Web Cache is not available in 12c.

Consider the following limitations if you are using Web Cache in your 11g environment:

- Web Cache is not available in Fusion Middleware 12c. Correspondingly, there is no upgrade for Web Cache.
- Web Cache 11g front-ending a 12c Oracle HTTP Server is not a certified combination.
- If you are using both Web Cache and Oracle HTTP Server in your 11g setup, you can only upgrade the Oracle HTTP Server to 12c. In that case, you need to disable the 11g Web Cache and change the configuration settings to divert the traffic to Oracle HTTP Server directly.
- If you are using both Web Cache and Oracle HTTP Server in your 11g setup, and you have associated them with a WebLogic Server (WLS) domain for administering them through the Enterprise Manager Fusion Middleware Control, then you must also upgrade the WLS domain to 12c. In that case, the associated Oracle HTTP Server is also upgraded to 12c. However, the Web Cache is removed from the 12c Fusion Middleware Control.

WebGate 11g Users

A WebGate is a web-server plug-in for Oracle Access Manager (OAM) that intercepts HTTP requests and forwards them to the Access Server for authentication and authorization. WebGate is included as part of the Oracle HTTP Server 12c installation and is upgraded as part of the Oracle HTTP Server upgrade process through Upgrade Assistant.

Note:

In Oracle Fusion Middleware 12c (12.2.1.4.0), WebGate log file is renamed from `oblog.log` to `webgate.log`. After you upgrade to the latest release of Oracle HTTP Server, you must manually update the `oblog_config_wg.xml` file and replace every `oblog.log` occurrence with `webgate.log`. To complete this task, see [Updating the WebGate Log file-name](#).

Updating the WebGate Log file-name

In Oracle Fusion Middleware 12c, the WebGate log file is renamed from `oblog.log` to `webgate.log`. After you upgrade to the latest release of Oracle HTTP Server, you must manually update the `oblog_config_wg.xml` file and replace every occurrence of `oblog.log` with `webgate.log`.

To update the WebGate log file-name:

1. Change directory to the following path:
`DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_instance/webgate/config`
2. Open the `oblog_config_wg.xml` file in edit mode.

3. Replace every occurrence of `oblog.log` with `webgate.log`.

Application Artifacts from 11g

Application artifacts include all your web resources: JSP files, images, stylesheets, Javascript, static HTML pages in addition to your Java classes and source files and web application configuration files. The integrated development environment (IDE) uses all these resources, and refers to them as web application artifacts.

If you have 11g application artifacts that you want to continue using in 12c, carefully review the following:

- As part of upgrading Oracle HTTP Server from an 11g Oracle instance to a 12c domain, the Oracle HTTP Server configuration directory layout is being migrated from an Oracle instance to a standalone domain.
- Oracle HTTP Server 11g configuration files that reside in the component configuration directory of the Oracle instance are migrated automatically.
- Application artifacts that reside within the Oracle instance, including any combination of static files (such as HTML or images, CGI or FastCGI scripts or applications, or third-party modules), must be migrated manually after the upgrade to 12c.

See [Migrating 11g Application Artifacts](#).

Migrating 11g Application Artifacts

You need to manually migrate any 11g application artifacts that reside within the Oracle instance, including any combination of static files such as HTML or images, CGI or FastCGI scripts or applications, or third-party modules. Application artifacts that were referred to by the 11g configuration, but were stored in directories outside of the Oracle instance, will continue to be referenced by the migrated configuration used by Oracle HTTP Server 12c.

For example, if a third-party plug-in module was installed into the Oracle home with Oracle HTTP Server 11g, and the configuration referenced it by the Oracle home location using the configuration in the example below, the plug-in module must be installed manually into the Oracle home with 12c Oracle HTTP Server or the upgraded configuration must be modified to reference it elsewhere.

```
LoadModule example_module "ORACLE_HOME/ohs/modules/mod_example.so"
```

Similarly, if static files were copied into the `/htdocs` directory within the 11g component configuration directory, then they too must be manually copied into the 12c instance configuration directory or the upgraded configuration must be modified to reference it elsewhere. Other types of application artifacts must be manually migrated in a similar manner.

Installing the Standalone Oracle HTTP Server

Before you begin your upgrade, download the Oracle HTTP Server 12c (12.2.1.4.0) distribution on the target system and install it by using the Oracle Universal Installer.

To install the standalone Oracle HTTP Server:

1. Sign in to the target system.
2. Download the following from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:

Oracle HTTP Server (UNIX: `fmw_12.2.1.4.0_ohs_linux64.bin`), (Windows: `setup_fmw_12.2.1.4.0_ohs_win64.exe`)

3. Change to the directory where you downloaded the 12c (12.2.1.4.0) product distribution.
4. Check that your machine meet the following requirements:
 - Ensure that the system, patch, kernel, and other requirements are met as specified in the Roadmap for Verifying Your System Environment.
 - Because Oracle HTTP Server is installed by default on port 7777, you must ensure that port 7777 is not used by any service on the nodes. To check if this port is in use, run the following command before you install Oracle HTTP Server. You must free the port if it is in use.

```
netstat -an | grep 7777
```

5. Enter the following command to start the installation program:
(UNIX) `./fmw_12.2.1.4.0_ohs_linux64.bin`
(Windows) `setup_fmw_12.2.1.4.0_ohs_win64.exe`
6. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an 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, and click **Next**.

 **Note:**

The Installation Inventory Setup screen does not appear on Windows operating systems.

7. On the Welcome screen, review the information to make sure that you have met all the prerequisites. Click **Next**.
8. On the Auto Updates screen, select an option:
 - **Skip Auto Updates:** If you do not want your system to check for software updates at this time.
 - **Select patches from directory:** To navigate to a local directory if you downloaded patch files.
 - **Search My Oracle Support for Updates:** To automatically download software updates if you have a My Oracle Support account. You must enter Oracle Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

9. On the Installation Location screen, specify the location for the Oracle home directory and click **Next**.

As you are installing the standard installation topology for Oracle HTTP Server in a standalone domain, you can specify an Oracle home directory of your choice. However, ensure that you install the software in a new Oracle home.

See *Selecting Directories for Installation and Configuration* in *Planning an Installation of Oracle Fusion Middleware*.

10. On the Installation Type screen, select **Standalone HTTP Server (Managed independently of WebLogic server)** and click **Next**.
11. The Prerequisite Checks screen analyzes the host computer to ensure that the specific operating system prerequisites have been met.

To view the list of tasks that are 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).
12. On the Security Updates screen, indicate how you would like to receive security updates if you already have an Oracle Support account.

If you do not have an Oracle Support account and you are sure that you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
13. On the Installation Summary screen, verify the installation options that 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.
Click **Install** to begin the installation.
14. On the Installation Progress screen, when the progress bar displays 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.
15. The Installation Complete screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.

You have installed the Oracle HTTP Server in a standalone mode.

Stopping Servers and Processes

A standalone Web Tier consists of Webcache and OHS as server components, which can be managed by the Oracle Process Manager and Notification (OPMN) server. Before you run the Upgrade Assistant to upgrade domain configurations, you must shut down the OPMN server.

Stop all the OPMN components in the Oracle Instance that you are upgrading, using the `opmnctl` command.

For example:

```
$ORACLE_INSTANCE/bin/opmnctl stopall
```

Upgrading Domain Component Configurations

After reconfiguring the domain, use the Upgrade Assistant to upgrade the domain *component* configurations inside the domain to match the updated domain configuration.

- **Starting the Upgrade Assistant**
Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).

- [Upgrading the 11g Domain Components Using the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to upgrade component configurations in the WebLogic domain.
- [Verifying the Domain-Specific-Component Configurations Upgrade](#)
To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).

To start the Upgrade Assistant:



Note:

Before you start the Upgrade Assistant, make sure that the JVM character encoding is set to UTF-8 for the platform on which the Upgrade Assistant is running. If the character encoding is not set to UTF-8, then you will not be able to download files containing Unicode characters in their names. This can cause the upgrade to fail.

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant:
 - (UNIX) `./ua`
 - (Windows) `ua.bat`

For information about other parameters that you can specify on the command line, such as logging parameters, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 3-2 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.
<code>-response</code>	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
<code>-examine</code>	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.
<code>-logLevel attribute</code>	Optional	Sets the logging level, specifying one of the following attributes: <ul style="list-style-type: none"> TRACE NOTIFICATION WARNING ERROR INCIDENT_ERROR The default logging level is NOTIFICATION. Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.

Table 3-2 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logDir location</code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Upgrading the 11g Domain Components Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade component configurations in the WebLogic domain.

To upgrade the 11g domain using Upgrade Assistant:

1. On the Welcome screen, review an introduction to the Upgrade Assistant and information about important pre-upgrade tasks. Click **Next**.

Note:

For more information about any Upgrade Assistant screen, click **Help** on the screen.

2. On the Domain Directory screen, select **Create a New Domain (choose when upgrading from 11g)**.
3. On the Standalone Components screen, select **Standalone System Component Configurations**.

Select **Create a New Domain** and click **Next**.

 **Note:**

- **Create a New Domain** option should only be used if you are upgrading from version 11g.
- If you have already created a new 12c standalone domain for your upgraded 11g system components, you can extend the standalone domain with the standalone Oracle HTTP Server using **Update an Existing Domain** option.
- **Update an Existing Domain** option should only be used if you are upgrading from a previous 12c release (12.1.2.0.0, 12.1.3.0.0, 12.2.1.0.0, 12.2.1.1.0, 12.2.1.2.0, and 12.2.1.3.0) to the latest 12c release (12.2.1.4.0).

4. On the Component List screen, verify that the list includes all the components for which you want to upgrade configurations and click **Next**.

If you do not see the components you want to upgrade, click **Back** to go to the previous screen and specify a different domain.

5. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

 **Note:**

The Upgrade Assistant does not verify whether the prerequisites have been met.

6. On the Instance Directories screen, select **11g Source** and provide the directory location of the 11g instance that you want to upgrade.

This instance is used as a basis for configuring the 12c Oracle HTTP Server instance. The 11g Oracle HTTP Server installation is not altered.

Click **+** to add additional 11g instance directories, if necessary. Click **Next**.

 **Note:**

You cannot use the Upgrade Assistant to upgrade Oracle 10g instances to Oracle 12c. You must first upgrade Oracle 10g instances to 11g. For more information on migrating 10g to 11g, see the 11g upgrade documentation for your components.

7. On the Node Manager screen, specify the following to create a new Node Manager to administer your standalone domain.

Table 3-3 Field Descriptions for the Node Manager Screen

Option	Description
User Name	To specify the user name used to access Node Manager.

Table 3-3 (Cont.) Field Descriptions for the Node Manager Screen

Option	Description
Password	To specify the password used to access Node Manager. Re-enter the password for confirmation.
Listen Address	To enter the DNS name or IP address upon which Node Manager listens.
Listen Port	To enter the listening port number of Node Manager.

 **Note:**

The user name and password are only used to authenticate connections between Node Manager and clients. They are independent of the server Administrator credentials.

- On the Examine screen, review the status of the Upgrade Assistant as it examines each standalone component, verifying that the standalone component configuration is ready for upgrade. If the status is **Examine finished**, click **Next**.

If the examine phase fails, Oracle recommends that you cancel the upgrade by clicking **No** in the Examination Failure dialog. Click **View Log** to see what caused the error and refer to Troubleshooting Your Upgrade in *Upgrading with the Upgrade Assistant* for information on resolving common upgrade errors.

 **Note:**

- If you resolve any issues detected during the examine phase without proceeding with the upgrade, you can start the Upgrade Assistant again without restoring from backup. However, if you proceed by clicking **Yes** in the Examination Failure dialog box, you need to restore your pre-upgrade environment from backup before starting the Upgrade Assistant again.
- Canceling the examination process has no effect on the configuration data; the only consequence is that the information the Upgrade Assistant has collected must be collected again in a future upgrade session.

- On the Upgrade Summary screen, review the summary of the options you have selected for component configuration upgrade.

The response file collects and stores all the information that you have entered, and enables you to perform a silent upgrade at a later time. The silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file.

Click **Upgrade** to start the upgrade process.

- On the Upgrade Progress screen, monitor the status of the upgrade.

 **Caution:**

Allow the Upgrade Assistant enough time to perform the upgrade. Do not cancel the upgrade operation unless absolutely necessary. Doing so may result in an unstable environment.

If any components are not upgraded successfully, refer to the Upgrade Assistant log files for more information.

 **Note:**

The progress bar on this screen displays the progress of the current upgrade procedure. It does not indicate the time remaining for the upgrade.

Click **Next**.

11. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard. The Post-Upgrade Actions window describes the manual tasks you must perform to make components functional in the new installation. This window appears only if a component has post-upgrade steps.

If the upgrade fails: On the Upgrade Failure screen, click **View Log** to view and troubleshoot the errors. The logs are available at `ORACLE_HOME/oracle_common/upgrade/logs`. Note If the upgrade fails you must restore your pre-upgrade environment from backup, fix the issues, then restart the Upgrade Assistant.

Verifying the Domain-Specific-Component Configurations Upgrade

To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

To sign in to the Administration Console, go to: `http://
administration_server_host:administration_server_port/console`

To sign in to Oracle Enterprise Manager Fusion Middleware Control Console, go to: `http://
administration_server_host:administration_server_port/em`

 **Note:**

After upgrade, make sure you run the administration tools from the new 12c (12.2.1.4.0) Oracle home directory and not from the previous Oracle home directory.

During the upgrade process, some OWSM documents, including policy sets and predefined documents such as policies and assertion templates, may need to be upgraded. If a policy set or a predefined document is upgraded, its version number is incremented by 1.

If you created the FMW user to run the Upgrade Assistant, ensure that you delete the account after verifying your upgrade was successful.

Starting Servers and Processes

After a successful upgrade, start the Oracle Process Manager and Notification (OPMN) server and the Web Tier OHS server components.

 **Note:**

WebCache is not upgraded/supported in 12c.

You can use the `opmnctl` command to start the OPMN server and the Web Tier components after the upgrade.

For example:

```
$ORACLE_INSTANCE/bin/opmnctl startall
```

 **Note:**

If you changed to internal authentication before upgrading product schemas, change back to external authentication after starting the servers and processes.

- [Starting the Standalone Oracle HTTP Server](#)
After the upgrade, start the standalone Oracle HTTP Server.

Starting the Standalone Oracle HTTP Server

After the upgrade, start the standalone Oracle HTTP Server.

To start the standalone Oracle HTTP Server:

1. Go to the `DOMAIN_HOME/bin` directory:
(UNIX) `ORACLE_HOME/user_projects/domains/<domain name>/bin`

(WINDOWS) ORACLE_HOME\user_projects\domains*<domain name>*\bin

2. Enter the command to start the standalone server.

On UNIX operating systems:

```
./startComponent.sh ohs_name
```

On Windows operating systems:

```
startComponent.cmd ohs_name
```

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

Verifying the Upgrade

You can verify that the upgrade is successful if you are able to start the Node Manager and the Standalone Oracle HTTP Server properly.

If you experience post-upgrade issues, you need to troubleshoot the installation and retry the upgrade. See Troubleshooting Oracle HTTP Server in *Administrator's Guide for Oracle HTTP Server*.

If you are not able to start the newly upgraded environment, a possible cause could be the use of MD5 certificates in your Oracle wallet. To check whether you are using MD5 signatures and how to replace them with SHA-2 certificates, see [Replacing Certificate Signed Using MD5 Algorithm with Certificate Signed Using SHA-2 Algorithm](#) .

Post-Upgrade Task

Perform the following update after upgrading the Oracle HTTP Server.

Updating to 19c Database Client:

Install the new patches to update the OHS Database client and OSS patches to use 19c Database client libraries. For more information about upgrading the new Database client 19c for Oracle Fusion Middleware 12.2.1.4.0, see [Doc ID 2921245.1](#).

4

Upgrading a Managed Oracle HTTP Server from 11g to 12c

This chapter describes the procedure to upgrade a managed Oracle HTTP Server from 11g to 12c.

For more information about the supported versions, see [About the Starting Points for an Oracle HTTP Server Upgrade](#).

Complete the steps in the following topics to perform the upgrade:

- [About the Managed Oracle HTTP Server Upgrade Process from 11g to 12c](#)
Review the flowchart and roadmap for an overview of the upgrade process for Managed Oracle HTTP Server.
- [Installing the Product Distributions](#)
Before you begin your upgrade, download Oracle Fusion Middleware Infrastructure and Oracle HTTP Server 12c (12.2.1.4.0) distributions on the target system and install them using Oracle Universal Installer.
- [Running a Pre-Upgrade Readiness Check](#)
To identify potential issues with the upgrade, Oracle recommends that you run a readiness check before you start the upgrade process. Be aware that the readiness check may not be able to discover all potential issues with your upgrade. An upgrade may still fail, even if the readiness check reports success.
- [Stopping Servers and Processes](#)
Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.
- [Creating the Required 12c Schemas with the RCU](#)
When upgrading from 11g, you must create the required 12c schemas. You can use the Repository Creation Utility (RCU) to create customized schemas or, optionally, you can use the Upgrade Assistant to create schemas using the default schema settings. This procedure describes how to create schemas using the RCU. Information about using the Upgrade Assistant to create schemas is covered in the upgrade procedures.
- [About Reconfiguring the Domain](#)
Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1.4.0).
- [Upgrading Domain Component Configurations](#)
After reconfiguring the domain, use the Upgrade Assistant to upgrade the domain *component* configurations inside the domain to match the updated domain configuration.
- [Starting Servers and Processes](#)
After a successful upgrade, restart all processes and servers, including the Administration Server and any Managed Servers.
- [Verifying the Upgrade](#)
If you can log in to the Administration Console, the Enterprise Manager Fusion Middleware Control, and the OHS Home Page, then your upgrade is successful.

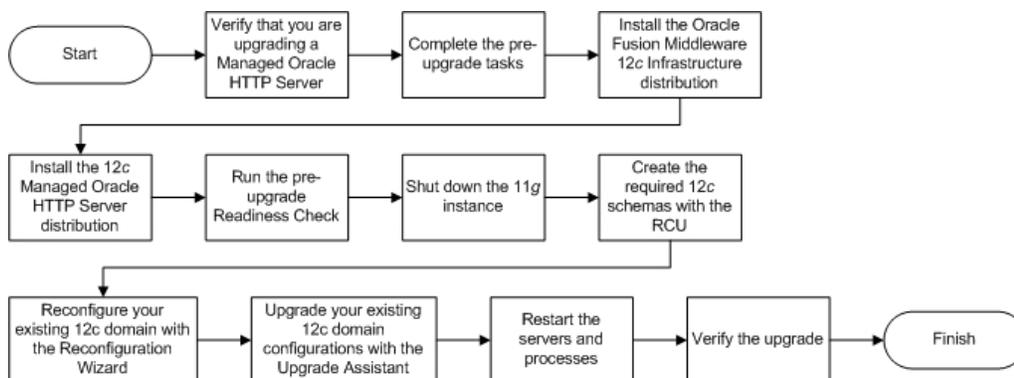
- [Importing Wallets to the KSS Database after an Upgrade Using WLST](#)
When you use Upgrade Assistant to upgrade from a previous version of Oracle HTTP Server to 12c (12.2.1.4.0), you must perform an additional wallet management task.
- [Post-Upgrade Task](#)
Perform the following update after upgrading the Oracle HTTP Server.

About the Managed Oracle HTTP Server Upgrade Process from 11g to 12c

Review the flowchart and roadmap for an overview of the upgrade process for Managed Oracle HTTP Server.

[Figure 4-1](#) shows the process flow to upgrade a Managed Oracle HTTP Server to the latest 12c release. The tools used for each step are also listed.

Figure 4-1 Upgrade Process Flowchart for Managed Oracle HTTP Server from 11g to 12c



[Table 6-1](#) describes the tasks that must be completed to upgrade a Managed Oracle HTTP Server from a 11g release to 12c (12.2.1.4.0).

Table 4-1 Tasks for Upgrading Managed Oracle HTTP Server from 11g to 12c

Task	Description
Required Verify that you are upgrading a Managed Oracle HTTP Server.	To determine which Oracle HTTP Server you have in your existing environment, see Determining whether Oracle HTTP Server is Standalone or Managed (Collocated) .
Required If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.	The pre-upgrade tasks include cloning your production environment, verifying system requirements and certifications, purging unused data, and creating a non-SYSDBA user. For a complete list of pre-upgrade tasks, see Preparing to Upgrade Oracle HTTP Server .

Table 4-1 (Cont.) Tasks for Upgrading Managed Oracle HTTP Server from 11g to 12c

Task	Description
<p>Required Download and install the 12.2.1.4.0 Fusion Middleware Infrastructure and Oracle HTTP Server distributions.</p>	<p>The Infrastructure distribution packs the WebLogic Server and the Java Required Files (JRF) that are required to set up the foundation to install other Fusion Middleware products.</p> <p>As per the upgrade topology defined in this guide, you must install the Infrastructure in a new Oracle home directory.</p> <p>You must install Oracle HTTP Server in the Oracle home directory that is created when you installed the 12.2.1.4.0 Infrastructure.</p> <p>See Installing the Product Distributions.</p>
<p>Optional Run the Readiness Check.</p>	<p>See Running a Pre-Upgrade Readiness Check.</p>
<p>Required Shut down the 11g instance.</p>	<p>Before you start the upgrade process, shut down the Administration Server, the Managed Servers, and your 11g instance.</p> <p>See Stopping Servers and Processes.</p>
<p>Required Create the required 12c schemas.</p>	<p>See Creating the Required 12c Schemas with the RCU.</p>
<p>Required Reconfigure your existing 12c domain with the Reconfiguration Wizard.</p>	<p>Run the Reconfiguration Wizard from the 12c Oracle HTTP Server installation to reconfigure the existing domain.</p> <p>See About Reconfiguring the Domain.</p>
<p>Required Upgrade your existing 12c domain configurations with the Upgrade Assistant.</p>	<p>After the installation, you need to use the Upgrade Assistant to upgrade the Oracle HTTP Server and system component infrastructure.</p> <p>See Upgrading Domain Component Configurations.</p>
<p>Required Restart the servers and processes.</p>	<p>The upgrade process is complete. You can now restart the Administration Server, the Managed Servers, and your 12.2.1.4.0 instance.</p> <p>See Starting Servers and Processes.</p>
<p>Required Verify the upgrade.</p>	<p>For verifying the upgrade, see Verifying the Upgrade.</p> <p>Complete the following post-upgrade task: Importing Wallets to the KSS Database after an Upgrade Using WLST.</p>

Installing the Product Distributions

Before you begin your upgrade, download Oracle Fusion Middleware Infrastructure and Oracle HTTP Server 12c (12.2.1.4.0) distributions on the target system and install them using Oracle Universal Installer.

Note:

When Infrastructure is required for the upgrade, you must install the Oracle Fusion Middleware distribution first before you install other Fusion Middleware products.

To install the 12c (12.2.1.4.0) distributions:

1. Sign in to the target system.
2. Download the following from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle Fusion Middleware Infrastructure (`fmw_12.2.1.4.0_infrastructure_generic.jar`)
 - Oracle HTTP Server (UNIX: `fmw_12.2.1.4.0_ohs_linux64.bin`), (Windows: `setup_fmw_12.2.1.4.0_ohs_win64.exe`)
3. Check that your machines meet the following requirements:
 - Ensure that the system, patch, kernel, and other requirements are met as specified in *Installing and Configuring Oracle HTTP Server*.
 - Because Oracle HTTP Server is installed by default on port 7777, you must ensure that port 7777 is not used by any service on the nodes. To check if this port is in use, run the following command before you install Oracle HTTP Server. You must free the port if it is in use.

```
netstat -an | grep 7777
```

4. On UNIX platforms, if the `/etc/oraInst.loc` file exists, check that its contents are correct. Specifically, check that the inventory directory is correct and that you have write permissions for that directory.

If the `/etc/oraInst.loc` file does not exist, you can skip this step.

5. Change to the directory where you downloaded the 12c (12.2.1.4.0) product distribution.
6. Start the installation program for Oracle Fusion Middleware Infrastructure:
 - (UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.4.0_infrastructure.jar`
 - (Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.4.0_infrastructure.jar`
7. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an 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, and click **Next**.

 **Note:**

The Installation Inventory Setup screen does not appear on Windows operating systems.

8. On the Welcome screen, review the information to make sure that you have met all the prerequisites. Click **Next**.
9. On the Auto Updates screen, select an option:
 - **Skip Auto Updates:** If you do not want your system to check for software updates at this time.
 - **Select patches from directory:** To navigate to a local directory if you downloaded patch files.
 - **Search My Oracle Support for Updates:** To automatically download software updates if you have a My Oracle Support account. You must enter Oracle Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

10. On the Installation Location screen, specify the location for the Oracle home directory and click **Next**.

Since you are installing the standard installation topology for a collocated Oracle HTTP Server in a WebLogic Server domain, enter the path to an existing Oracle Fusion Middleware Infrastructure Oracle home.

See Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.

11. On the Installation Type screen, select the following:
 - For Infrastructure, select **Fusion Middleware Infrastructure**
 - For Oracle HTTP Server, select **Collocated HTTP Server (Managed through WebLogic Server)**

Click **Next**.

12. The Prerequisite Checks screen analyzes the host computer to ensure that the specific operating system prerequisites have been met.

To view the list of tasks that are 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).

13. On the Installation Summary screen, verify the installation options that 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.

Click **Install** to begin the installation.

14. On the Installation Progress screen, when the progress bar displays 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.

15. The Installation Complete screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.
16. After you have installed Oracle Fusion Middleware Infrastructure, enter the following command to start the installer for your product distribution and repeat the steps above to navigate through the installer screens:

```
(UNIX) ./fmw_12.2.1.4.0_ohs_linux64.bin
```

```
(Windows) setup_fmw_12.2.1.4.0_ohs_win64.exe
```

You have installed the Oracle HTTP Server in a collocated mode.

Running a Pre-Upgrade Readiness Check

To identify potential issues with the upgrade, Oracle recommends that you run a readiness check before you start the upgrade process. Be aware that the readiness check may not be able to discover all potential issues with your upgrade. An upgrade may still fail, even if the readiness check reports success.

- [About Running a Pre-Upgrade Readiness Check](#)
You can run the Upgrade Assistant in `-readiness` mode to detect issues before you perform the actual upgrade. You can run the readiness check in GUI mode using the Upgrade Assistant or in silent mode using a response file.
- [Starting the Upgrade Assistant in Readiness Mode](#)
Use the `-readiness` parameter to start the Upgrade Assistant in readiness mode.
- [Performing a Readiness Check with the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to complete the pre-upgrade readiness check.
- [Understanding the Readiness Report](#)
After performing a readiness check for your domain, review the report to determine whether you need to take any action for a successful upgrade.

About Running a Pre-Upgrade Readiness Check

You can run the Upgrade Assistant in `-readiness` mode to detect issues before you perform the actual upgrade. You can run the readiness check in GUI mode using the Upgrade Assistant or in silent mode using a response file.

The Upgrade Assistant readiness check performs a read-only, pre-upgrade review of your Fusion Middleware schemas and WebLogic domain configurations that are at a supported starting point. The review is a read-only operation.

The readiness check generates a formatted, time-stamped readiness report so you can address potential issues before you attempt the actual upgrade. If no issues are detected, you can begin the upgrade process. Oracle recommends that you read this report thoroughly before performing an upgrade.

You can run the readiness check while your existing Oracle Fusion Middleware domain is online (while other users are actively using it) or offline.

You can run the readiness check any number of times before performing any actual upgrade. However, do not run the readiness check after an upgrade has been performed, as the report results may differ from the result of pre-upgrade readiness checks.

 **Note:**

To prevent performance from being affected, Oracle recommends that you run the readiness check during off-peak hours.

Starting the Upgrade Assistant in Readiness Mode

Use the `-readiness` parameter to start the Upgrade Assistant in readiness mode.

To perform a readiness check on your pre-upgrade environment with the Upgrade Assistant:

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant.
 - (UNIX) `./ua -readiness`
 - (Windows) `ua.bat -readiness`

 **Note:**

If the `DISPLAY` environment variable is not set up properly to allow for GUI mode, you may encounter the following error:

```
Xlib: connection to ":1.0" refused by server
Xlib: No protocol specified
```

To resolve this issue, set the `DISPLAY` environment variable to the system name or IP address of your local workstation, and rerun Upgrade Assistant.

If you continue to receive these errors after setting `DISPLAY`, try launching another GUI tool, such as `vncconfig`. If you see the same errors, your `DISPLAY` environment variable may still not be set correctly.

For information about other parameters that you can specify on the command line, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 4-2 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.
<code>-response</code>	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
<code>-examine</code>	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.
<code>-logLevel attribute</code>	Optional	Sets the logging level, specifying one of the following attributes: <ul style="list-style-type: none"> TRACE NOTIFICATION WARNING ERROR INCIDENT_ERROR The default logging level is NOTIFICATION. Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.

Table 4-2 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logDir <i>location</i></code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Performing a Readiness Check with the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to complete the pre-upgrade readiness check.

Readiness checks are performed only on schemas or component configurations that are at a supported upgrade starting point.

To complete the readiness check:

1. On the Welcome screen, review information about the readiness check. Click **Next**.
2. On the Readiness Check Type screen, select the readiness check that you want to perform:
 - **Individually Selected Schemas** allows you to select individual schemas for review before upgrade. The readiness check reports whether a schema is supported for an upgrade or where an upgrade is needed. When you select this option, the screen name changes to Selected Schemas.
 - **Domain Based** allows the Upgrade Assistant to discover and select all upgrade-eligible schemas or component configurations in the domain specified in the **Domain Directory** field. When you select this option, the screen name changes to Schemas and Configuration.

Leave the default selection if you want the Upgrade Assistant to check all schemas and component configurations at the same time, or select a specific option:

- **Include checks for all schemas** to discover and review all components that have a schema available to upgrade.
- **Include checks for all configurations** to review component configurations for a managed WebLogic Server domain.

Click **Next**.

3. If you selected **Individually Selected Schemas**: On the Available Components screen, select the components that have a schema available to upgrade for which you want to perform a readiness check.

If you selected **Domain Based**: On the Component List screen, review the list of components that are present in your domain for which you want to perform a readiness check.

If you select a component that has dependent components, those components are automatically selected. For example, if you select Oracle Platform Security Services, Oracle Audit Services is automatically selected.

Depending on the components you select, additional screens may display. For example, you may need to:

- Specify the domain directory.
- Specify schema credentials to connect to the selected schema: **Database Type**, **DBA User Name**, and **DBA Password**. Then click **Connect**.

 **Note:**

Oracle database is the default database type. Make sure that you select the correct database type before you continue. If you discover that you selected the wrong database type, do not go back to this screen to change it to the correct type. Instead, close the Upgrade Assistant and restart the readiness check with the correct database type selected to ensure that the correct database type is applied to all schemas.

- Select the **Schema User Name** option and specify the **Schema Password**.

 **Note:**

The Upgrade Assistant automatically enables default credentials. If you are unable to connect, make sure that you manually enter the credentials for your schema before you continue.

Click **Next** to start the readiness check.

4. On the Readiness Summary screen, review the summary of the readiness checks that will be performed based on your selections.

If you want to save your selections to a response file to run the Upgrade Assistant again later in response (or silent) mode, click **Save Response File** and provide the location and name of the response file. A silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again.

For a detailed report, click **View Log**.

Click **Next**.

5. On the Readiness Check screen, review the status of the readiness check. The process can take several minutes.

If you are checking multiple components, the progress of each component displays in its own progress bar in parallel.

When the readiness check is complete, click **Continue**.

6. On the End of Readiness screen, review the results of the readiness check (**Readiness Success** or **Readiness Failure**):
 - If the readiness check is successful, click **View Readiness Report** to review the complete report. Oracle recommends that you review the Readiness Report before you perform the actual upgrade even when the readiness check is successful. Use the **Find** option to search for a particular word or phrase within the report. The report also indicates where the completed Readiness Check Report file is located.
 - If the readiness check encounters an issue or error, click **View Log** to review the log file, identify and correct the issues, and then restart the readiness check. The log file is managed by the command-line options you set.

Understanding the Readiness Report

After performing a readiness check for your domain, review the report to determine whether you need to take any action for a successful upgrade.

The format of the readiness report file is:

```
readiness<timestamp>.txt
```

Where, *timestamp* indicates the date and time of when the readiness check was run.

A readiness report contains the following information:

Table 4-3 Readiness Report Elements

Report Information	Description	Required Action
Overall Readiness Status: SUCCESS or FAILURE	The top of the report indicates whether the readiness check passed or completed with one or more errors.	If the report completed with one or more errors, search for FAIL and correct the failing issues before attempting to upgrade. You can re-run the readiness check as many times as necessary before an upgrade.
Timestamp	The date and time that the report was generated.	No action required.
Log file location <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs	The directory location of the generated log file.	No action required.
Readiness report location <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs	The directory location of the generated readiness report.	No action required.

Table 4-3 (Cont.) Readiness Report Elements

Report Information	Description	Required Action
Names of components that were checked	The names and versions of the components included in the check and status.	If your domain includes components that cannot be upgraded to this release, such as SOA Core Extension, do not attempt an upgrade.
Names of schemas that were checked	The names and current versions of the schemas included in the check and status.	Review the version numbers of your schemas. If your domain includes schemas that cannot be upgraded to this release, do not attempt an upgrade.
Individual Object Test Status: FAIL	The readiness check test detected an issue with a specific object.	Do not upgrade until all failed issues have been resolved.
Individual Object Test Status: PASS	The readiness check test detected no issues for the specific object.	If your readiness check report shows only the PASS status, you can upgrade your environment. Note, however, that the Readiness Check cannot detect issues with externals such as hardware or connectivity during an upgrade. You should always monitor the progress of your upgrade.
Completed Readiness Check of <Object> Status: FAILURE	The readiness check detected one or more errors that must be resolved for a particular object such as a schema, an index, or datatype.	Do not upgrade until all failed issues have been resolved.
Completed Readiness Check of <Object> Status: SUCCESS	The readiness check test detected no issues.	No action required.

Here is a sample Readiness Report file. Your report may not include all of these checks.

```
Upgrade readiness check completed with one or more errors.
```

```
This readiness check report was created on Tue March 30 11:15:52 EDT 2019
```

```
Log file is located at: ORACLE_HOME/oracle_common/upgrade/logs/ua2016-05-30-11-14-06AM.log
```

```
Readiness Check Report File: ORACLE_HOME/oracle_common/upgrade/logs/readiness2016-05-30-11-15-52AM.txt
```

```
Starting readiness check of components.
```

```
Oracle Metadata Services
```

```
Starting readiness check of Oracle Metadata Services.
```

```
Schema User Name: DEV11_MDS
```

```
Database Type: Oracle Database
```

```
Database Connect String: machinename@yourcompany.com
```

```
VERSION Schema DEV11_MDS is currently at version 12.2.1.4.0.
```

```
Readiness checks will now be performed.
```

```
Starting schema test: TEST_REQUIRED_TABLES Test that the schema
```

```

contains all the required tables
  Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema
contains all the required tables +++ PASS
  Starting schema test: TEST_REQUIRED_PROCEDURES Test that the schema
contains all the required stored procedures
EXCEPTION Schema is missing a required procedure:
GETREPOSITORYFEATURES
  Completed schema test: TEST_REQUIRED_PROCEDURES --> Test that the schema
contains all the required stored procedures +++ FAIL
  Starting schema test: TEST_REQUIRED_VIEWS Test that the schema contains
all the required database views
  Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema
contains all the required database views +++ PASS
  Starting index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes
  Completed index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes +++ PASS
  Starting index test for table MDS_COMPONENTS: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes
  Completed index test for table MDS_TXN_LOCKS: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes +++ PASS
  Starting schema test: TEST_REQUIRED_TRIGGERS Test that the schema has
all the required triggers
  Completed schema test: TEST_REQUIRED_TRIGGERS --> Test that the schema
has all the required triggers +++ PASS
  Starting schema test: TEST_MISSING_COLUMNS Test that tables and views
are not missing any required columns
  Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and
views are not missing any required columns +++ PASS
  Starting schema test: TEST_UNEXPECTED_TABLES Test that the schema does
not contain any unexpected tables
  Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema
does not contain any unexpected tables +++ PASS
  Starting schema test: TEST_UNEXPECTED_PROCEDURES Test that the schema
does not contain any unexpected stored procedures
  Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the
schema does not contain any unexpected stored procedures +++ PASS
  Starting schema test: TEST_UNEXPECTED_VIEWS Test that the schema does
not contain any unexpected views
  Completed schema test: TEST_UNEXPECTED_VIEWS --> Test that the schema
does not contain any unexpected views +++ PASS
  Starting index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --
> Test that the table does not contain any unexpected indexes
  Completed index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --
> Test that the table does not contain any unexpected indexes +++ PASS
  Completed index test for table MDS_LABELS: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
  Starting index test for table MDS_LARGE_ATTRIBUTES:
TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any
unexpected indexes
  Starting schema test: TEST_UNEXPECTED_TRIGGERS Test that the schema
does not contain any unexpected triggers
  Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema
does not contain any unexpected triggers +++ PASS
  Starting schema test: TEST_UNEXPECTED_COLUMNS Test that tables and

```

```
views do not contain any unexpected columns
  Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables
and views do not contain any unexpected columns +++ PASS
  Starting datatype test for table MDS_ATTRIBUTES:
TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the
proper datatypes
  Completed datatype test for table MDS_ATTRIBUTES:
TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the
proper datatypes +++ PASS
  Starting datatype test for table MDS_COMPONENTS:
TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the
proper datatypes
  Starting permissions test: TEST_DBA_TABLE_GRANTS Test that DBA
user has privilege to view all user tables
  Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA
user has privilege to view all user tables +++ PASS
  Starting schema test: TEST_ENOUGH_TABLESPACE Test that the schema
tablespaces automatically extend if full
  Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the
schema tablespaces automatically extend if full +++ PASS
  Starting schema test: TEST_USER_TABLESPACE_QUOTA Test that
tablespace quota for this user is sufficient to perform the upgrade
  Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that
tablespace quota for this user is sufficient to perform the upgrade ++
+ PASS
  Starting schema test: TEST_ONLINE_TABLESPACE Test that schema
tablespaces are online
  Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema
tablespaces are online +++ PASS
  Starting schema test: TEST_DATABASE_VERSION Test that the
database server version number is supported for upgrade
  INFO Database product version: Oracle Database 12c Enterprise
Edition Release 12.2.1.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options
  Completed schema test: TEST_DATABASE_VERSION --> Test that the
database server version number is supported for upgrade +++ PASS
  Finished readiness check of Oracle Metadata Services with status:
FAILURE.
```

If you are running the 12.1.3.0 version of Oracle Fusion Middleware IAU Schemas, and those schemas were upgraded from 11g (11.1.1.7 and later) or 12c (12.1.2.0), your readiness check may fail with the following error:

```
Starting index test for table IAU_COMMON: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
  INFO Audit schema index DYN_EVENT_CATEGORY_INDEX in table IAU_COMMON is
missing the required columns or index itself is missing. This maybe caused by
a known issue, anyway, this missing index will be added in 12.2.2 upgrade.
  INFO Audit schema index DYN_EVENT_TYPE_INDEX in table IAU_COMMON is
missing the required columns or index itself is missing. This maybe caused by
a known issue, anyway, this missing index will be added in 12.2.2 upgrade.
  INFO Audit schema index DYN_TENANT_INDEX in table IAU_COMMON is missing
the required columns or index itself is missing. This maybe caused by a known
issue, anyway, this missing index will be added in 12.2.2 upgrade.
```

INFO Audit schema index DYN_USER_INDEX in table IAU_COMMON is missing the required columns or index itself is missing. This maybe caused by a known issue, anyway, this missing index will be added in 12.2.2 upgrade.

INFO Audit schema index DYN_COMPONENT_TYPE_INDEX in table IAU_COMMON is missing the required columns or index itself is missing. This maybe caused by a known issue, anyway, this missing index will be added in 12.2.2 upgrade.

INFO Audit schema index DYN_USER_TENANT_INDEX in table IAU_COMMON is missing the required columns or index itself is missing. This maybe caused by a known issue, anyway, this missing index will be added in 12.2.2 upgrade.

Completed index test for table IAU_COMMON: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes +++ FAIL

 **Note:**

You can ignore the missing index error in the readiness report. This is a known issue. The corresponding missing index is added during the schema upgrade operation. This error does not occur if the schema to be upgraded was created in 12c (12.2.1.4.0) by using the RCU.

Stopping Servers and Processes

Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.

An Oracle Fusion Middleware environment can consist of an Oracle WebLogic Server domain, an Administration Server, multiple managed servers, Java components, system components such as Identity Management components, and a database used as a repository for metadata. The components may be dependent on each other, so they must be stopped in the correct order.

 **Note:**

The procedures in this section describe how to stop the existing, pre-upgrade servers and processes using the WLST command-line utility or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See Starting and Stopping Administration and Managed Servers and Node Manager.

To stop your pre-upgrade Fusion Middleware environment, navigate to the pre-upgrade domain and follow the steps below:

Step 1: Stop System Components

To stop system components, such as Oracle HTTP Server, use the `stopComponent` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopComponent.sh component_name`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopComponent.cmd component_name`

You can stop system components in any order.

Step 2: Stop the Managed Servers

To stop a WebLogic Server Managed Server, use the `stopManagedWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopManagedWebLogic.sh`
`managed_server_name admin_url`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopManagedWebLogic.cmd`
`managed_server_name admin_url`

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

Stop any Oracle Identity Management components, such as Oracle Internet Directory:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopComponent.sh` `component_name`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopComponent.cmd` `component_name`

Step 4: Stop the Administration Server

When you stop the Administration Server, you also stop the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To stop the Administration Server, use the `stopWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopWebLogic.sh`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Step 5: Stop Node Manager

To stop Node Manager, close the command shell in which it is running.

Alternatively, after setting the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down. See `stopNodeManager` in *WLST Command Reference for WebLogic Server*.

Creating the Required 12c Schemas with the RCU

When upgrading from 11g, you must create the required 12c schemas. You can use the Repository Creation Utility (RCU) to create customized schemas or, optionally, you can use the Upgrade Assistant to create schemas using the default schema settings. This procedure describes how to create schemas using the RCU. Information about using the Upgrade Assistant to create schemas is covered in the upgrade procedures.

Note:

If you are upgrading from a previous 12c release of Oracle Fusion Middleware, you do not need to re-create these schemas if they already exist. Refer to the steps below to identify the existing schemas in your domain.

The following schemas must exist before you upgrade to 12c. If you are upgrading from 11g, and you are not sure which schemas you currently have, refer to the steps below to identify the existing schemas in your domain. You do not need to re-create these schemas if they already exist.

- **Service Table** schema (*prefix_STB*). This schema is new in 12c and is required for domain-based upgrades. It stores basic schema configuration information (for example, schema prefixes and passwords) that can be accessed and used by other Oracle Fusion Middleware components during the domain creation. This schema is automatically created when you run the Repository Creation Utility (RCU), where you specify the existing schema owner prefix that you used for your other 11g schemas.

 **Note:**

If the Service Table schema does not exist, you may encounter the error message UPGAST-00328 : The schema version registry table does not exist on this database. If that happens it is necessary to create the service table schema in order to run Upgrade Assistant

- **Oracle Platform Security Services (OPSS)** schema (*prefix_OPSS*). This schema is required if you are using an OID-based security store in 11g. This schema is automatically created when you run the Repository Creation Utility (RCU). The only supported LDAP-based OPSS security store is Oracle Internet Directory (OID). An LDAP-based policy store is typically used in production environments. You do not need to reassociate an OID-based security store before upgrade. While the Upgrade Assistant is running, you can select the OPSS schema. The Upgrade Assistant upgrades the OID-based security store automatically.

 **Note:**

The 12c OPSS database schema is required so that you can reference the 12c schema during the reconfiguration of the domain. Your domain continues to use the OID-based security store after the upgrade is complete.

To create the 12c schemas with the RCU:

1. (Optional) If you are upgrading from 11g, and you wish to confirm the schemas which are present in your existing domain, then connect to the database as a user with DBA privileges, and run the following code from SQL*Plus:

```
SET LINE 120
COLUMN MRC_NAME FORMAT A14
COLUMN COMP_ID FORMAT A20
COLUMN VERSION FORMAT A12
COLUMN STATUS FORMAT A9
COLUMN UPGRADED FORMAT A8
SELECT MRC_NAME, COMP_ID, OWNER, VERSION, STATUS, UPGRADED FROM
SCHEMA_VERSION_REGISTRY ORDER BY MRC_NAME, COMP_ID ;
```

2. Verify that a certified JDK already exists on your system by running `java -version` from the command line. For 12c (12.2.1.4.0), the certified JDK is 1.8.0_211 and later.

Ensure that the `JAVA_HOME` environment variable is set to the location of the certified JDK. For example:

- (UNIX) `setenv JAVA_HOME /home/Oracle/Java/jdk1.8.0_211`
- (Windows) `set JAVA_HOME=C:\home\Oracle\Java\jdk1.8.0_211`

Add `$JAVA_HOME/bin` to `$PATH`.

3. Go to the `oracle_common/bin` directory:

- (UNIX) `NEW_ORACLE_HOME/oracle_common/bin`
- (Windows) `NEW_ORACLE_HOME\oracle_common\bin`

4. Start the RCU:

- (UNIX) `./rcu`
- (Windows) `rcu.bat`

5. On the Welcome screen, click **Next**.

6. On the Create Repository screen, select **Create Repository** and then select **System Load and Product Load**.

If you do not have DBA privileges, select **Prepare Scripts for System Load**. This will generate a SQL script containing all the same SQL statements and blocks that would have been called if the RCU were to execute the actions for the selected components. After the script is generated, a user with the necessary SYS or SYSDBA privileges can execute the script to complete the system load phase. Click **Next**.

7. On the Database Connection Details screen, select the **Database Type** and enter the connection information for the database that hosts the 11g schemas. See the pertinent table below.

Table 4-4 Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition

Option	Description and Example
Host Name	Specify the name of the server where your database is running in the following format: examplehost.exampledomain.com For Oracle RAC databases, specify the VIP name or one of the node names in this field.
Port	Specify the port number for your database. The default port number for Oracle databases is 1521.
Service Name	Specify the service name for the database. Typically, the service name is the same as the global database name. For Oracle RAC databases, specify the service name of one of the nodes in this field. For example: examplehost.exampledomain.com
Username	Enter the user name for your database. The default user name is SYS.
Password	Enter the password for your database user.
Role	Select the database user's role from the drop-down list: Normal or SYSDBA

Table 4-5 Connection Credentials for MySQL Databases

Option	Description and Example
Host Name	Specify the host name, IP address, or complete server name in <i>host\server</i> format of the server where your database is running.
Port	Specify the port number for your database.
Database Name	Specify the name of your database.
Username	Specify the name of a user with administrator privileges.
Password	Enter the password for your database user.

Table 4-6 Connection Credentials for Microsoft SQL Server Databases

Option	Description and Example
Unicode Support	Select Yes or No from the drop-down list.
Server Name	Specify the host name, IP address, or complete server name in <i>host\server</i> format of the server where your database is running. MSSQL named instances: A named instance is identified by the network name of the computer and the instance name that you specify during installation. The client must specify both the server name and the instance name when connecting.
Port	Specify the port number for your database.
Database Name	Specify the name of your database.
Username	Specify the name of a user with administrator privileges.
Password	Enter the password for your database user.

Table 4-7 Connection Credentials for IBM DB2 Databases

Option	Description and Example
Server Name	Specify the host name, IP address, or complete server name in <i>host\server</i> format of the server where your database is running.
Port	Specify the port number for your database.
Database Name	Specify the name of your database.
Username	Specify the name of a user with DB Owner privileges. The default user name for IBM DB2 databases is <code>db2admin</code> .
Password	Enter the password for your database user.

If the prerequisite check is successful, click **OK** to continue to the next screen. If the check fails, review the details you entered and try again.

- On the Select Components screen, select **Select existing prefix** and select the prefix that was used to create the existing 11g schemas from the drop-down menu (for example, `DEV11G`). This prefix is used to logically group schemas together for use in this domain.

Select **AS Common Schemas**.

 **Note:**

The Common Infrastructure Services (*prefix_STB*) and Oracle Platform Security Services (*prefix_OPSS*) schemas are selected by default if they have not yet been created.

Make a note of the prefix and schema names for the components you are installing as you will need this information when you configure the installation. Click **Next**.

9. In the Checking Prerequisites dialog, verify that the prerequisites check is successful, then click **OK**.
10. On the Schema Passwords screen, specify the passwords for your schema owners.

Make a note of the passwords you enter on this screen as you will need this information while configuring your product installation.

11. On the Map Tablespaces screen, configure the required tablespace mapping for the schemas you want to create.

Click **Next**, then click **OK** in the confirmation dialog. When the progress dialog shows the tablespace creation is complete, click **OK**.

You see the **Encrypt Tablespace** check box only if you have enabled Transparent Data Encryption (TDE) in the database (Oracle or Oracle EBR) when you start the RCU. Select the **Encrypt Tablespace** check box on the Map Tablespaces screen to encrypt all new tablespaces that the RCU creates.

12. Verify the information on the Summary screen and click **Create** to begin schema creation.

This screen contains information about the log files that were created from this RCU operation. Click on the name of a particular log file to view the contents of that file.

13. Review the information on the Completion Summary screen to verify that the operation is completed successfully. Click **Close** to complete the schema creation.

About Reconfiguring the Domain

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1.4.0).

When you reconfigure a standalone domain, the following items are automatically updated, depending on the applications in the domain:

- WebLogic Server core infrastructure
- Domain version

 **Note:**

The Reconfiguration Wizard does not update any of your own applications that are included in the domain.

Specifically, when you reconfigure a domain, the following occurs:

- The domain version number in the `config.xml` file for the domain is updated to the version of the target Oracle HTTP Server.
- The reconfiguration template for Oracle HTTP Server is automatically selected and applied to the domain. This template defines any reconfiguration tasks that are required to make the standalone domain compatible with the target Oracle HTTP Server version.
- The start scripts are updated.

If you want to preserve your modified start scripts, be sure to back them up before starting the Reconfiguration Wizard.

 **Note:**

When the domain reconfiguration process starts, you can't undo the changes that it makes. Before running the Reconfiguration Wizard, ensure that you have backed up the domain as covered in the pre-upgrade checklist. If an error or other interruption occurs while running the Reconfiguration Wizard, you must restore the domain by copying the files and directories from the backup location to the original domain directory. This is the only way to ensure that the domain has been returned to its original state before reconfiguration.

Follow these instructions to reconfigure the existing domain using the Reconfiguration Wizard. See Reconfiguring WebLogic Domains in *Upgrading Oracle WebLogic Server*.

- [Backing Up the Domain](#)
- [Starting the Reconfiguration Wizard](#)
- [Reconfiguring the Domain with the Reconfiguration Wizard](#)
Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain.

Backing Up the Domain

Before running the Reconfiguration Wizard, create a backup copy of the domain directory.

To create a backup of the domain directory:

1. Copy the source domain to a separate location to preserve the contents.
(Windows) `copy C:\domains\mydomain to C:\domains\mydomain_backup.`
(UNIX) `cp mydomain /domains/mydomain_backup`
2. Before updating the domain on each remote Managed Server, create a backup copy of the domain directory on each remote machine.
3. Verify that the backed up versions of the domain are complete.

If domain reconfiguration fails for any reason, you must copy all files and directories from the backup directory into the original domain directory to ensure that the domain is returned entirely to its original state before reconfiguration.

Starting the Reconfiguration Wizard

Note:

Shut down the administration server and all collocated managed servers before starting the reconfiguration process. See [Stopping Servers and Processes](#) .

To start the Reconfiguration Wizard in graphical mode:

1. Sign in to the system on which the domain resides.
2. Open the command shell (on UNIX operating systems) or open a command prompt window (on Windows operating systems).
3. Go to the `oracle_common/common/bin` directory:
 - (UNIX) `NEW_ORACLE_HOME/oracle_common/common/bin`
 - (Windows) `NEW_ORACLE_HOME\oracle_common\commom\bin`
4. Start the Reconfiguration Wizard with the following logging options:
 - (UNIX) `./reconfig.sh -log=log_file -log_priority=ALL`
 - (Windows) `reconfig.cmd -log=log_file -log_priority=ALL`

where `log_file` is the absolute path of the log file you'd like to create for the domain reconfiguration session. This can be helpful if you need to troubleshoot the reconfiguration process.

The parameter `-log_priority=ALL` ensures that logs are logged in fine mode.

Note:

When you run this command, the following error message might appear to indicate that the default cache directory is not valid:

```
*sys-package-mgr*: can't create package cache dir
```

You can change the cache directory by setting the environment variable `CONFIG_JVM_ARGS`. For example:

```
CONFIG_JVM_ARGS=-Dpython.cachedir=valid_directory
```

Reconfiguring the Domain with the Reconfiguration Wizard

Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain.

To reconfigure the domain:

1. On the Select Domain screen, specify the location of the domain you want to upgrade or click **Browse** to navigate and select the domain directory. Click **Next**.

2. On the Reconfiguration Setup Progress screen, view the progress of the setup process. When complete, click **Next**.

During this process:

- The reconfiguration templates for your installed products, including Fusion Middleware products, are automatically applied. This updates various domain configuration files such as `config.xml`, `config-groups.xml`, and `security.xml` (among others).
 - Schemas, scripts, and other such files that support your Fusion Middleware products are updated.
 - The domain upgrade is validated.
3. On the Domain Mode and JDK screen, select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use. The supported JDK version for 12c (12.2.1.4.0) is 1.8.0_211 and later. Click **Next**.

**Note:**

You cannot change the **Domain Mode** at this stage.

For a list of JDKs that are supported for a specific platform, see Oracle Fusion Middleware Supported System Configurations.

4. On the JDBC Data Sources screen, configure the JDBC data sources defined in your domain source.

The JDBC data sources associated with the products for which you are creating the domain are listed in the lower half of the screen. A JDBC data source contains a pool of database connections that are created when the data source instance is created, deployed or targeted, or at server startup. Applications look up a data source on the JNDI tree, and then request a connection. When the applications no longer need the connections, they return the connections to the connection pool in the data source.

From the **Data Source Name** drop-down list, select the data source(s) for which you want to specify the settings. The values that you specify are displayed in the appropriate columns in the data source list, for the selected data source.

For Oracle RAC Configuration for data sources, you can select one of the three options:

- Convert to GridLink
- Convert to RAC multi data source
- Don't convert

For more information about each option, click **Help**.

After specifying the details, click **Next**.

If you do not select any data sources on the JDBC Data Sources screen, the following warning displays:

Missing Driver

Click **Ok** to proceed without verification, click **Cancel** to return to the JDBC Data Sources page.

In this case, if you click **Ok**, the data sources are not verified.

5. On the JDBC Data Sources Test screen, select the check box for the data source connection you configured on the JDBC Data Sources screen and click **Test Selected Connections** to test the data source connection.

 **Note:**

To test the database connections, the database to which you are connecting must be running. If you do not want to test the connections at this time, do not select any data sources. Click **Next** to continue.

6. On the Database Configuration Type screen, select **RCU Data** to connect to the Server Table (_STB) schema.

Enter the database connection details using the RCU service table (_STB) schema credentials and click **Get RCU Configuration**.

The Reconfiguration Wizard uses this connection to automatically configure the data sources required for components in your domain.

 **Note:**

By default **Oracle's Driver (Thin) for Service connections; Versions: Any** is the selected driver. If you specified an instance name in your connection details — instead of the service name — you must select **Oracle's Driver (Thin) for pooled instance connections; Versions: Any** if you do not change the driver type, then the connection will fail.

 **Note:**

For any existing 12c (12.2.1.3.0) datasource, the reconfiguration will preserve the existing values. For new datasources where the schema was created for 12c (12.2.1.4.0) by the RCU, the default connection data will be retrieved from the _STB schema. If no connection data for a given schema is found in the _STB schema, then the default connection data is used.

If the check is successful, click **Next**. If the check fails, reenter the connection details correctly and try again.

 **Note:**

If you are upgrading from 12c (12.2.1.3.0), and your database has _OPSS or _IAU 12c (12.2.1.3.0) database schemas, you must manually enter database connection details for those schemas. These schemas were not required in 12c and had to be created manually. Users could assign any name to these schemas, therefore the Reconfiguration Wizard does not recognize them. When providing connection information for _IAU, use the IAU_APPEND user information.

7. On the JDBC Component Schema Test screen, select all the component schemas and click **Test Selected Connections** to test the connection for each schema. The result of the test is indicated in the Status column.

When the check is complete, click **Next**.

8. The Node Manager screen is only displayed if the domain you are reconfiguring is currently using a per host Node Manager.

On the Node Manager screen, select the Node Manager configuration to use for the reconfigured domain. The resulting configuration depends on the combination of options you select for **Node Manager Type** and **Node Manager Configuration**.

Table 4-8 Field Descriptions for Node Manager Screen

Option	Description
Per Domain Default Location	If you select this option, the Node Manager home is redefined to <i>DOMAIN_NAME/nodemanager</i> and you cannot edit the Node Manager home.
Per Domain Custom Location	Select this option if you want the per domain Node Manager configuration files to be created in a specific location for this domain. Specify the directory in the Node Manager Home field, or click Browse to use the navigation tree to select the location. The specified directory must be empty. The <i>nodemanager.properties</i> and <i>nodemanager.domains</i> files are created in this directory.
Node Manager Home	If you selected the Per Domain Custom Location option, click Browse to navigate to the directory location that you want to use to store the per domain Node Manager configuration.
Manual Node Manager Setup	If you select this option, creation of the Node Manager configuration for the domain is skipped (all remaining fields cannot be modified), and if you want to use Node Manager in the domain, you must manually configure Node Manager as described in <i>Completing the Node Manager Configuration</i> . The reconfigured domain will still use a per host Node Manager configuration. You should also select this option if your existing domain is not configured to use Node Manager and you do not want to use Node Manager in the reconfigured domain. For more information about Node Manager configuration, see <i>Administering Node Manager for Oracle WebLogic Server</i> .
Node Manager Configuration	Select one of the following two options. These fields are not available if you selected Manual Node Manager Setup .
Create New Configuration	A per domain Node Manager configuration will be automatically created for the reconfigured domain using default settings in <i>nodemanager.properties</i> . If necessary, you can modify <i>nodemanager.properties</i> after the domain has been successfully reconfigured.

Table 4-8 (Cont.) Field Descriptions for Node Manager Screen

Option	Description
Migrate Existing Configuration	The existing per host Node Manager configuration will be migrated to a per domain configuration for the reconfigured domain. This does not include environment-specific settings for ListenAddress, ListenPort, StartScriptName, JavaHome, and LogFile.
Node Manager Home	If you selected the Migrate Existing Configuration option, enter or browse to the Node Manager home directory that you want to migrate to the reconfigured domain.
Apply Oracle Recommended Defaults	<p>If you selected the Migrate Existing Configuration option, select this check box if you want to use Oracle-recommended defaults in the nodemanager.properties file. Deselect this check box if you want to continue using the settings in the nodemanager.properties file being migrated.</p> <p>Oracle-recommended properties with default values are as follows:</p> <pre> LogLimit=0 AuthenticationEnabled=true LogLevel=INFO DomainsFileEnabled=true NativeVersionEnabled=true LogToStderr=true SecureListener=true LogCount=1 StopScriptEnabled=false QuitEnabled=false LogAppend=true StateCheckInterval=500 CrashRecoveryEnabled=false StartScriptEnabled=true LogFormatter=weblogic.nodemanager.server.LogFormatter ListenBacklog=50 </pre>
Node Manager Credentials: Username, Password	Specify the username and password that you want to use to start Node Manager in the reconfigured domain.

- On the Advanced Configuration screen, you can select all categories for which you want to perform advanced configuration. For each category you select, the appropriate configuration screen is displayed to allow you to perform advanced configuration.

 **Note:**

The categories that are listed on the Advanced Configuration screen depend on the resources defined in the templates you selected for the domain.

For this upgrade, select none of the options and click **Next**.

10. On the Configuration Summary screen, review the detailed configuration settings of the domain before continuing.

You can limit the items that are displayed in the right-most panel by selecting a filter option from the **View** drop-down list.

To change the configuration, click **Back** to return to the appropriate screen. To reconfigure the domain, click **Reconfig**.

 **Note:**

The location of the domain does not change when you reconfigure it.

11. The Reconfiguration Progress screen displays the progress of the reconfiguration process.

During this process:

- Domain information is extracted, saved, and updated.
- Schemas, scripts, and other such files that support your Fusion Middleware products are updated.

When the progress bar shows 100%, click **Next**.

12. The End of Configuration screen indicates whether the reconfiguration process completed successfully or failed. It also displays the location of the domain that was reconfigured as well as the Administration Server URL (including the listen port). If the reconfiguration is successful, it displays **Oracle WebLogic Server Reconfiguration Succeeded**.

If the reconfiguration process did not complete successfully, an error message is displayed indicates the reason. Take appropriate action to resolve the issue. If you cannot resolve the issue, contact My Oracle Support.

Note the Domain Location and the Admin Server URL for further operations.

Upgrading Domain Component Configurations

After reconfiguring the domain, use the Upgrade Assistant to upgrade the domain *component* configurations inside the domain to match the updated domain configuration.

- [Starting the Upgrade Assistant](#)
Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).
- [Upgrading Domain Components Using the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to upgrade component configurations in the WebLogic domain.
- [Verifying the Domain-Specific-Component Configurations Upgrade](#)
To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).

To start the Upgrade Assistant:



Note:

Before you start the Upgrade Assistant, make sure that the JVM character encoding is set to UTF-8 for the platform on which the Upgrade Assistant is running. If the character encoding is not set to UTF-8, then you will not be able to download files containing Unicode characters in their names. This can cause the upgrade to fail.

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant:
 - (UNIX) `./ua`
 - (Windows) `ua.bat`

For information about other parameters that you can specify on the command line, such as logging parameters, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 4-9 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.

Table 4-9 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.
<code>-response</code>	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
<code>-examine</code>	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.
<code>-logLevel attribute</code>	Optional	Sets the logging level, specifying one of the following attributes: <ul style="list-style-type: none">• TRACE• NOTIFICATION• WARNING• ERROR• INCIDENT_ERROR The default logging level is NOTIFICATION. Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.

Table 4-9 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logDir <i>location</i></code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Upgrading Domain Components Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade component configurations in the WebLogic domain.

After running the Reconfiguration Wizard to reconfigure the WebLogic domain to 12c (12.2.1.4.0), you must run the Upgrade Assistant to upgrade the domain *component* configurations to match the updated domain configuration.

To upgrade domain component configurations with the Upgrade Assistant:

1. On the Welcome screen, review an introduction to the Upgrade Assistant and information about important pre-upgrade tasks. Click **Next**.

Note:

For more information about any Upgrade Assistant screen, click **Help** on the screen.

2. On the WebLogic Components screen, select the **WebLogic Component Configurations** option to upgrade component configurations for a managed WebLogic Server domain. Enter the connection details required to connect to the WebLogic Administration Server that is managing the domain and click **Next**.
3. On the Component List screen, verify that the list includes all the components for which you want to upgrade configurations and click **Next**.

If you do not see the components you want to upgrade, click **Back** to go to the previous screen and specify a different domain.

4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

 **Note:**

The Upgrade Assistant does not verify whether the prerequisites have been met.

5. On the Examine screen, review the status of the Upgrade Assistant as it examines each component, verifying that the component configuration is ready for upgrade. If the status is **Examine finished**, click **Next**.

If the examine phase fails, Oracle recommends that you cancel the upgrade by clicking **No** in the Examination Failure dialog. Click **View Log** to see what caused the error and refer to Troubleshooting Your Upgrade in *Upgrading with the Upgrade Assistant* for information on resolving common upgrade errors.

 **Note:**

- If you resolve any issues detected during the examine phase without proceeding with the upgrade, you can start the Upgrade Assistant again without restoring from backup. However, if you proceed by clicking **Yes** in the Examination Failure dialog box, you need to restore your pre-upgrade environment from backup before starting the Upgrade Assistant again.
- Canceling the examination process has no effect on the configuration data; the only consequence is that the information the Upgrade Assistant has collected must be collected again in a future upgrade session.

6. On the Upgrade Summary screen, review the summary of the options you have selected for component configuration upgrade.

The response file collects and stores all the information that you have entered, and enables you to perform a silent upgrade at a later time. The silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file.

Click **Upgrade** to start the upgrade process.

7. On the Upgrade Progress screen, monitor the status of the upgrade.

 **Caution:**

Allow the Upgrade Assistant enough time to perform the upgrade. Do not cancel the upgrade operation unless absolutely necessary. Doing so may result in an unstable environment.

If any components are not upgraded successfully, refer to the Upgrade Assistant log files for more information.

 **Note:**

The progress bar on this screen displays the progress of the current upgrade procedure. It does not indicate the time remaining for the upgrade.

Click **Next**.

8. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard. The Post-Upgrade Actions window describes the manual tasks you must perform to make components functional in the new installation. This window appears only if a component has post-upgrade steps.

If the upgrade fails: On the Upgrade Failure screen, click **View Log** to view and troubleshoot the errors. The logs are available at `ORACLE_HOME/oracle_common/upgrade/logs`.

 **Note:**

If the upgrade fails you must restore your pre-upgrade environment from backup, fix the issues, then restart the Upgrade Assistant.

Verifying the Domain-Specific-Component Configurations Upgrade

To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

To sign in to the Administration Console, go to: `http://administration_server_host:administration_server_port/console`

To sign in to Oracle Enterprise Manager Fusion Middleware Control Console, go to: `http://administration_server_host:administration_server_port/em`

 **Note:**

After upgrade, make sure you run the administration tools from the new 12c (12.2.1.4.0) Oracle home directory and not from the previous Oracle home directory.

During the upgrade process, some OWSM documents, including policy sets and predefined documents such as policies and assertion templates, may need to be upgraded. If a policy set or a predefined document is upgraded, its version number is incremented by 1.

If you created the FMW user to run the Upgrade Assistant, ensure that you delete the account after verifying your upgrade was successful.

Starting Servers and Processes

After a successful upgrade, restart all processes and servers, including the Administration Server and any Managed Servers.

The components may be dependent on each other so they must be started in the correct order.

 **Note:**

The procedures in this section describe how to start servers and process using the WLST command line or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See *Starting and Stopping Administration and Managed Servers and Node Manager in Administering Oracle Fusion Middleware*.

To start your Fusion Middleware environment, follow the steps below:

Step 1: Start the Administration Server

When you start the Administration Server, you also start the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To start the Administration Server, use the `startWebLogic` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startWebLogic.sh`
- (Windows) `NEW_DOMAIN_HOME\bin\startWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Step 2: Start Node Manager

To start Node Manager, use the `startNodeManager` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startNodeManager.sh`
- (Windows) `NEW_DOMAIN_HOME\bin\startNodeManager.cmd`

Step 3: Start Oracle Identity Management Components

Start any Oracle Identity Management components, such as Oracle Internet Directory, that form part of your environment:

- (UNIX) `NEW_DOMAIN_HOME/bin/startComponent.sh component_name`
- (Windows) `NEW_DOMAIN_HOME\bin\startComponent.cmd component_name`

Step 4: Start the Managed Servers

To start a WebLogic Server Managed Server, use the `startManagedWebLogic` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url`
- (Windows) `NEW_DOMAIN_HOME\bin\startManagedWebLogic.cmd managed_server_name admin_url`

When prompted, enter your user name and password.

 **Note:**

The startup of a Managed Server will typically start the applications that are deployed to it. Therefore, it should not be necessary to manually start applications after the Managed Server startup.

Step 5: Start System Components

To start system components, such as Oracle HTTP Server, use the `startComponent` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startComponent.sh component_name`
- (Windows) `NEW_DOMAIN_HOME\bin\startComponent.cmd component_name`

You can start system components in any order.

- [Starting the Oracle HTTP Server](#)
You can use Enterprise Manager Fusion Middleware Control to start, stop, and restart Oracle HTTP Server.

Starting the Oracle HTTP Server

You can use Enterprise Manager Fusion Middleware Control to start, stop, and restart Oracle HTTP Server.

You can start the Oracle HTTP Server with the Node Manager by following the procedure mentioned in Running Oracle HTTP Server Remotely in *Administering Oracle HTTP Server*.

To start an Oracle HTTP Server using Enterprise Manager Fusion Middleware Control:

1. Go to the Oracle HTTP Server home page.
2. From the Oracle HTTP Server menu:
 - a. Select **Control**.

Install the new patches to update the OHS Database client and OSS patches to use 19c Database client libraries. For more information about upgrading the new Database client 19c for Oracle Fusion Middleware 12.2.1.4.0, see [Doc ID 2921245.1](#).

Part II

Upgrading Oracle HTTP Server from a Previous 12c Release

Part II contains the following chapters:

- [Upgrading a Standalone Oracle HTTP Server from a Previous 12c Release](#)
- [Upgrading a Managed Oracle HTTP Server from a Previous 12c Release](#)
This chapter describes the procedure to upgrade a managed Oracle HTTP Server from a previous 12c release to the latest 12c release . The valid starting point for this upgrade are 12c releases (12.1.3.0, 12.2.1.2, and 12.2.1.3).

5

Upgrading a Standalone Oracle HTTP Server from a Previous 12c Release

This chapter describes the process to upgrade a Standalone Oracle HTTP Server to the latest 12c from a previous Standalone Oracle HTTP Server 12c release.



Note:

The information in this chapter assumes that you have read and performed the required pre-upgrade tasks listed in Planning an Upgrade to Oracle Fusion Middleware 12c (12.2.1.4.0) in *Planning an Upgrade of Oracle Fusion Middleware*.

This chapter contains the following sections:

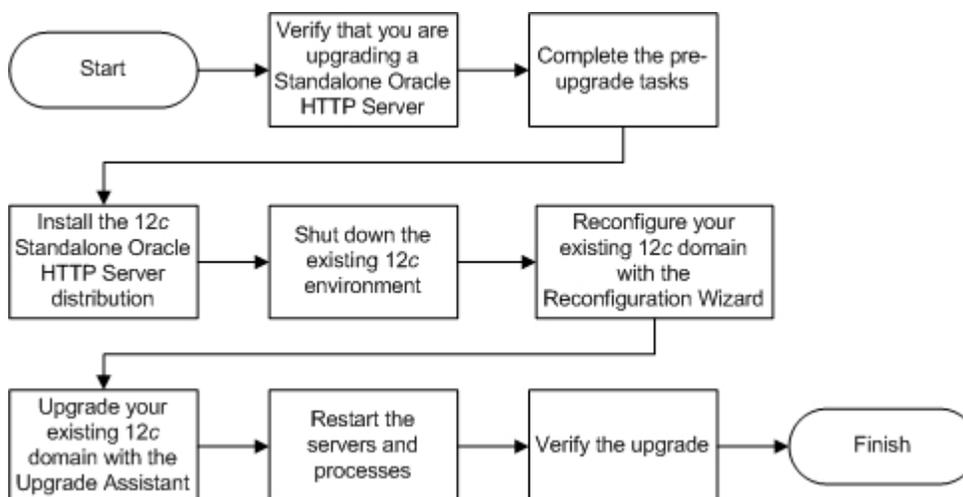
- [About the Standalone Oracle HTTP Server Upgrade Process from a Previous 12c Release](#)
Review the flowchart and roadmap for an overview of the upgrade process for Oracle HTTP Server.
- [Installing the Standalone Oracle HTTP Server](#)
Before you begin your upgrade, download the Oracle HTTP Server 12c (12.2.1.4.0) distribution on the target system and install it by using the Oracle Universal Installer.
- [Stopping Servers and Processes](#)
Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.
- [About Reconfiguring the Domain](#)
Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1.4.0).
- [Upgrading Standalone System Component Configurations](#)
Use the Upgrade Assistant to upgrade the standalone agent's component configurations when the agent is managed by OPMN. The standalone component configuration upgrade does not support upgrade of a standalone agent that is not managed by OPMN.
- [Verifying the Upgrade](#)
You can verify that the upgrade is successful if you are able to start the Node Manager and the Standalone Oracle HTTP Server properly.
- [Starting Servers and Processes](#)
After a successful upgrade, restart all processes and servers, including the Administration Server and any Managed Servers.
- [Post-Upgrade Task](#)
Perform the following update after upgrading the Oracle HTTP Server.

About the Standalone Oracle HTTP Server Upgrade Process from a Previous 12c Release

Review the flowchart and roadmap for an overview of the upgrade process for Oracle HTTP Server.

Figure 5-1 shows the high-level procedures associated with a standalone Oracle HTTP Server upgrade when the starting point is another 12c release. The tools used for each step are also listed.

Figure 5-1 Upgrade Process Flowchart for Standalone Oracle HTTP Server from a Previous 12c release



The following table describes the tasks that must be completed to upgrade a standalone Oracle HTTP Server from a previous release of 12c to the latest version.

Table 5-1 Tasks for Upgrading Standalone Oracle HTTP Server from a Previous 12c Release

Task	Description
Required Verify that you are upgrading a standalone Oracle HTTP Server.	To determine which Oracle HTTP Server you have in your existing environment, see Determining whether Oracle HTTP Server is Standalone or Managed (Collocated) .
Required Complete the pre-upgrade tasks.	The pre-upgrade tasks include cloning your production environment, verifying system requirements and certifications, purging unused data, and creating a non-SYSDBA user. For a complete list of pre-upgrade tasks, see Preparing to Upgrade Oracle HTTP Server .

Table 5-1 (Cont.) Tasks for Upgrading Standalone Oracle HTTP Server from a Previous 12c Release

Task	Description
Required Install the Standalone Oracle HTTP Server.	Run the installation program to install the software. Select the installation type Standalone Oracle HTTP Server (managed independently of WebLogic server) . This transfers the software to your system and creates a new Oracle home directory. See Installing the Standalone Oracle HTTP Server .
Required Shut down the 12c environment.	See Stopping Servers and Processes .
Required Reconfigure your existing domain.	Run the Reconfiguration Wizard from 12c Oracle HTTP Server installation to reconfigure the existing domain. See About Reconfiguring the Domain .
Required Upgrade the standalone system component configurations.	See Upgrading Standalone System Component Configurations .
Required Restart the servers and processes.	See Starting Servers and Processes .
Required Verify the upgrade.	Your Oracle HTTP Server should continue to function as expected. If you have post-upgrade issues, you should troubleshoot the installation and retry the upgrade. See Troubleshooting Oracle HTTP Server in Administering Oracle HTTP Server .

Installing the Standalone Oracle HTTP Server

Before you begin your upgrade, download the Oracle HTTP Server 12c (12.2.1.4.0) distribution on the target system and install it by using the Oracle Universal Installer.

To install the standalone Oracle HTTP Server:

1. Sign in to the target system.
2. Download the following from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:

Oracle HTTP Server (UNIX: `fmw_12.2.1.4.0_ohs_linux64.bin`), (Windows: `setup_fmw_12.2.1.4.0_ohs_win64.exe`)

3. Change to the directory where you downloaded the 12c (12.2.1.4.0) product distribution.
4. Check that your machine meet the following requirements:
 - Ensure that the system, patch, kernel, and other requirements are met as specified in the Roadmap for Verifying Your System Environment.
 - Because Oracle HTTP Server is installed by default on port 7777, you must ensure that port 7777 is not used by any service on the nodes. To check if this port is in use, run the following command before you install Oracle HTTP Server. You must free the port if it is in use.

```
netstat -an | grep 7777
```

5. Enter the following command to start the installation program:
(UNIX) `./fmw_12.2.1.4.0_ohs_linux64.bin`
(Windows) `setup_fmw_12.2.1.4.0_ohs_win64.exe`
6. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an 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, and click **Next**.

 **Note:**

The Installation Inventory Setup screen does not appear on Windows operating systems.

7. On the Welcome screen, review the information to make sure that you have met all the prerequisites. Click **Next**.
8. On the Auto Updates screen, select an option:
 - **Skip Auto Updates:** If you do not want your system to check for software updates at this time.
 - **Select patches from directory:** To navigate to a local directory if you downloaded patch files.
 - **Search My Oracle Support for Updates:** To automatically download software updates if you have a My Oracle Support account. You must enter Oracle Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

9. On the Installation Location screen, specify the location for the Oracle home directory and click **Next**.

As you are installing the standard installation topology for Oracle HTTP Server in a standalone domain, you can specify an Oracle home directory of your choice. However, ensure that you install the software in a new Oracle home.

See *Selecting Directories for Installation and Configuration* in *Planning an Installation of Oracle Fusion Middleware*.

10. On the Installation Type screen, select **Standalone HTTP Server (Managed independently of WebLogic server)** and click **Next**.
11. The Prerequisite Checks screen analyzes the host computer to ensure that the specific operating system prerequisites have been met.

To view the list of tasks that are 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).
12. On the Security Updates screen, indicate how you would like to receive security updates if you already have an Oracle Support account.

If you do not have an Oracle Support account and you are sure that you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.

13. On the Installation Summary screen, verify the installation options that 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.

Click **Install** to begin the installation.

14. On the Installation Progress screen, when the progress bar displays 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.

15. The Installation Complete screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.

You have installed the Oracle HTTP Server in a standalone mode.

Stopping Servers and Processes

Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.

An Oracle Fusion Middleware environment can consist of an Oracle WebLogic Server domain, an Administration Server, multiple managed servers, Java components, system components such as Identity Management components, and a database used as a repository for metadata. The components may be dependent on each other, so they must be stopped in the correct order.

Note:

The procedures in this section describe how to stop the existing, pre-upgrade servers and processes using the WLST command-line utility or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See Starting and Stopping Administration and Managed Servers and Node Manager.

To stop your pre-upgrade Fusion Middleware environment, navigate to the pre-upgrade domain and follow the steps below:

Step 1: Stop System Components

To stop system components, such as Oracle HTTP Server, use the `stopComponent` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopComponent.sh component_name`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopComponent.cmd component_name`

You can stop system components in any order.

Step 2: Stop the Managed Servers

To stop a WebLogic Server Managed Server, use the `stopManagedWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopManagedWebLogic.sh`
`managed_server_name admin_url`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopManagedWebLogic.cmd`
`managed_server_name admin_url`

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

Stop any Oracle Identity Management components, such as Oracle Internet Directory:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopComponent.sh` `component_name`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopComponent.cmd` `component_name`

Step 4: Stop the Administration Server

When you stop the Administration Server, you also stop the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To stop the Administration Server, use the `stopWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopWebLogic.sh`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Step 5: Stop Node Manager

To stop Node Manager, close the command shell in which it is running.

Alternatively, after setting the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down. See `stopNodeManager` in *WLST Command Reference for WebLogic Server*.

About Reconfiguring the Domain

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1.4.0).

When you reconfigure a standalone domain, the following items are automatically updated, depending on the applications in the domain:

- WebLogic Server core infrastructure
- Domain version



Note:

The Reconfiguration Wizard does not update any of your own applications that are included in the domain.

Specifically, when you reconfigure a domain, the following occurs:

- The domain version number in the `config.xml` file for the domain is updated to the version of the target Oracle HTTP Server.
- The reconfiguration template for Oracle HTTP Server is automatically selected and applied to the domain. This template defines any reconfiguration tasks that are required to make the standalone domain compatible with the target Oracle HTTP Server version.
- The start scripts are updated.

If you want to preserve your modified start scripts, be sure to back them up before starting the Reconfiguration Wizard.

 **Note:**

When the domain reconfiguration process starts, you can't undo the changes that it makes. Before running the Reconfiguration Wizard, ensure that you have backed up the domain as covered in the pre-upgrade checklist. If an error or other interruption occurs while running the Reconfiguration Wizard, you must restore the domain by copying the files and directories from the backup location to the original domain directory. This is the only way to ensure that the domain has been returned to its original state before reconfiguration.

Follow these instructions to reconfigure the existing domain using the Reconfiguration Wizard. See *Reconfiguring WebLogic Domains in Upgrading Oracle WebLogic Server*.

- [Backing Up the Domain](#)
- [Starting the Reconfiguration Wizard](#)
- [Reconfiguring the Existing Domain with the Reconfiguration Wizard](#)

Backing Up the Domain

Before running the Reconfiguration Wizard, create a backup copy of the domain directory.

To create a backup of the domain directory:

1. Copy the source domain to a separate location to preserve the contents.
(Windows) `copy C:\domains\mydomain to C:\domains\mydomain_backup.`
(UNIX) `cp mydomain /domains/mydomain_backup`
2. Before updating the domain on each remote Managed Server, create a backup copy of the domain directory on each remote machine.
3. Verify that the backed up versions of the domain are complete.

If domain reconfiguration fails for any reason, you must copy all files and directories from the backup directory into the original domain directory to ensure that the domain is returned entirely to its original state before reconfiguration.

Starting the Reconfiguration Wizard

 **Note:**

Shut down the administration server and all collocated managed servers before starting the reconfiguration process. See [Stopping Servers and Processes](#).

To start the Reconfiguration Wizard in graphical mode:

1. Sign in to the system on which the domain resides.
2. Open the command shell (on UNIX operating systems) or open a command prompt window (on Windows operating systems).
3. Go to the `oracle_common/common/bin` directory:
 - (UNIX) `NEW_ORACLE_HOME/oracle_common/common/bin`
 - (Windows) `NEW_ORACLE_HOME\oracle_common\commom\bin`
4. Start the Reconfiguration Wizard with the following logging options:
 - (UNIX) `./reconfig.sh -log=log_file -log_priority=ALL`
 - (Windows) `reconfig.cmd -log=log_file -log_priority=ALL`

where `log_file` is the absolute path of the log file you'd like to create for the domain reconfiguration session. This can be helpful if you need to troubleshoot the reconfiguration process.

The parameter `-log_priority=ALL` ensures that logs are logged in fine mode.

 **Note:**

When you run this command, the following error message might appear to indicate that the default cache directory is not valid:

```
*sys-package-mgr*: can't create package cache dir
```

You can change the cache directory by setting the environment variable `CONFIG_JVM_ARGS`. For example:

```
CONFIG_JVM_ARGS=-Dpython.cachedir=valid_directory
```

Reconfiguring the Existing Domain with the Reconfiguration Wizard

Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain.

To reconfigure the domain with the Reconfiguration Wizard:

1. On the Select Domain screen, specify the location of the domain you want to upgrade or click **Browse** to navigate and select the domain directory. Click **Next**.

2. On the Reconfiguration Setup Progress screen, view the progress of the setup process. When complete, click **Next**.

During this process:

- The reconfiguration templates for your installed products, including Fusion Middleware products, are automatically applied. This updates various domain configuration files such as `config.xml`, `config-groups.xml`, and `security.xml` (among others).
 - Scripts and other such files that support Oracle HTTP Server are updated.
 - The domain upgrade is validated.
3. On the Domain Mode and JDK screen, select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use. The supported JDK version for 12c (12.2.1.4.0) is 1.8.0_211 and later. Click **Next**.

 **Note:**

You cannot change the **Domain Mode** at this stage.

For a list of JDKs that are supported for a specific platform, see Oracle Fusion Middleware Supported System Configurations.

4. The Node Manager screen is only displayed if the domain you are reconfiguring is currently using a per host Node Manager.

On the Node Manager screen, select the Node Manager configuration to use for the reconfigured domain. The resulting configuration depends on the combination of options you select for **Node Manager Type** and **Node Manager Configuration**.

Table 5-2 Field Descriptions for Node Manager Screen

Option	Description
Per Domain Default Location	If you select this option, the Node Manager home is redefined to <code>DOMAIN_NAME/nodemanager</code> and you cannot edit the Node Manager home.
Per Domain Custom Location	Select this option if you want the per domain Node Manager configuration files to be created in a specific location for this domain. Specify the directory in the Node Manager Home field, or click Browse to use the navigation tree to select the location. The specified directory must be empty. The <code>nodemanager.properties</code> and <code>nodemanager.domains</code> files are created in this directory.
Node Manager Home	If you selected the Per Domain Custom Location option, click Browse to navigate to the directory location that you want to use to store the per domain Node Manager configuration.

Table 5-2 (Cont.) Field Descriptions for Node Manager Screen

Option	Description
Manual Node Manager Setup	<p>If you select this option, creation of the Node Manager configuration for the domain is skipped and the remaining fields cannot be modified. If you want to use Node Manager in the domain, you must manually configure Node Manager as described in <i>Completing the Node Manager Configuration</i>. The reconfigured domain will still use a per host Node Manager configuration.</p> <p>You should also select this option if your existing domain is not configured to use Node Manager and you do not want to use Node Manager in the reconfigured domain.</p> <p>For more information about Node Manager configuration, see <i>Administering Node Manager for Oracle WebLogic Server</i>.</p>
Node Manager Configuration	<p>Select one of the following two options. These fields are not available if you selected Manual Node Manager Setup.</p>
Create New Configuration	<p>A per domain Node Manager configuration is automatically created for the reconfigured domain using default settings in <code>nodemanager.properties</code>. If necessary, you can modify <code>nodemanager.properties</code> after the domain has been successfully reconfigured.</p>
Migrate Existing Configuration	<p>The existing per host Node Manager configuration is migrated to a per domain configuration for the reconfigured domain. This does not include environment-specific settings for <code>ListenAddress</code>, <code>ListenPort</code>, <code>StartScriptName</code>, <code>JavaHome</code>, and <code>LogFile</code>.</p>
Node Manager Home	<p>If you selected the Migrate Existing Configuration option, enter or browse to the Node Manager home directory that you want to migrate to the reconfigured domain.</p>

Table 5-2 (Cont.) Field Descriptions for Node Manager Screen

Option	Description
Apply Oracle Recommended Defaults	<p>If you selected the Migrate Existing Configuration option, select this check box if you want to use Oracle-recommended defaults in the <code>nodemanager.properties</code> file. Deselect this check box if you want to continue using the settings in the <code>nodemanager.properties</code> file being migrated.</p> <p>Oracle-recommended properties with default values are as follows:</p> <pre> LogLimit=0 AuthenticationEnabled=true LogLevel=INFO DomainsFileEnabled=true NativeVersionEnabled=true LogToStderr=true SecureListener=true LogCount=1 StopScriptEnabled=false QuitEnabled=false LogAppend=true StateCheckInterval=500 CrashRecoveryEnabled=false StartScriptEnabled=true LogFormatter=weblogic.nodemanager.server.LogFormatter ListenBacklog=50 </pre>
Node Manager Credentials: Username and Password	Specify the user name and password that you want to use to start Node Manager in the reconfigured domain.

- On the Configuration Summary screen, review the detailed configuration settings of the domain before continuing.

You can limit the items that are displayed in the right-most panel by selecting a filter option from the **View** drop-down list.

To change the configuration, click **Back** to return to the appropriate screen. To reconfigure the domain, click **Reconfig**.

 **Note:**

The location of the domain does not change when you reconfigure it.

- The Reconfiguration Progress screen displays the progress of the reconfiguration process.

During this process:

- Domain information is extracted, saved, and updated.
- Scripts and other such files that support Oracle HTTP Server are updated.

When the progress bar shows 100%, click **Next**.

- The End of Configuration screen indicates whether the reconfiguration process completed successfully or failed. It also displays the location of the domain that was

reconfigured. If the reconfiguration is successful, it displays `Domain Configuration Succeeded`.

If the reconfiguration process did not complete successfully, an error message is displayed indicates the reason. Take appropriate action to resolve the issue. If you cannot resolve the issue, contact My Oracle Support.

Note the Domain Location for further operations.

Upgrading Standalone System Component Configurations

Use the Upgrade Assistant to upgrade the standalone agent's component configurations when the agent is managed by OPMN. The standalone component configuration upgrade does not support upgrade of a standalone agent that is not managed by OPMN.

- [Starting the Upgrade Assistant](#)
Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).
- [Upgrading Standalone System Component Configurations Using the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to upgrade standalone system component configurations.
- [Verifying the Domain-Specific-Component Configurations Upgrade](#)
To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).

To start the Upgrade Assistant:

Note:

Before you start the Upgrade Assistant, make sure that the JVM character encoding is set to UTF-8 for the platform on which the Upgrade Assistant is running. If the character encoding is not set to UTF-8, then you will not be able to download files containing Unicode characters in their names. This can cause the upgrade to fail.

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant:
 - (UNIX) `./ua`

- (Windows) `ua.bat`

For information about other parameters that you can specify on the command line, such as logging parameters, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 5-3 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.
<code>-response</code>	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
<code>-examine</code>	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.

Table 5-3 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logLevel attribute</code>	Optional	<p>Sets the logging level, specifying one of the following attributes:</p> <ul style="list-style-type: none"> • TRACE • NOTIFICATION • WARNING • ERROR • INCIDENT_ERROR <p>The default logging level is NOTIFICATION.</p> <p>Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.</p>
<code>-logDir location</code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Upgrading Standalone System Component Configurations Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade standalone system component configurations.

To upgrade the existing 12c domain using the Upgrade Assistant:

1. On the Welcome screen, review an introduction to the Upgrade Assistant and information about important pre-upgrade tasks. Click **Next**.

 **Note:**

For more information about any Upgrade Assistant screen, click **Help** on the screen.

2. On the Standalone Components screen, select **Standalone System Component Configurations**.

Select **Update an Existing Domain** and enter the location of the existing 12c domain in the **Domain Directory** field. You can also click **Browse** to select the existing domain directory by using the navigation tree. Click **Next**.

 **Note:**

- The **Create a New Domain** option should only be used if you are upgrading from version 11g.
- If you have already created a new 12c standalone domain for your upgraded 11g system components, you can extend the standalone domain with the standalone Oracle HTTP Server by using the **Update an Existing Domain** option.
- The **Update an Existing Domain** option should only be used if you are upgrading from a previous 12c release (12.1.2.0.0, 12.1.3.0.0, 12.2.1.0.0, 12.2.1.1.0, 12.2.1.2.0, and 12.2.1.3.0) to the latest 12c release (12.2.1.4.0).

3. On the Component List screen, verify that the list includes all the components for which you want to upgrade configurations and click **Next**.

If you do not see the components you want to upgrade, click **Back** to go to the previous screen and specify a different domain.

4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

 **Note:**

The Upgrade Assistant does not verify whether the prerequisites have been met.

5. On the Instance Directories screen, select **12c Source** and click **Next**.

You do not need to enter the Oracle instance directories when upgrading to the latest 12c release from a previous 12c release.

6. On the Examine screen, review the status of the Upgrade Assistant as it examines each standalone component, verifying that the standalone component configuration is ready for upgrade. If the status is **Examine finished**, click **Next**.

If the examine phase fails, Oracle recommends that you cancel the upgrade by clicking **No** in the Examination Failure dialog. Click **View Log** to see what caused the error and

refer to Troubleshooting Your Upgrade in *Upgrading with the Upgrade Assistant* for information on resolving common upgrade errors.

 **Note:**

- If you resolve any issues detected during the examine phase without proceeding with the upgrade, you can start the Upgrade Assistant again without restoring from backup. However, if you proceed by clicking **Yes** in the Examination Failure dialog box, you need to restore your pre-upgrade environment from backup before starting the Upgrade Assistant again.
- Canceling the examination process has no effect on the configuration data; the only consequence is that the information the Upgrade Assistant has collected must be collected again in a future upgrade session.

7. On the Upgrade Summary screen, review the summary of the options you have selected for component configuration upgrade.

The response file collects and stores all the information that you have entered, and enables you to perform a silent upgrade at a later time. The silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file.

Click **Upgrade** to start the upgrade process.

8. On the Upgrade Progress screen, monitor the status of the upgrade.

 **Caution:**

Allow the Upgrade Assistant enough time to perform the upgrade. Do not cancel the upgrade operation unless absolutely necessary. Doing so may result in an unstable environment.

If any components are not upgraded successfully, refer to the Upgrade Assistant log files for more information.

 **Note:**

The progress bar on this screen displays the progress of the current upgrade procedure. It does not indicate the time remaining for the upgrade.

Click **Next**.

9. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard. The Post-Upgrade Actions window describes the manual tasks you must perform to make components functional in

the new installation. This window appears only if a component has post-upgrade steps.

If the upgrade fails: On the Upgrade Failure screen, click **View Log** to view and troubleshoot the errors. The logs are available at `ORACLE_HOME/oracle_common/upgrade/logs`. Note If the upgrade fails you must restore your pre-upgrade environment from backup, fix the issues, then restart the Upgrade Assistant.

Verifying the Domain-Specific-Component Configurations Upgrade

To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

To sign in to the Administration Console, go to: `http://administration_server_host:administration_server_port/console`

To sign in to Oracle Enterprise Manager Fusion Middleware Control Console, go to: `http://administration_server_host:administration_server_port/em`

Note:

After upgrade, make sure you run the administration tools from the new 12c (12.2.1.4.0) Oracle home directory and not from the previous Oracle home directory.

During the upgrade process, some OWSM documents, including policy sets and predefined documents such as policies and assertion templates, may need to be upgraded. If a policy set or a predefined document is upgraded, its version number is incremented by 1.

If you created the FMW user to run the Upgrade Assistant, ensure that you delete the account after verifying your upgrade was successful.

Verifying the Upgrade

You can verify that the upgrade is successful if you are able to start the Node Manager and the Standalone Oracle HTTP Server properly.

If you experience post-upgrade issues, you need to troubleshoot the installation and retry the upgrade. See Troubleshooting Oracle HTTP Server in *Administrator's Guide for Oracle HTTP Server*.

If you are not able to start the newly upgraded environment, a possible cause could be the use of MD5 certificates in your Oracle wallet. To check whether you are using MD5 signatures and how to replace them with SHA-2 certificates, see [Replacing Certificate Signed Using MD5 Algorithm with Certificate Signed Using SHA-2 Algorithm](#).

- [Starting the Node Manager](#)
Server instances in a WebLogic Server production environment are often distributed across multiple domains, machines, and geographic locations. Node Manager is a WebLogic Server utility that enables you to manage (start, shut down, and restart) your Oracle HTTP Server instance.

- [Starting the Standalone Oracle HTTP Server](#)
After the upgrade, start the standalone Oracle HTTP Server.

Starting the Node Manager

Server instances in a WebLogic Server production environment are often distributed across multiple domains, machines, and geographic locations. Node Manager is a WebLogic Server utility that enables you to manage (start, shut down, and restart) your Oracle HTTP Server instance.

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. For more information, see [Running Node Manager as a Startup Service](#) in *Administering Node Manager for Oracle WebLogic Server*.



Note:

On UNIX platforms, do not run Node Manager as the root user.

Starting the Standalone Oracle HTTP Server

After the upgrade, start the standalone Oracle HTTP Server.

To start the standalone Oracle HTTP Server:

1. Go to the DOMAIN_HOME/bin directory:
(UNIX) ORACLE_HOME/user_projects/domains/<domain name>/bin
(WINDOWS) ORACLE_HOME\user_projects\domains\<domain name>\bin
2. Enter the command to start the standalone server.

On UNIX operating systems:

```
./startComponent.sh ohs_name
```

On Windows operating systems:

```
startComponent.cmd ohs_name
```

See [Starting and Stopping System Components](#) in *Administering Oracle Fusion Middleware*.

Starting Servers and Processes

After a successful upgrade, restart all processes and servers, including the Administration Server and any Managed Servers.

The components may be dependent on each other so they must be started in the correct order.

 **Note:**

The procedures in this section describe how to start servers and process using the WLST command line or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See *Starting and Stopping Administration and Managed Servers and Node Manager in Administering Oracle Fusion Middleware*.

To start your Fusion Middleware environment, follow the steps below:

Step 1: Start the Administration Server

When you start the Administration Server, you also start the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To start the Administration Server, use the `startWebLogic` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startWebLogic.sh`
- (Windows) `NEW_DOMAIN_HOME\bin\startWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Step 2: Start Node Manager

To start Node Manager, use the `startNodeManager` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startNodeManager.sh`
- (Windows) `NEW_DOMAIN_HOME\bin\startNodeManager.cmd`

Step 3: Start Oracle Identity Management Components

Start any Oracle Identity Management components, such as Oracle Internet Directory, that form part of your environment:

- (UNIX) `NEW_DOMAIN_HOME/bin/startComponent.sh component_name`
- (Windows) `NEW_DOMAIN_HOME\bin\startComponent.cmd component_name`

Step 4: Start the Managed Servers

To start a WebLogic Server Managed Server, use the `startManagedWebLogic` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url`
- (Windows) `NEW_DOMAIN_HOME\bin\startManagedWebLogic.cmd managed_server_name admin_url`

When prompted, enter your user name and password.

 **Note:**

The startup of a Managed Server will typically start the applications that are deployed to it. Therefore, it should not be necessary to manually start applications after the Managed Server startup.

Step 5: Start System Components

To start system components, such as Oracle HTTP Server, use the `startComponent` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startComponent.sh component_name`
- (Windows) `NEW_DOMAIN_HOME\bin\startComponent.cmd component_name`

You can start system components in any order.

Post-Upgrade Task

Perform the following update after upgrading the Oracle HTTP Server.

Updating to 19c Database Client:

Install the new patches to update the OHS Database client and OSS patches to use 19c Database client libraries. For more information about upgrading the new Database client 19c for Oracle Fusion Middleware 12.2.1.4.0, see [Doc ID 2921245.1](#).

6

Upgrading a Managed Oracle HTTP Server from a Previous 12c Release

This chapter describes the procedure to upgrade a managed Oracle HTTP Server from a previous 12c release to the latest 12c release . The valid starting point for this upgrade are 12c releases (12.1.3.0, 12.2.1.2, and 12.2.1.3).

This procedure assumes that you have an existing 12c Fusion Middleware Infrastructure installed and configured (WebLogic Domain either in Full-JRF or Restricted JRF mode) on your system.

Note:

You can install and configure a managed/collocated Oracle HTTP Server in either Full-JRF or Restricted-JRF domain. The key difference between the two is that there is no database dependency in the case of the Restricted-JRF mode. See [About Full-JRF and Restricted-JRF Modes](#).

You have to upgrade the database schemas only for the Full-JRF mode, if required.

- [About the Managed Oracle HTTP Server Upgrade Process from a Previous 12c Release](#)
Review the flowchart and roadmap for an overview of the upgrade process for Managed Oracle HTTP Server.
- [Installing the Product Distributions](#)
Before you begin your upgrade, download Oracle Fusion Middleware Infrastructure and Oracle HTTP Server 12c (12.2.1.4.0) distributions on the target system and install them using Oracle Universal Installer.
- [Running a Pre-Upgrade Readiness Check](#)
To identify potential issues with the upgrade, Oracle recommends that you run a readiness check before you start the upgrade process. Be aware that the readiness check may not be able to discover all potential issues with your upgrade. An upgrade may still fail, even if the readiness check reports success.
- [Stopping Servers and Processes](#)
Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.
- [Upgrading Product Schemas \(Applicable for Full-JRF Only\)](#)
After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.
- [About Reconfiguring the Domain](#)
Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1.4.0).

- [Upgrading Domain Component Configurations](#)
After reconfiguring the domain, use the Upgrade Assistant to upgrade the domain *component* configurations inside the domain to match the updated domain configuration.
- [Starting Servers and Processes](#)
After a successful upgrade, restart all processes and servers, including the Administration Server and any Managed Servers.
- [Verifying the Upgrade](#)
If you can log in to the Administration Console, the Enterprise Manager Fusion Middleware Control, and the OHS Home Page, then your upgrade is successful.
- [Post-Upgrade Task](#)
Perform the following update after upgrading the Oracle HTTP Server.

About the Managed Oracle HTTP Server Upgrade Process from a Previous 12c Release

Review the flowchart and roadmap for an overview of the upgrade process for Managed Oracle HTTP Server.

Figure 6-1 shows the process flow to upgrade a Managed Oracle HTTP Server to the latest 12c release. The tools used for each step are also listed.

Figure 6-1 Process Flowchart for the Upgrade of Managed Oracle HTTP Server from a Previous 12c Release

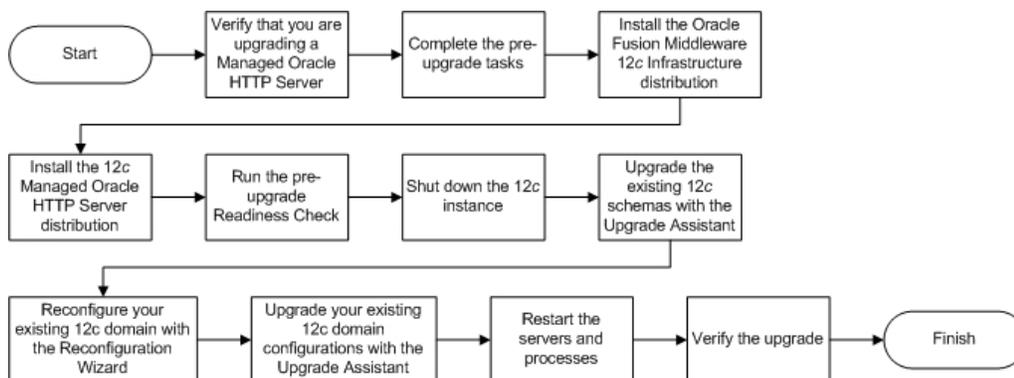


Table 6-1 describes the tasks that must be completed to upgrade a Managed Oracle HTTP Server from a previous 12c release to 12c (12.2.1.4.0).

Table 6-1 Tasks for Upgrading Managed Oracle HTTP Server from a Previous 12c Release

Task	Description
Required Verify that you are upgrading a Managed Oracle HTTP Server.	To determine which Oracle HTTP Server you have in your existing environment, see Determining whether Oracle HTTP Server is Standalone or Managed (Collocated) .

Table 6-1 (Cont.) Tasks for Upgrading Managed Oracle HTTP Server from a Previous 12c Release

Task	Description
<p>Required If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.</p>	<p>The pre-upgrade tasks include cloning your production environment, verifying system requirements and certifications, purging unused data, and creating non-SYSDBA user.</p> <p>For a complete list of pre-upgrade tasks, see Preparing to Upgrade Oracle HTTP Server.</p>
<p>Required Download and install the 12.2.1.4.0 Fusion Middleware Infrastructure and Oracle HTTP Server distributions.</p>	<p>The Infrastructure distribution packs the WebLogic Server and the Java Required Files (JRF) that are required to set up the foundation to install other Fusion Middleware products.</p> <p>As per the upgrade topology defined in this guide, you must install the Infrastructure in a new Oracle home. You must install Oracle HTTP Server in the Oracle home that is created when you installed the 12.2.1.4.0 Infrastructure.</p> <p>See Installing the Product Distributions.</p>
<p>Optional Run the Readiness Check.</p>	<p>Running the Readiness Check helps using the Upgrade Assistant helps you to determine whether your pre-upgrade environment is ready for upgrade. For the complete procedure, see Running a Pre-Upgrade Readiness Check.</p>
<p>Required Shut down the 12c instance.</p>	<p>Before starting the upgrade process, shut down the Administration Server, the Managed Servers, and your 12c instance.</p> <p>See Stopping Servers and Processes.</p>
<p>Required Upgrade the existing schemas with the Upgrade Assistant.</p>	<p>See Upgrading Product Schemas (Applicable for Full-JRF Only).</p>
<p>Required Reconfigure your existing domain.</p>	<p>Run the Reconfiguration Wizard from 12c Oracle HTTP Server installation to reconfigure the existing domain.</p> <p>See Reconfiguring the Existing Domain with the Reconfiguration Wizard.</p>
<p>Required Upgrade your existing domain configurations.</p>	<p>After the installation, you need to use the Upgrade Assistant to upgrade the Oracle HTTP Server and system component infrastructure.</p> <p>See Upgrading Domain Components Using the Upgrade Assistant.</p>
<p>Required Restart the servers and processes.</p>	<p>The upgrade process is complete. You can now restart the Administration Server, the Managed Servers, and your 12.2.1.4.0 instance.</p> <p>See Starting Servers and Processes.</p>
<p>Required Verify the upgrade.</p>	<p>For verifying the upgrade, see Verifying the Upgrade.</p>

Installing the Product Distributions

Before you begin your upgrade, download Oracle Fusion Middleware Infrastructure and Oracle HTTP Server 12c (12.2.1.4.0) distributions on the target system and install them using Oracle Universal Installer.

Note:

When Infrastructure is required for the upgrade, you must install the Oracle Fusion Middleware distribution first before you install other Fusion Middleware products.

To install the 12c (12.2.1.4.0) distributions:

1. Sign in to the target system.
2. Download the following from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle Fusion Middleware Infrastructure (`fmw_12.2.1.4.0_infrastructure_generic.jar`)
 - Oracle HTTP Server (UNIX: `fmw_12.2.1.4.0_ohs_linux64.bin`), (Windows: `setup_fmw_12.2.1.4.0_ohs_win64.exe`)
3. Check that your machines meet the following requirements:
 - Ensure that the system, patch, kernel, and other requirements are met as specified in *Installing and Configuring Oracle HTTP Server*.
 - Because Oracle HTTP Server is installed by default on port 7777, you must ensure that port 7777 is not used by any service on the nodes. To check if this port is in use, run the following command before you install Oracle HTTP Server. You must free the port if it is in use.

```
netstat -an | grep 7777
```

4. On UNIX platforms, if the `/etc/oraInst.loc` file exists, check that its contents are correct. Specifically, check that the inventory directory is correct and that you have write permissions for that directory.

If the `/etc/oraInst.loc` file does not exist, you can skip this step.

5. Change to the directory where you downloaded the 12c (12.2.1.4.0) product distribution.
6. Start the installation program for Oracle Fusion Middleware Infrastructure:
 - (UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.4.0_infrastructure.jar`
 - (Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.4.0_infrastructure.jar`
7. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an 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, and click **Next**.

 **Note:**

The Installation Inventory Setup screen does not appear on Windows operating systems.

8. On the Welcome screen, review the information to make sure that you have met all the prerequisites. Click **Next**.
9. On the Auto Updates screen, select an option:
 - **Skip Auto Updates:** If you do not want your system to check for software updates at this time.
 - **Select patches from directory:** To navigate to a local directory if you downloaded patch files.
 - **Search My Oracle Support for Updates:** To automatically download software updates if you have a My Oracle Support account. You must enter Oracle Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

10. On the Installation Location screen, specify the location for the Oracle home directory and click **Next**.

Since you are installing the standard installation topology for a collocated Oracle HTTP Server in a WebLogic Server domain, enter the path to an existing Oracle Fusion Middleware Infrastructure Oracle home.

See Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.

11. On the Installation Type screen, select the following:
 - For Infrastructure, select **Fusion Middleware Infrastructure**
 - For Oracle HTTP Server, select **Collocated HTTP Server (Managed through WebLogic Server)**

Click **Next**.

12. The Prerequisite Checks screen analyzes the host computer to ensure that the specific operating system prerequisites have been met.

To view the list of tasks that are 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).

13. On the Installation Summary screen, verify the installation options that 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.

Click **Install** to begin the installation.

14. On the Installation Progress screen, when the progress bar displays 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.

15. The Installation Complete screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.
16. After you have installed Oracle Fusion Middleware Infrastructure, enter the following command to start the installer for your product distribution and repeat the steps above to navigate through the installer screens:

```
(UNIX) ./fmw_12.2.1.4.0_ohs_linux64.bin
```

```
(Windows) setup_fmw_12.2.1.4.0_ohs_win64.exe
```

You have installed the Oracle HTTP Server in a collocated mode.

Running a Pre-Upgrade Readiness Check

To identify potential issues with the upgrade, Oracle recommends that you run a readiness check before you start the upgrade process. Be aware that the readiness check may not be able to discover all potential issues with your upgrade. An upgrade may still fail, even if the readiness check reports success.

- [About Running a Pre-Upgrade Readiness Check](#)
You can run the Upgrade Assistant in `-readiness` mode to detect issues before you perform the actual upgrade. You can run the readiness check in GUI mode using the Upgrade Assistant or in silent mode using a response file.
- [Starting the Upgrade Assistant in Readiness Mode](#)
Use the `-readiness` parameter to start the Upgrade Assistant in readiness mode.
- [Performing a Readiness Check with the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to complete the pre-upgrade readiness check.
- [Understanding the Readiness Report](#)
After performing a readiness check for your domain, review the report to determine whether you need to take any action for a successful upgrade.

About Running a Pre-Upgrade Readiness Check

You can run the Upgrade Assistant in `-readiness` mode to detect issues before you perform the actual upgrade. You can run the readiness check in GUI mode using the Upgrade Assistant or in silent mode using a response file.

The Upgrade Assistant readiness check performs a read-only, pre-upgrade review of your Fusion Middleware schemas and WebLogic domain configurations that are at a supported starting point. The review is a read-only operation.

The readiness check generates a formatted, time-stamped readiness report so you can address potential issues before you attempt the actual upgrade. If no issues are detected, you can begin the upgrade process. Oracle recommends that you read this report thoroughly before performing an upgrade.

You can run the readiness check while your existing Oracle Fusion Middleware domain is online (while other users are actively using it) or offline.

You can run the readiness check any number of times before performing any actual upgrade. However, do not run the readiness check after an upgrade has been performed, as the report results may differ from the result of pre-upgrade readiness checks.

**Note:**

To prevent performance from being affected, Oracle recommends that you run the readiness check during off-peak hours.

Starting the Upgrade Assistant in Readiness Mode

Use the `-readiness` parameter to start the Upgrade Assistant in readiness mode.

To perform a readiness check on your pre-upgrade environment with the Upgrade Assistant:

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant.
 - (UNIX) `./ua -readiness`
 - (Windows) `ua.bat -readiness`

**Note:**

If the `DISPLAY` environment variable is not set up properly to allow for GUI mode, you may encounter the following error:

```
Xlib: connection to ":1.0" refused by server
Xlib: No protocol specified
```

To resolve this issue, set the `DISPLAY` environment variable to the system name or IP address of your local workstation, and rerun Upgrade Assistant.

If you continue to receive these errors after setting `DISPLAY`, try launching another GUI tool, such as `vnconfig`. If you see the same errors, your `DISPLAY` environment variable may still not be set correctly.

For information about other parameters that you can specify on the command line, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 6-2 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.
<code>-response</code>	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
<code>-examine</code>	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.
<code>-logLevel attribute</code>	Optional	Sets the logging level, specifying one of the following attributes: <ul style="list-style-type: none"> TRACE NOTIFICATION WARNING ERROR INCIDENT_ERROR The default logging level is NOTIFICATION. Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.

Table 6-2 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logDir <i>location</i></code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Performing a Readiness Check with the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to complete the pre-upgrade readiness check.

Readiness checks are performed only on schemas or component configurations that are at a supported upgrade starting point.

To complete the readiness check:

1. On the Welcome screen, review information about the readiness check. Click **Next**.
2. On the Readiness Check Type screen, select the readiness check that you want to perform:
 - **Individually Selected Schemas** allows you to select individual schemas for review before upgrade. The readiness check reports whether a schema is supported for an upgrade or where an upgrade is needed. When you select this option, the screen name changes to Selected Schemas.
 - **Domain Based** allows the Upgrade Assistant to discover and select all upgrade-eligible schemas or component configurations in the domain specified in the **Domain Directory** field. When you select this option, the screen name changes to Schemas and Configuration.

Leave the default selection if you want the Upgrade Assistant to check all schemas and component configurations at the same time, or select a specific option:

- **Include checks for all schemas** to discover and review all components that have a schema available to upgrade.
- **Include checks for all configurations** to review component configurations for a managed WebLogic Server domain.

Click **Next**.

3. If you selected **Individually Selected Schemas**: On the Available Components screen, select the components that have a schema available to upgrade for which you want to perform a readiness check.

If you selected **Domain Based**: On the Component List screen, review the list of components that are present in your domain for which you want to perform a readiness check.

If you select a component that has dependent components, those components are automatically selected. For example, if you select Oracle Platform Security Services, Oracle Audit Services is automatically selected.

Depending on the components you select, additional screens may display. For example, you may need to:

- Specify the domain directory.
- Specify schema credentials to connect to the selected schema: **Database Type**, **DBA User Name**, and **DBA Password**. Then click **Connect**.

 **Note:**

Oracle database is the default database type. Make sure that you select the correct database type before you continue. If you discover that you selected the wrong database type, do not go back to this screen to change it to the correct type. Instead, close the Upgrade Assistant and restart the readiness check with the correct database type selected to ensure that the correct database type is applied to all schemas.

- Select the **Schema User Name** option and specify the **Schema Password**.

 **Note:**

The Upgrade Assistant automatically enables default credentials. If you are unable to connect, make sure that you manually enter the credentials for your schema before you continue.

Click **Next** to start the readiness check.

4. On the Readiness Summary screen, review the summary of the readiness checks that will be performed based on your selections.

If you want to save your selections to a response file to run the Upgrade Assistant again later in response (or silent) mode, click **Save Response File** and provide the location and name of the response file. A silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again.

For a detailed report, click **View Log**.

Click **Next**.

5. On the Readiness Check screen, review the status of the readiness check. The process can take several minutes.

If you are checking multiple components, the progress of each component displays in its own progress bar in parallel.

When the readiness check is complete, click **Continue**.

6. On the End of Readiness screen, review the results of the readiness check (**Readiness Success** or **Readiness Failure**):
 - If the readiness check is successful, click **View Readiness Report** to review the complete report. Oracle recommends that you review the Readiness Report before you perform the actual upgrade even when the readiness check is successful. Use the **Find** option to search for a particular word or phrase within the report. The report also indicates where the completed Readiness Check Report file is located.
 - If the readiness check encounters an issue or error, click **View Log** to review the log file, identify and correct the issues, and then restart the readiness check. The log file is managed by the command-line options you set.

Understanding the Readiness Report

After performing a readiness check for your domain, review the report to determine whether you need to take any action for a successful upgrade.

The format of the readiness report file is:

```
readiness<timestamp>.txt
```

Where, *timestamp* indicates the date and time of when the readiness check was run.

A readiness report contains the following information:

Table 6-3 Readiness Report Elements

Report Information	Description	Required Action
Overall Readiness Status: SUCCESS or FAILURE	The top of the report indicates whether the readiness check passed or completed with one or more errors.	If the report completed with one or more errors, search for FAIL and correct the failing issues before attempting to upgrade. You can re-run the readiness check as many times as necessary before an upgrade.
Timestamp	The date and time that the report was generated.	No action required.
Log file location <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs	The directory location of the generated log file.	No action required.
Readiness report location <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs	The directory location of the generated readiness report.	No action required.

Table 6-3 (Cont.) Readiness Report Elements

Report Information	Description	Required Action
Names of components that were checked	The names and versions of the components included in the check and status.	If your domain includes components that cannot be upgraded to this release, such as SOA Core Extension, do not attempt an upgrade.
Names of schemas that were checked	The names and current versions of the schemas included in the check and status.	Review the version numbers of your schemas. If your domain includes schemas that cannot be upgraded to this release, do not attempt an upgrade.
Individual Object Test Status: FAIL	The readiness check test detected an issue with a specific object.	Do not upgrade until all failed issues have been resolved.
Individual Object Test Status: PASS	The readiness check test detected no issues for the specific object.	If your readiness check report shows only the PASS status, you can upgrade your environment. Note, however, that the Readiness Check cannot detect issues with externals such as hardware or connectivity during an upgrade. You should always monitor the progress of your upgrade.
Completed Readiness Check of <Object> Status: FAILURE	The readiness check detected one or more errors that must be resolved for a particular object such as a schema, an index, or datatype.	Do not upgrade until all failed issues have been resolved.
Completed Readiness Check of <Object> Status: SUCCESS	The readiness check test detected no issues.	No action required.

Here is a sample Readiness Report file. Your report may not include all of these checks.

```
Upgrade readiness check completed with one or more errors.
```

```
This readiness check report was created on Tue March 30 11:15:52 EDT 2019
```

```
Log file is located at: ORACLE_HOME/oracle_common/upgrade/logs/ua2016-05-30-11-14-06AM.log
```

```
Readiness Check Report File: ORACLE_HOME/oracle_common/upgrade/logs/readiness2016-05-30-11-15-52AM.txt
```

```
Starting readiness check of components.
```

```
Oracle Metadata Services
```

```
Starting readiness check of Oracle Metadata Services.
```

```
Schema User Name: DEV11_MDS
```

```
Database Type: Oracle Database
```

```
Database Connect String: machinename@yourcompany.com
```

```
VERSION Schema DEV11_MDS is currently at version 12.2.1.4.0.
```

```
Readiness checks will now be performed.
```

```
Starting schema test: TEST_REQUIRED_TABLES Test that the schema
```

```

contains all the required tables
  Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema
contains all the required tables +++ PASS
  Starting schema test: TEST_REQUIRED_PROCEDURES Test that the schema
contains all the required stored procedures
EXCEPTION Schema is missing a required procedure:
GETREPOSITORYFEATURES
  Completed schema test: TEST_REQUIRED_PROCEDURES --> Test that the schema
contains all the required stored procedures +++ FAIL
  Starting schema test: TEST_REQUIRED_VIEWS Test that the schema contains
all the required database views
  Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema
contains all the required database views +++ PASS
  Starting index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes
  Completed index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes +++ PASS
  Starting index test for table MDS_COMPONENTS: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes
  Completed index test for table MDS_TXN_LOCKS: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes +++ PASS
  Starting schema test: TEST_REQUIRED_TRIGGERS Test that the schema has
all the required triggers
  Completed schema test: TEST_REQUIRED_TRIGGERS --> Test that the schema
has all the required triggers +++ PASS
  Starting schema test: TEST_MISSING_COLUMNS Test that tables and views
are not missing any required columns
  Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and
views are not missing any required columns +++ PASS
  Starting schema test: TEST_UNEXPECTED_TABLES Test that the schema does
not contain any unexpected tables
  Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema
does not contain any unexpected tables +++ PASS
  Starting schema test: TEST_UNEXPECTED_PROCEDURES Test that the schema
does not contain any unexpected stored procedures
  Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the
schema does not contain any unexpected stored procedures +++ PASS
  Starting schema test: TEST_UNEXPECTED_VIEWS Test that the schema does
not contain any unexpected views
  Completed schema test: TEST_UNEXPECTED_VIEWS --> Test that the schema
does not contain any unexpected views +++ PASS
  Starting index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --
> Test that the table does not contain any unexpected indexes
  Completed index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --
> Test that the table does not contain any unexpected indexes +++ PASS
  Completed index test for table MDS_LABELS: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
  Starting index test for table MDS_LARGE_ATTRIBUTES:
TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any
unexpected indexes
  Starting schema test: TEST_UNEXPECTED_TRIGGERS Test that the schema
does not contain any unexpected triggers
  Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema
does not contain any unexpected triggers +++ PASS
  Starting schema test: TEST_UNEXPECTED_COLUMNS Test that tables and

```

```
views do not contain any unexpected columns
  Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables
and views do not contain any unexpected columns +++ PASS
  Starting datatype test for table MDS_ATTRIBUTES:
TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the
proper datatypes
  Completed datatype test for table MDS_ATTRIBUTES:
TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the
proper datatypes +++ PASS
  Starting datatype test for table MDS_COMPONENTS:
TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the
proper datatypes
  Starting permissions test: TEST_DBA_TABLE_GRANTS Test that DBA
user has privilege to view all user tables
  Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA
user has privilege to view all user tables +++ PASS
  Starting schema test: TEST_ENOUGH_TABLESPACE Test that the schema
tablespaces automatically extend if full
  Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the
schema tablespaces automatically extend if full +++ PASS
  Starting schema test: TEST_USER_TABLESPACE_QUOTA Test that
tablespace quota for this user is sufficient to perform the upgrade
  Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that
tablespace quota for this user is sufficient to perform the upgrade ++
+ PASS
  Starting schema test: TEST_ONLINE_TABLESPACE Test that schema
tablespaces are online
  Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema
tablespaces are online +++ PASS
  Starting schema test: TEST_DATABASE_VERSION Test that the
database server version number is supported for upgrade
  INFO Database product version: Oracle Database 12c Enterprise
Edition Release 12.2.1.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options
  Completed schema test: TEST_DATABASE_VERSION --> Test that the
database server version number is supported for upgrade +++ PASS
  Finished readiness check of Oracle Metadata Services with status:
FAILURE.
```

If you are running the 12.1.3.0 version of Oracle Fusion Middleware IAU Schemas, and those schemas were upgraded from 11g (11.1.1.7 and later) or 12c (12.1.2.0), your readiness check may fail with the following error:

```
Starting index test for table IAU_COMMON: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
  INFO Audit schema index DYN_EVENT_CATEGORY_INDEX in table IAU_COMMON is
missing the required columns or index itself is missing. This maybe caused by
a known issue, anyway, this missing index will be added in 12.2.2 upgrade.
  INFO Audit schema index DYN_EVENT_TYPE_INDEX in table IAU_COMMON is
missing the required columns or index itself is missing. This maybe caused by
a known issue, anyway, this missing index will be added in 12.2.2 upgrade.
  INFO Audit schema index DYN_TENANT_INDEX in table IAU_COMMON is missing
the required columns or index itself is missing. This maybe caused by a known
issue, anyway, this missing index will be added in 12.2.2 upgrade.
```

INFO Audit schema index DYN_USER_INDEX in table IAU_COMMON is missing the required columns or index itself is missing. This maybe caused by a known issue, anyway, this missing index will be added in 12.2.2 upgrade.

INFO Audit schema index DYN_COMPONENT_TYPE_INDEX in table IAU_COMMON is missing the required columns or index itself is missing. This maybe caused by a known issue, anyway, this missing index will be added in 12.2.2 upgrade.

INFO Audit schema index DYN_USER_TENANT_INDEX in table IAU_COMMON is missing the required columns or index itself is missing. This maybe caused by a known issue, anyway, this missing index will be added in 12.2.2 upgrade.

Completed index test for table IAU_COMMON: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes +++ FAIL

 **Note:**

You can ignore the missing index error in the readiness report. This is a known issue. The corresponding missing index is added during the schema upgrade operation. This error does not occur if the schema to be upgraded was created in 12c (12.2.1.4.0) by using the RCU.

Stopping Servers and Processes

Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.

An Oracle Fusion Middleware environment can consist of an Oracle WebLogic Server domain, an Administration Server, multiple managed servers, Java components, system components such as Identity Management components, and a database used as a repository for metadata. The components may be dependent on each other, so they must be stopped in the correct order.

 **Note:**

The procedures in this section describe how to stop the existing, pre-upgrade servers and processes using the WLST command-line utility or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See Starting and Stopping Administration and Managed Servers and Node Manager.

To stop your pre-upgrade Fusion Middleware environment, navigate to the pre-upgrade domain and follow the steps below:

Step 1: Stop System Components

To stop system components, such as Oracle HTTP Server, use the `stopComponent` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopComponent.sh component_name`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopComponent.cmd component_name`

You can stop system components in any order.

Step 2: Stop the Managed Servers

To stop a WebLogic Server Managed Server, use the `stopManagedWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopManagedWebLogic.sh`
`managed_server_name admin_url`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopManagedWebLogic.cmd`
`managed_server_name admin_url`

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

Stop any Oracle Identity Management components, such as Oracle Internet Directory:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopComponent.sh` `component_name`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopComponent.cmd` `component_name`

Step 4: Stop the Administration Server

When you stop the Administration Server, you also stop the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To stop the Administration Server, use the `stopWebLogic` script:

- (UNIX) `EXISTING_DOMAIN_HOME/bin/stopWebLogic.sh`
- (Windows) `EXISTING_DOMAIN_HOME\bin\stopWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Step 5: Stop Node Manager

To stop Node Manager, close the command shell in which it is running.

Alternatively, after setting the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down. See `stopNodeManager` in *WLST Command Reference for WebLogic Server*.

Upgrading Product Schemas (Applicable for Full-JRF Only)

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

The Upgrade Assistant allows you to upgrade individually selected schemas or all schemas associated with a domain. The option you select determines which Upgrade Assistant screens you will use.

- [Identifying Existing Schemas Available for Upgrade](#)
This optional task enables you to review the list of available schemas before you begin the upgrade by querying the schema version registry. The registry contains schema information such as version number, component name and ID, date of creation and modification, and custom prefix.

- [Starting the Upgrade Assistant](#)
Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).
- [Upgrading the Product Schemas Using the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.
- [Verifying the Schema Upgrade](#)
After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

Identifying Existing Schemas Available for Upgrade

This optional task enables you to review the list of available schemas before you begin the upgrade by querying the schema version registry. The registry contains schema information such as version number, component name and ID, date of creation and modification, and custom prefix.

You can let the Upgrade Assistant upgrade all of the schemas in the domain, or you can select individual schemas to upgrade. To help decide, follow these steps to view a list of all the schemas that are available for an upgrade:

1. If you are using an Oracle database, connect to the database by using an account that has Oracle DBA privileges, and run the following from SQL*Plus:

```
SET LINE 120
COLUMN MRC_NAME FORMAT A14
COLUMN COMP_ID FORMAT A20
COLUMN VERSION FORMAT A12
COLUMN STATUS FORMAT A9
COLUMN UPGRADED FORMAT A8
SELECT MRC_NAME, COMP_ID, OWNER, VERSION, STATUS, UPGRADED FROM
SCHEMA_VERSION_REGISTRY ORDER BY MRC_NAME, COMP_ID;
```

2. Examine the report that is generated.

If an upgrade is not needed for a schema, the `schema_version_registry` table retains the schema at its pre-upgrade version.

3. Note the schema prefix name that was used for your existing schemas. You will use the same prefix when you create new 12c (12.2.1.4.0) schemas.

 **Notes:**

- If your existing schemas are not from a supported version, then you must upgrade them to a supported version before using the 12c (12.2.1.4.0) upgrade procedures. Refer to your pre-upgrade version documentation for more information.
- Some components, such as Oracle Enterprise Data Quality, Oracle GoldenGate Monitor, and Oracle GoldenGate Veridata, support an upgrade from versions other than the standard Oracle Fusion Middleware supported versions.
- If you used an OID-based policy store in the earlier versions, make sure to create a new OPSS schema before you perform the upgrade. After the upgrade, the OPSS schema remains an LDAP-based store.
- You can only upgrade schemas for products that are available for upgrade in Oracle Fusion Middleware release 12c (12.2.1.4.0). Do not attempt to upgrade a domain that includes components that are not yet available for upgrade to 12c (12.2.1.4.0).

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).

To start the Upgrade Assistant:

 **Note:**

Before you start the Upgrade Assistant, make sure that the JVM character encoding is set to UTF-8 for the platform on which the Upgrade Assistant is running. If the character encoding is not set to UTF-8, then you will not be able to download files containing Unicode characters in their names. This can cause the upgrade to fail.

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant:
 - (UNIX) `./ua`
 - (Windows) `ua.bat`

For information about other parameters that you can specify on the command line, such as logging parameters, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 6-4 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
-readiness	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
-threads	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.
-response	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
-examine	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.
-logLevel <i>attribute</i>	Optional	Sets the logging level, specifying one of the following attributes: <ul style="list-style-type: none"> • TRACE • NOTIFICATION • WARNING • ERROR • INCIDENT_ERROR The default logging level is NOTIFICATION. Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.

Table 6-4 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logDir <i>location</i></code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Upgrading the Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

To upgrade the OPSS and IAU schemas:

1. On the Welcome screen, review the introduction to the Upgrade Assistant and information about important pre-upgrade tasks. Click **Next**.

Note:

For more information about any Upgrade Assistant screen, click **Help** on the screen.

2. On the Selected Schemas screen, select the following option:
 - **Individually Selected Schemas** if you want to select individual schemas for upgrade and you do not want to upgrade all of the schemas used by the domain.

 **Caution:**

Upgrade only those schemas that are used to support your 12c (12.2.1.4.0) components. Do not upgrade schemas that are currently being used to support components that are not included in Oracle Fusion Middleware 12c (12.2.1.4.0).

Click **Next**.

3. The Available Components screen provides a list of installed Oracle Fusion Middleware components that have schemas that can be upgraded. When you select a component, the schemas and any dependencies are automatically selected.

Select **Oracle Platform Security Services**. Selecting this automatically selects **Oracle Audit Services**. Click **Next**.

4. On the Domain Directory screen, enter the 12c WebLogic domain directory. Click **Browse** and use the navigation tree to select the 12c WebLogic domain directory.

The Upgrade Assistant requires the 12c domain location to access the `jps-config.xml` file.

5. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

 **Note:**

The Upgrade Assistant does not verify whether the prerequisites have been met.

6. On the Create Schemas screen, specify if you want the Upgrade Assistant to create the missing schemas. By default the **Create missing schemas for the specified domain** option is enabled. The Upgrade Assistant will attempt to create the missing schemas for the domain using the database connection details and schema owner name provided. The Upgrade Assistant creates the schemas using the default tablespace settings.

Select **Use same passwords for all schemas** if the same password is used for all schemas. Enter and confirm the password in the table. You only have to supply the password once.

 **Note:**

Do not allow the Upgrade Assistant to create schemas for you if you require customized options for your schemas. The schemas are created using the default Repository Creation Utility (RCU) settings. For example, if your schemas require additional tablespace, you must use the RCU to create the schemas.

If you do not want the Upgrade Assistant to create these schemas for you, deselect the **Create missing schemas for the specified domain** option and click **Next**. You will have to run the Repository Creation Utility to create the schemas.

7. The Create Schema Defaults screen appears if you selected the **Create missing schemas for the specified domain** option. The default datafile size is listed for each

component schema and auxiliary schema. If you need to modify the size of the tablespace datafile, or make any other changes to the default schema settings, use the Repository Creation Utility to create the schemas. You cannot modify the tablespace settings from the Upgrade Assistant.

8. On the Examine screen, review the status of the Upgrade Assistant as it examines each schema, verifying that the schema is ready for upgrade. If the status is **Examine finished**, click **Next**.

If the examine phase fails, Oracle recommends that you cancel the upgrade by clicking **No** in the Examination Failure dialog. Click **View Log** to see what caused the error and refer to Troubleshooting Your Upgrade in *Upgrading with the Upgrade Assistant* for information on resolving common upgrade errors.

 **Note:**

- If you resolve any issues detected during the examine phase without proceeding with the upgrade, you can start the Upgrade Assistant again without restoring from backup. However, if you proceed by clicking **Yes** in the Examination Failure dialog box, you need to restore your pre-upgrade environment from backup before you start the Upgrade Assistant again.
- Canceling the examination process has no effect on the schemas or configuration data; the only consequence is that the information the Upgrade Assistant has collected must be collected again in a future upgrade session.

9. The Upgrade Summary screen lists the schemas that will be upgraded and/or created.

Verify that the correct Source and Target Versions are listed for each schema that you intend to upgrade.

If you want to save these options to a response file to run the Upgrade Assistant again later in response (or silent) mode, click **Save Response File** and provide the location and the name of the response file. A silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again.

Click **Next**.

10. On the Upgrade Progress screen, monitor the status of the upgrade.

 **Caution:**

Allow the Upgrade Assistant enough time to perform the upgrade. Do not cancel the upgrade operation unless absolutely necessary. Doing so may result in an unstable environment.

If any schemas are not upgraded successfully, refer to the Upgrade Assistant log files for more information.

 **Note:**

The progress bar on this screen displays the progress of the current upgrade procedure. It does not indicate the time remaining for the upgrade.

Click **Next**.

11. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard.

If the upgrade fails: On the Upgrade Failure screen, click **View Log** to view and troubleshoot the errors. The logs are available at `NEW_ORACLE_HOME/oracle_common/upgrade/logs`.

 **Note:**

If the upgrade fails, you must restore your pre-upgrade environment from backup, fix the issues, then restart the Upgrade Assistant.

Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

If you are using an Oracle database, connect to the database as a user having Oracle DBA privileges, and run the following from SQL*Plus to get the current version numbers:

```
SET LINE 120
COLUMN MRC_NAME FORMAT A14
COLUMN COMP_ID FORMAT A20
COLUMN VERSION FORMAT A12
COLUMN STATUS FORMAT A9
COLUMN UPGRADED FORMAT A8
SELECT MRC_NAME, COMP_ID, OWNER, VERSION, STATUS, UPGRADED FROM
SCHEMA_VERSION_REGISTRY ORDER BY MRC_NAME, COMP_ID ;
```

In the query result:

- Check that the number in the `VERSION` column matches the latest version number for that schema. For example, verify that the schema version number is 12.2.1.4.0.

 **Note:**

However, that not all schema versions will be updated. Some schemas do not require an upgrade to this release and will retain their pre-upgrade version number.

- The `STATUS` field will be either `UPGRADING` or `UPGRADED` during the schema patching operation, and will become `VALID` when the operation is completed.
- If the status appears as `INVALID`, the schema update failed. You should examine the logs files to determine the reason for the failure.

- Synonym objects owned by `IAU_APPEND` and `IAU_VIEWER` will appear as `INVALID`, but that does not indicate a failure.

They become invalid because the target object changes after the creation of the synonym. The synonyms objects will become valid when they are accessed. You can safely ignore these `INVALID` objects.

About Reconfiguring the Domain

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1.4.0).

When you reconfigure a standalone domain, the following items are automatically updated, depending on the applications in the domain:

- WebLogic Server core infrastructure
- Domain version

Note:

The Reconfiguration Wizard does not update any of your own applications that are included in the domain.

Specifically, when you reconfigure a domain, the following occurs:

- The domain version number in the `config.xml` file for the domain is updated to the version of the target Oracle HTTP Server.
- The reconfiguration template for Oracle HTTP Server is automatically selected and applied to the domain. This template defines any reconfiguration tasks that are required to make the standalone domain compatible with the target Oracle HTTP Server version.
- The start scripts are updated.

If you want to preserve your modified start scripts, be sure to back them up before starting the Reconfiguration Wizard.

Note:

When the domain reconfiguration process starts, you can't undo the changes that it makes. Before running the Reconfiguration Wizard, ensure that you have backed up the domain as covered in the pre-upgrade checklist. If an error or other interruption occurs while running the Reconfiguration Wizard, you must restore the domain by copying the files and directories from the backup location to the original domain directory. This is the only way to ensure that the domain has been returned to its original state before reconfiguration.

Follow these instructions to reconfigure the existing domain using the Reconfiguration Wizard. See *Reconfiguring WebLogic Domains in [Upgrading Oracle WebLogic Server](#)*.

- [Backing Up the Domain](#)

- [Starting the Reconfiguration Wizard](#)
- [Reconfiguring the Domain with the Reconfiguration Wizard](#)
Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain.

Backing Up the Domain

Before running the Reconfiguration Wizard, create a backup copy of the domain directory.

To create a backup of the domain directory:

1. Copy the source domain to a separate location to preserve the contents.
(Windows) `copy C:\domains\mydomain to C:\domains\mydomain_backup.`
(UNIX) `cp mydomain /domains/mydomain_backup`
2. Before updating the domain on each remote Managed Server, create a backup copy of the domain directory on each remote machine.
3. Verify that the backed up versions of the domain are complete.

If domain reconfiguration fails for any reason, you must copy all files and directories from the backup directory into the original domain directory to ensure that the domain is returned entirely to its original state before reconfiguration.

Starting the Reconfiguration Wizard



Note:

Shut down the administration server and all collocated managed servers before starting the reconfiguration process. See [Stopping Servers and Processes](#) .

To start the Reconfiguration Wizard in graphical mode:

1. Sign in to the system on which the domain resides.
2. Open the command shell (on UNIX operating systems) or open a command prompt window (on Windows operating systems).
3. Go to the `oracle_common/common/bin` directory:
 - (UNIX) `NEW_ORACLE_HOME/oracle_common/common/bin`
 - (Windows) `NEW_ORACLE_HOME\oracle_common\commom\bin`
4. Start the Reconfiguration Wizard with the following logging options:
 - (UNIX) `./reconfig.sh -log=log_file -log_priority=ALL`
 - (Windows) `reconfig.cmd -log=log_file -log_priority=ALL`

where `log_file` is the absolute path of the log file you'd like to create for the domain reconfiguration session. This can be helpful if you need to troubleshoot the reconfiguration process.

The parameter `-log_priority=ALL` ensures that logs are logged in fine mode.

 **Note:**

When you run this command, the following error message might appear to indicate that the default cache directory is not valid:

```
*sys-package-mgr*: can't create package cache dir
```

You can change the cache directory by setting the environment variable `CONFIG_JVM_ARGS`. For example:

```
CONFIG_JVM_ARGS=-Dpython.cachedir=valid_directory
```

Reconfiguring the Domain with the Reconfiguration Wizard

Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain.

To reconfigure the domain:

1. On the Select Domain screen, specify the location of the domain you want to upgrade or click **Browse** to navigate and select the domain directory. Click **Next**.
2. On the Reconfiguration Setup Progress screen, view the progress of the setup process. When complete, click **Next**.

During this process:

- The reconfiguration templates for your installed products, including Fusion Middleware products, are automatically applied. This updates various domain configuration files such as `config.xml`, `config-groups.xml`, and `security.xml` (among others).
 - Schemas, scripts, and other such files that support your Fusion Middleware products are updated.
 - The domain upgrade is validated.
3. On the Domain Mode and JDK screen, select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use. The supported JDK version for 12c (12.2.1.4.0) is 1.8.0_211 and later. Click **Next**.

 **Note:**

You cannot change the **Domain Mode** at this stage.

For a list of JDKs that are supported for a specific platform, see Oracle Fusion Middleware Supported System Configurations.

4. On the JDBC Data Sources screen, configure the JDBC data sources defined in your domain source.

The JDBC data sources associated with the products for which you are creating the domain are listed in the lower half of the screen. A JDBC data source contains a pool of database connections that are created when the data source instance is created, deployed or targeted, or at server startup. Applications look up a data source on the JNDI tree, and then request a connection. When the applications no

longer need the connections, they return the connections to the connection pool in the data source.

From the **Data Source Name** drop-down list, select the data source(s) for which you want to specify the settings. The values that you specify are displayed in the appropriate columns in the data source list, for the selected data source.

For Oracle RAC Configuration for data sources, you can select one of the three options:

- Convert to GridLink
- Convert to RAC multi data source
- Don't convert

For more information about each option, click **Help**.

After specifying the details, click **Next**.

If you do not select any data sources on the JDBC Data Sources screen, the following warning displays:

```
Missing Driver
```

```
Click Ok to proceed without verification, click Cancel to return to the JDBC Data Sources page.
```

In this case, if you click **Ok**, the data sources are not verified.

5. On the JDBC Data Sources Test screen, select the check box for the data source connection you configured on the JDBC Data Sources screen and click **Test Selected Connections** to test the data source connection.

 **Note:**

To test the database connections, the database to which you are connecting must be running. If you do not want to test the connections at this time, do not select any data sources. Click **Next** to continue.

6. On the Database Configuration Type screen, select **RCU Data** to connect to the Server Table (`_STB`) schema.

Enter the database connection details using the RCU service table (`_STB`) schema credentials and click **Get RCU Configuration**.

The Reconfiguration Wizard uses this connection to automatically configure the data sources required for components in your domain.

 **Note:**

By default **Oracle's Driver (Thin) for Service connections; Versions: Any** is the selected driver. If you specified an instance name in your connection details — instead of the service name — you must select **Oracle's Driver (Thin) for pooled instance connections; Versions: Any**. If you do not change the driver type, then the connection will fail.

 **Note:**

For any existing 12c (12.2.1.3.0) datasource, the reconfiguration will preserve the existing values. For new datasources where the schema was created for 12c (12.2.1.4.0) by the RCU, the default connection data will be retrieved from the `_STB` schema. If no connection data for a given schema is found in the `_STB` schema, then the default connection data is used.

If the check is successful, click **Next**. If the check fails, reenter the connection details correctly and try again.

 **Note:**

If you are upgrading from 12c (12.2.1.3.0), and your database has `_OPSS` or `_IAU` 12c (12.2.1.3.0) database schemas, you must manually enter database connection details for those schemas. These schemas were not required in 12c and had to be created manually. Users could assign any name to these schemas, therefore the Reconfiguration Wizard does not recognize them. When providing connection information for `_IAU`, use the `IAU_APPEND` user information.

7. On the JDBC Component Schema Test screen, select all the component schemas and click **Test Selected Connections** to test the connection for each schema. The result of the test is indicated in the Status column.

When the check is complete, click **Next**.

8. The Node Manager screen is only displayed if the domain you are reconfiguring is currently using a per host Node Manager.

On the Node Manager screen, select the Node Manager configuration to use for the reconfigured domain. The resulting configuration depends on the combination of options you select for **Node Manager Type** and **Node Manager Configuration**.

Table 6-5 Field Descriptions for Node Manager Screen

Option	Description
Per Domain Default Location	If you select this option, the Node Manager home is redefined to <code>DOMAIN_NAME/nodemanager</code> and you cannot edit the Node Manager home.
Per Domain Custom Location	Select this option if you want the per domain Node Manager configuration files to be created in a specific location for this domain. Specify the directory in the Node Manager Home field, or click Browse to use the navigation tree to select the location. The specified directory must be empty. The <code>nodemanager.properties</code> and <code>nodemanager.domains</code> files are created in this directory.

Table 6-5 (Cont.) Field Descriptions for Node Manager Screen

Option	Description
Node Manager Home	If you selected the Per Domain Custom Location option, click Browse to navigate to the directory location that you want to use to store the per domain Node Manager configuration.
Manual Node Manager Setup	<p>If you select this option, creation of the Node Manager configuration for the domain is skipped (all remaining fields cannot be modified), and if you want to use Node Manager in the domain, you must manually configure Node Manager as described in Completing the Node Manager Configuration. The reconfigured domain will still use a per host Node Manager configuration.</p> <p>You should also select this option if your existing domain is not configured to use Node Manager and you do not want to use Node Manager in the reconfigured domain.</p> <p>For more information about Node Manager configuration, see Administering Node Manager for Oracle WebLogic Server.</p>
Node Manager Configuration	Select one of the following two options. These fields are not available if you selected Manual Node Manager Setup .
Create New Configuration	A per domain Node Manager configuration will be automatically created for the reconfigured domain using default settings in <code>nodemanager.properties</code> . If necessary, you can modify <code>nodemanager.properties</code> after the domain has been successfully reconfigured.
Migrate Existing Configuration	The existing per host Node Manager configuration will be migrated to a per domain configuration for the reconfigured domain. This does not include environment-specific settings for <code>ListenAddress</code> , <code>ListenPort</code> , <code>StartScriptName</code> , <code>JavaHome</code> , and <code>LogFile</code> .
Node Manager Home	If you selected the Migrate Existing Configuration option, enter or browse to the Node Manager home directory that you want to migrate to the reconfigured domain.

Table 6-5 (Cont.) Field Descriptions for Node Manager Screen

Option	Description
Apply Oracle Recommended Defaults	<p>If you selected the Migrate Existing Configuration option, select this check box if you want to use Oracle-recommended defaults in the <code>nodemanager.properties</code> file. Deselect this check box if you want to continue using the settings in the <code>nodemanager.properties</code> file being migrated.</p> <p>Oracle-recommended properties with default values are as follows:</p> <pre> LogLimit=0 AuthenticationEnabled=true LogLevel=INFO DomainsFileEnabled=true NativeVersionEnabled=true LogToStderr=true SecureListener=true LogCount=1 StopScriptEnabled=false QuitEnabled=false LogAppend=true StateCheckInterval=500 CrashRecoveryEnabled=false StartScriptEnabled=true LogFormatter=weblogic.nodemanager.server.LogFormatter ListenBacklog=50 </pre>
Node Manager Credentials: Username, Password	Specify the username and password that you want to use to start Node Manager in the reconfigured domain.

- On the Advanced Configuration screen, you can select all categories for which you want to perform advanced configuration. For each category you select, the appropriate configuration screen is displayed to allow you to perform advanced configuration.

 **Note:**

The categories that are listed on the Advanced Configuration screen depend on the resources defined in the templates you selected for the domain.

For this upgrade, select none of the options and click **Next**.

- On the Configuration Summary screen, review the detailed configuration settings of the domain before continuing.

You can limit the items that are displayed in the right-most panel by selecting a filter option from the **View** drop-down list.

To change the configuration, click **Back** to return to the appropriate screen. To reconfigure the domain, click **Reconfig**.

 **Note:**

The location of the domain does not change when you reconfigure it.

11. The Reconfiguration Progress screen displays the progress of the reconfiguration process.

During this process:

- Domain information is extracted, saved, and updated.
- Schemas, scripts, and other such files that support your Fusion Middleware products are updated.

When the progress bar shows 100%, click **Next**.

12. The End of Configuration screen indicates whether the reconfiguration process completed successfully or failed. It also displays the location of the domain that was reconfigured as well as the Administration Server URL (including the listen port). If the reconfiguration is successful, it displays **Oracle WebLogic Server Reconfiguration Succeeded**.

If the reconfiguration process did not complete successfully, an error message is displayed indicates the reason. Take appropriate action to resolve the issue. If you cannot resolve the issue, contact My Oracle Support.

Note the Domain Location and the Admin Server URL for further operations.

Upgrading Domain Component Configurations

After reconfiguring the domain, use the Upgrade Assistant to upgrade the domain *component* configurations inside the domain to match the updated domain configuration.

- [Starting the Upgrade Assistant](#)
Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).
- [Upgrading Domain Components Using the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to upgrade component configurations in the WebLogic domain.
- [Verifying the Domain-Specific-Component Configurations Upgrade](#)
To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.4.0).

To start the Upgrade Assistant:

 **Note:**

Before you start the Upgrade Assistant, make sure that the JVM character encoding is set to UTF-8 for the platform on which the Upgrade Assistant is running. If the character encoding is not set to UTF-8, then you will not be able to download files containing Unicode characters in their names. This can cause the upgrade to fail.

1. Go to the `oracle_common/upgrade/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/upgrade/bin`
 - (Windows) `ORACLE_HOME\oracle_common\upgrade\bin`
2. Start the Upgrade Assistant:
 - (UNIX) `./ua`
 - (Windows) `ua.bat`

For information about other parameters that you can specify on the command line, such as logging parameters, see:

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

When you start the Upgrade Assistant from the command line, you can specify additional parameters.

Table 6-6 Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks Note: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server).	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.

Table 6-6 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-response</code>	Required for silent upgrades or silent readiness checks	Runs the Upgrade Assistant using inputs saved to a response file generated from the data that is entered when the Upgrade Assistant is run in GUI mode. Using this parameter runs the Upgrade Assistant in <i>silent mode</i> (without displaying Upgrade Assistant screens).
<code>-examine</code>	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the <code>-readiness</code> parameter.
<code>-logLevel attribute</code>	Optional	<p>Sets the logging level, specifying one of the following attributes:</p> <ul style="list-style-type: none">• TRACE• NOTIFICATION• WARNING• ERROR• INCIDENT_ERROR <p>The default logging level is NOTIFICATION.</p> <p>Consider setting the <code>-logLevel TRACE</code> attribute to so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.</p>

Table 6-6 (Cont.) Upgrade Assistant Command-Line Parameters

Parameter	Required or Optional	Description
<code>-logDir <i>location</i></code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant creates log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX)</p> <pre>ORACLE_HOME/ oracle_common/upgrade/ logs ORACLE_HOME/ oracle_common/upgrade/ temp</pre> <p>(Windows)</p> <pre>ORACLE_HOME\oracle_commo n\upgrade\logs ORACLE_HOME\oracle_commo n\upgrade\temp</pre>
<code>-help</code>	Optional	Displays all of the command-line options.

Upgrading Domain Components Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade component configurations in the WebLogic domain.

After running the Reconfiguration Wizard to reconfigure the WebLogic domain to 12c (12.2.1.4.0), you must run the Upgrade Assistant to upgrade the domain *component* configurations to match the updated domain configuration.

To upgrade domain component configurations with the Upgrade Assistant:

1. On the Welcome screen, review an introduction to the Upgrade Assistant and information about important pre-upgrade tasks. Click **Next**.

Note:

For more information about any Upgrade Assistant screen, click **Help** on the screen.

2. On the WebLogic Components screen, select the **WebLogic Component Configurations** option to upgrade component configurations for a managed WebLogic Server domain. Enter the connection details required to connect to the WebLogic Administration Server that is managing the domain and click **Next**.
3. On the Component List screen, verify that the list includes all the components for which you want to upgrade configurations and click **Next**.

If you do not see the components you want to upgrade, click **Back** to go to the previous screen and specify a different domain.

4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

 **Note:**

The Upgrade Assistant does not verify whether the prerequisites have been met.

5. On the Examine screen, review the status of the Upgrade Assistant as it examines each component, verifying that the component configuration is ready for upgrade. If the status is **Examine finished**, click **Next**.

If the examine phase fails, Oracle recommends that you cancel the upgrade by clicking **No** in the Examination Failure dialog. Click **View Log** to see what caused the error and refer to Troubleshooting Your Upgrade in *Upgrading with the Upgrade Assistant* for information on resolving common upgrade errors.

 **Note:**

- If you resolve any issues detected during the examine phase without proceeding with the upgrade, you can start the Upgrade Assistant again without restoring from backup. However, if you proceed by clicking **Yes** in the Examination Failure dialog box, you need to restore your pre-upgrade environment from backup before starting the Upgrade Assistant again.
- Canceling the examination process has no effect on the configuration data; the only consequence is that the information the Upgrade Assistant has collected must be collected again in a future upgrade session.

6. On the Upgrade Summary screen, review the summary of the options you have selected for component configuration upgrade.

The response file collects and stores all the information that you have entered, and enables you to perform a silent upgrade at a later time. The silent upgrade performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file.

Click **Upgrade** to start the upgrade process.

7. On the Upgrade Progress screen, monitor the status of the upgrade.

 **Caution:**

Allow the Upgrade Assistant enough time to perform the upgrade. Do not cancel the upgrade operation unless absolutely necessary. Doing so may result in an unstable environment.

If any components are not upgraded successfully, refer to the Upgrade Assistant log files for more information.

 **Note:**

The progress bar on this screen displays the progress of the current upgrade procedure. It does not indicate the time remaining for the upgrade.

Click **Next**.

8. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard. The Post-Upgrade Actions window describes the manual tasks you must perform to make components functional in the new installation. This window appears only if a component has post-upgrade steps.

If the upgrade fails: On the Upgrade Failure screen, click **View Log** to view and troubleshoot the errors. The logs are available at `ORACLE_HOME/oracle_common/upgrade/logs`.

 **Note:**

If the upgrade fails you must restore your pre-upgrade environment from backup, fix the issues, then restart the Upgrade Assistant.

Verifying the Domain-Specific-Component Configurations Upgrade

To verify that the domain-specific-component configurations upgrade was successful, sign in to the Administration console and the Oracle Enterprise Manager Fusion Middleware Control and verify that the version numbers for each component is 12.2.1.4.0.

To sign in to the Administration Console, go to: `http://administration_server_host:administration_server_port/console`

To sign in to Oracle Enterprise Manager Fusion Middleware Control Console, go to: `http://administration_server_host:administration_server_port/em`

 **Note:**

After upgrade, make sure you run the administration tools from the new 12c (12.2.1.4.0) Oracle home directory and not from the previous Oracle home directory.

During the upgrade process, some OWSM documents, including policy sets and predefined documents such as policies and assertion templates, may need to be upgraded. If a policy set or a predefined document is upgraded, its version number is incremented by 1.

If you created the FMW user to run the Upgrade Assistant, ensure that you delete the account after verifying your upgrade was successful.

Starting Servers and Processes

After a successful upgrade, restart all processes and servers, including the Administration Server and any Managed Servers.

The components may be dependent on each other so they must be started in the correct order.

 **Note:**

The procedures in this section describe how to start servers and process using the WLST command line or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See *Starting and Stopping Administration and Managed Servers and Node Manager in Administering Oracle Fusion Middleware*.

To start your Fusion Middleware environment, follow the steps below:

Step 1: Start the Administration Server

When you start the Administration Server, you also start the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To start the Administration Server, use the `startWebLogic` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startWebLogic.sh`
- (Windows) `NEW_DOMAIN_HOME\bin\startWebLogic.cmd`

When prompted, enter your user name, password, and the URL of the Administration Server.

Step 2: Start Node Manager

To start Node Manager, use the `startNodeManager` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startNodeManager.sh`
- (Windows) `NEW_DOMAIN_HOME\bin\startNodeManager.cmd`

Step 3: Start Oracle Identity Management Components

Start any Oracle Identity Management components, such as Oracle Internet Directory, that form part of your environment:

- (UNIX) `NEW_DOMAIN_HOME/bin/startComponent.sh component_name`
- (Windows) `NEW_DOMAIN_HOME\bin\startComponent.cmd component_name`

Step 4: Start the Managed Servers

To start a WebLogic Server Managed Server, use the `startManagedWebLogic` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url`
- (Windows) `NEW_DOMAIN_HOME\bin\startManagedWebLogic.cmd managed_server_name admin_url`

When prompted, enter your user name and password.

 **Note:**

The startup of a Managed Server will typically start the applications that are deployed to it. Therefore, it should not be necessary to manually start applications after the Managed Server startup.

Step 5: Start System Components

To start system components, such as Oracle HTTP Server, use the `startComponent` script:

- (UNIX) `NEW_DOMAIN_HOME/bin/startComponent.sh component_name`
- (Windows) `NEW_DOMAIN_HOME\bin\startComponent.cmd component_name`

You can start system components in any order.

- [Starting the Oracle HTTP Server](#)
You can use Enterprise Manager Fusion Middleware Control to start, stop, and restart Oracle HTTP Server.

Starting the Oracle HTTP Server

You can use Enterprise Manager Fusion Middleware Control to start, stop, and restart Oracle HTTP Server.

You can start the Oracle HTTP Server with the Node Manager by following the procedure mentioned in Running Oracle HTTP Server Remotely in *Administering Oracle HTTP Server*.

To start an Oracle HTTP Server using Enterprise Manager Fusion Middleware Control:

1. Go to the Oracle HTTP Server home page.
2. From the Oracle HTTP Server menu:
 - a. Select **Control**.

- b. Select **Start Up** from the **Control** menu.
 - Or
3. From the Target Navigation tree:
 - a. Right-click the Oracle HTTP Server instance you want to start.
 - b. Select **Control**.
 - c. Select **Start Up** from the **Control** menu.

Verifying the Upgrade

If you can log in to the Administration Console, the Enterprise Manager Fusion Middleware Control, and the OHS Home Page, then your upgrade is successful.

To verify the upgrade:

1. Access the OHS default home page by using the following URL:
`http://MachineName:7777`
2. Access Enterprise Manager Fusion Middleware Control by using the following URL:
`http://MachineName:7001/em`

Post-Upgrade Task

Perform the following update after upgrading the Oracle HTTP Server.

Updating to 19c Database Client:

Install the new patches to update the OHS Database client and OSS patches to use 19c Database client libraries. For more information about upgrading the new Database client 19c for Oracle Fusion Middleware 12.2.1.4.0, see [Doc ID 2921245.1](#).

A

Replacing Certificate Signed Using MD5 Algorithm with Certificate Signed Using SHA-2 Algorithm

Oracle strongly recommends that you refrain from using a certificate signed with Message Digest 5 Algorithm (MD5), because the security of MD5 algorithm has been compromised. Therefore, you must replace the certificate signed using MD5 algorithm with a certificate signed with Secure Hashing Algorithm 2 (SHA-2). By default, certificates signed using MD5 algorithm are no longer supported in Oracle HTTP Server.

- [How to Check whether Certificate Signed with MD5 Algorithm is Present in the Wallet?](#)
You can use the `orapki` utility to display whether your wallet contains a certificate signed with MD5 algorithm.
- [Removing Certificate Signed with MD5 Algorithm from the Wallet](#)
If MDS is present in your wallet, the signature algorithm name is displayed as `MDSwithRSA`. You must replace this certificate with certificate signed using SHA2 algorithm.
- [Adding Certificate Signed with SHA-2 Algorithm to the Wallet](#)
If you are using CA-signed user certificate that is signed with MD5 algorithm, contact your certificate authority to get a new user certificate signed with SHA-2 algorithm and import it in to the wallet.
- [Enabling Support for Certificate Signed with MD5 Algorithm in your 12.2.1.4.0 Oracle HTTP Server Deployment](#)
By default, support of certificate signed with MD5 algorithm has been removed because the security of MD5 algorithm is severely compromised. If you still want to use certificate signed using MD5 algorithm, you can enable the support for a certificate signed with MD5 algorithm by following the procedure in this section. However, enabling support for certificates signed using MD5 algorithm is not recommended.

How to Check whether Certificate Signed with MD5 Algorithm is Present in the Wallet?

You can use the `orapki` utility to display whether your wallet contains a certificate signed with MD5 algorithm.

To check whether the MD5 certificate is present in your wallet:

1. Locate `orapki` and the `keytool` path:
(`orapki`) `ORACLE_HOME/oracle_common/bin/orapki`
(`Keytool`) `ORACLE_HOME/oracle_common/jdk/jre/bin`
2. Set the `JAVA_HOME` environment variable as follows:
`export JAVA_HOME=ORACLE_HOME/oracle_common/jdk/jre/`
3. Enter the following command to display the contents of the wallet:

```
orapki wallet display -wallet wallet_location
```

Sample command:

```
ORACLE_HOME/oracle_common/bin/orapki wallet display -
wallet /scratch/ohs12.2.1.x_install/walletohs2/
```

Output:

```
Oracle PKI Tool : Version 12.2.0.1.0
Copyright (c) 2004, 2016, Oracle and/or its affiliates. All rights
reserved.
```

```
Requested Certificates:
```

```
User Certificates:
```

```
Subject: CN=www.xyx.com,C=IN
```

```
Trusted Certificates:
```

```
Subject: CN=www.xyx.com,C=IN
```

4. Identify the domain name (DN) of the certificate present in the wallet. In this example, Subject of User Certificates is the DN of the certificate present in the wallet; which is CN=www.xyx.com,C=IN.
5. Export the certificate present in the wallet as shown in the following example:

```
orapki wallet export -wallet wallet_Location -dn 'DN_string'
-cert certificate_file
```

Sample command:

```
ORACLE_HOME/oracle_common/bin/orapki wallet export -wallet /
scratch/ohs12.2.1.x_install/walletohs2/ -
dn 'CN=www.xyx.com,C=IN' -cert wallet.cert
```

6. Use the keytool to check the signature algorithm used to sign the *certificate_file* that you exported in the previous step by entering the following command:

```
ORACLE_HOME/oracle_common/jdk/jre/bin/keytool -printcert -
file certificate_file
```

Sample command:

```
/scratch/ohs12.2.1.x_install/oracle_common/jdk/jre/bin/
keytool -printcert -file wallet.cert
```

Output:

```
Owner: CN="Self-Signed Certificate for ohs3 ", OU=OAS, O=ORACLE,
L=REDWOODSHORES, ST=CA, C=US
Issuer: CN="Self-Signed Certificate for ohs3 ", OU=OAS, O=ORACLE,
L=REDWOODSHORES, ST=CA, C=US
Serial number: cd7081c47adb9ff867da01e3fe383e0f
Valid from: Fri Jul 01 04:21:40 PDT 2016 until: Sat Jun 19 04:21:40
PDT 2066
Certificate fingerprints:
MD5: 89:4D:C4:B4:28:7E:D4:0A:93:CA:E4:97:0C:F0:CE:86
SHA1: 53:FB:D6:58:0D:09:85:E9:6D:55:E2:92:E7:87:08:B8:65:92:15:9E
Signature algorithm name: MD5withRSA
Version: 1
```

Signature algorithm name: MD5withRSA implies that MD5 algorithm is being used to sign the certificate present in the wallet.

Removing Certificate Signed with MD5 Algorithm from the Wallet

If MDS is present in your wallet, the signature algorithm name is displayed as `MDSwithRSA`. You must replace this certificate with certificate signed using SHA2 algorithm.

To remove a certificate signed with MD5 algorithm from your wallet:

1. Enter the following command to remove a user certificate signed using MD5 algorithm:

```
orapki wallet remove -wallet wallet_location -dn 'DN_string' -  
user_cert -auto_login_only
```

Sample command:

```
/scratch/ohs12.2.1.x_install/oracle_common/bin/orapki wallet  
remove -wallet /scratch/ohs12.2.1.x_install/walletohs2/ -  
dn 'CN=www.xyx.com,C=IN' -user_cert -auto_login_only
```

Output:

```
Oracle PKI Tool : Version 12.2.0.1.0  
Copyright (c) 2004, 2016, Oracle and/or its affiliates. All rights  
reserved.
```

2. If the user certificate is a self-signed certificate, you need to remove it from the *trusted certificate* and the *requested certificate* list by entering the following commands:

```
orapki wallet remove -wallet wallet_location -dn 'DN_string'  
-trusted_cert -auto_login_only
```

```
orapki wallet remove -wallet wallet_location -dn 'DN_string'  
-cert_req -auto_login_only
```

Sample command:

```
/scratch/ohs12.2.1.x_install/oracle_common/bin/orapki wallet  
remove -wallet /scratch/ohs12.2.1.x_install/walletohs2/ -dn  
'CN=www.xyx.com,C=IN' -trusted_cert -auto_login_only
```

Output:

```
Oracle PKI Tool : Version 12.2.0.1.0  
Copyright (c) 2004, 2016, Oracle and/or its affiliates. All rights  
reserved.
```

3. If a wallet has trusted certificate or a certificate request which is signed using MD5 algorithm, remove that certificate by entering the following commands:

```
orapki wallet remove -wallet wallet_location -dn 'DN_string' -  
trusted_cert -auto_login_only remove cert request
```

```
orapki wallet remove -wallet wallet_location -dn 'DN_string' -  
cert_req -auto_login_only
```

Adding Certificate Signed with SHA-2 Algorithm to the Wallet

If you are using CA-signed user certificate that is signed with MD5 algorithm, contact your certificate authority to get a new user certificate signed with SHA-2 algorithm and import it in to the wallet.

To add self-signed certificate, signed using SHA-2 algorithm:

1. If certificate is self-signed, enter the following command to add self-signed certificate signed using SHA-2 algorithm:

```
orapki wallet add -wallet wallet_Location -dn 'DN_String' -  
keysize 2048 -sign_alg sha256 -self_signed -validity 9125 -  
auto_login_only
```

Sample command:

```
/scratch/ohs12.2.1.x_install/oracle_common/bin/orapki wallet  
add -wallet /scratch/ohs12.2.1.x_install/walletohs2/ -  
dn 'CN=www.yyx.com,C=IN' -keysize 2048 -sign_alg sha256 -  
self_signed -validity 9125 -auto_login_only
```

2. Add trusted certificate signed using SHA-2 algorithm to wallet by entering the following command:

```
orapki -wallet add -wallet wallet_location -trusted_cert -  
cert certificate_file -auto_login_only
```

3. Add user certificate signed using SHA-2 algorithm to wallet by entering the following command:

```
orapki -wallet add -wallet wallet_location -user_cert -cert  
certificate_file -auto_login_only
```

The operation is successful if you see signature algorithm name equal to *SHA256withRSA* when you print your certificate file. Following is a sample output:

```
Owner: CN=www.yyx.com, C=IN  
Issuer: CN=www.yyx.com, C=IN  
Serial number: f689ec6986c70f973138962eb2f0e5f9  
Valid from: Wed May 11 04:01:24 PDT 2016 until: Sat Oct 27 04:01:24  
PDT 2018  
Certificate fingerprints:  
MD5: D7:0F:CB:00:A7:04:33:DA:2F:8A:AD:C9:2A:9E:DF:26  
SHA1: D4:6C:51:DB:53:B5:F5:C7:60:8D:8B:95:68:E6:B0:5E:E8:00:ED:DF  
SHA256:  
B1:EF:73:98:EA:6A:1A:60:FF:1F:10:89:8C:B8:16:63:71:03:1B:6E:38:D1:2D:AE  
:E9:BD:3E:13:BE:AF:A0:76  
Signature algorithm name: SHA256withRSA  
Version: 1
```

Enabling Support for Certificate Signed with MD5 Algorithm in your 12.2.1.4.0 Oracle HTTP Server Deployment

By default, support of certificate signed with MD5 algorithm has been removed because the security of MD5 algorithm is severely compromised. If you still want to use certificate signed using MD5 algorithm, you can enable the support for a certificate signed with MD5 algorithm by following the procedure in this section. However, enabling support for certificates signed using MD5 algorithm is not recommended.

To enable support for a certificate signed with MD5 algorithm:

1. For Standalone Oracle HTTP Server deployment:

- a.** Stop the Oracle HTTP Server instance and the Node Manager.
- b.** Change to the following staging directory:

```
(UNIX) ORACLE_HOME/user_projects/domains/base_domain/config/  
fmwconfig/components/OHS/ohs1/
```

(Windows)

```
ORACLE_HOME\user_projects\domains\base_domain\config\fmwconfig  
\components\OHS\ohs1\
```

- c.** Open the `ohs.plugins.nodemanager.properties` file in edit mode and add the following line:

```
environment.ORACLE_SSL_ALLOW_MD5_CERT_SIGNATURES = 1
```

- d.** Restart the Node Manager and the Oracle HTTP Server instance.

2. For Managed Oracle HTTP Server deployment:

- a.** Stop the Oracle HTTP Server instance, Node Manager, and the WebLogic Server.
- b.** Change to the following staging directory:

```
(UNIX) ORACLE_HOME/user_projects/domains/base_domain/config/  
fmwconfig/components/OHS/ohs1/
```

(Windows)

```
ORACLE_HOME\user_projects\domains\base_domain\config\fmwconfig  
\components\OHS\ohs1\
```

- c.** Open the `ohs.plugins.nodemanager.properties` file in edit mode and add the following line:

```
environment.ORACLE_SSL_ALLOW_MD5_CERT_SIGNATURES = 1
```

- d.** Restart the WebLogic Server, Node Manager, and the Oracle HTTP Server instance.

You must perform this procedure for each Oracle HTTP Server instance.

B

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 8u191. 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_211`.

- 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 Oracle 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 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 `1.8.0_191`.

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

1. 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.

2. 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_211`.

C

About the Reduced Downtime Upgrade Process

Review the flowchart and roadmap for an overview of the upgrade process for Oracle Fusion Middleware if you are performing a reduced downtime upgrade.

A reduced downtime upgrade process is different from a standard Fusion Middleware upgrade process. This type of upgrade requires at least a two-node cluster environment and is performed in a rolling fashion so that one node is always up to accomplish the reduced downtime upgrade. In the standard upgrade flow, all servers and processes are stopped prior to initiating the upgrade.

Before you begin the reduced downtime upgrade:

- Back up the database schemas.
- Back up the domain directory and the application directory.
- Back up the UI /similar customizations.

For details about backing up, see [Creating a Complete Backup](#).

After taking the required back ups, uninstall the previous version of the software to ensure that an empty Oracle Home is available to install the new product distributions. This is the key difference between the two upgrade processes. In addition, if your product requires a schema and/or a config upgrade, you will need to run the Upgrade Assistant twice, once each for schema upgrade and for config upgrade.

Note:

For a managed/collocated Oracle HTTP Server, if the WebLogic Server and OHS binaries are installed in the same Oracle Home, ensure that this directory is empty before you install the new binaries in the same directory.

Note:

Oracle Fusion Middleware 12c (12.2.1.3) is the only supported starting point for a reduced downtime upgrade. You cannot perform a reduced downtime upgrade if you are upgrading from a supported Fusion Middleware 11g or 12c (12.2.1.2.0 or earlier) release, or if you do not have a multi-node environment. To perform a reduced downtime upgrade from 12c (12.1.3 or 12.2.1.2), you must first upgrade to 12.2.1.3 using the standard upgrade process. See [Upgrading Oracle HTTP Server from a Previous 12c Release](#).

Figure C-1 Process Flowchart for a Reduced Downtime Upgrade

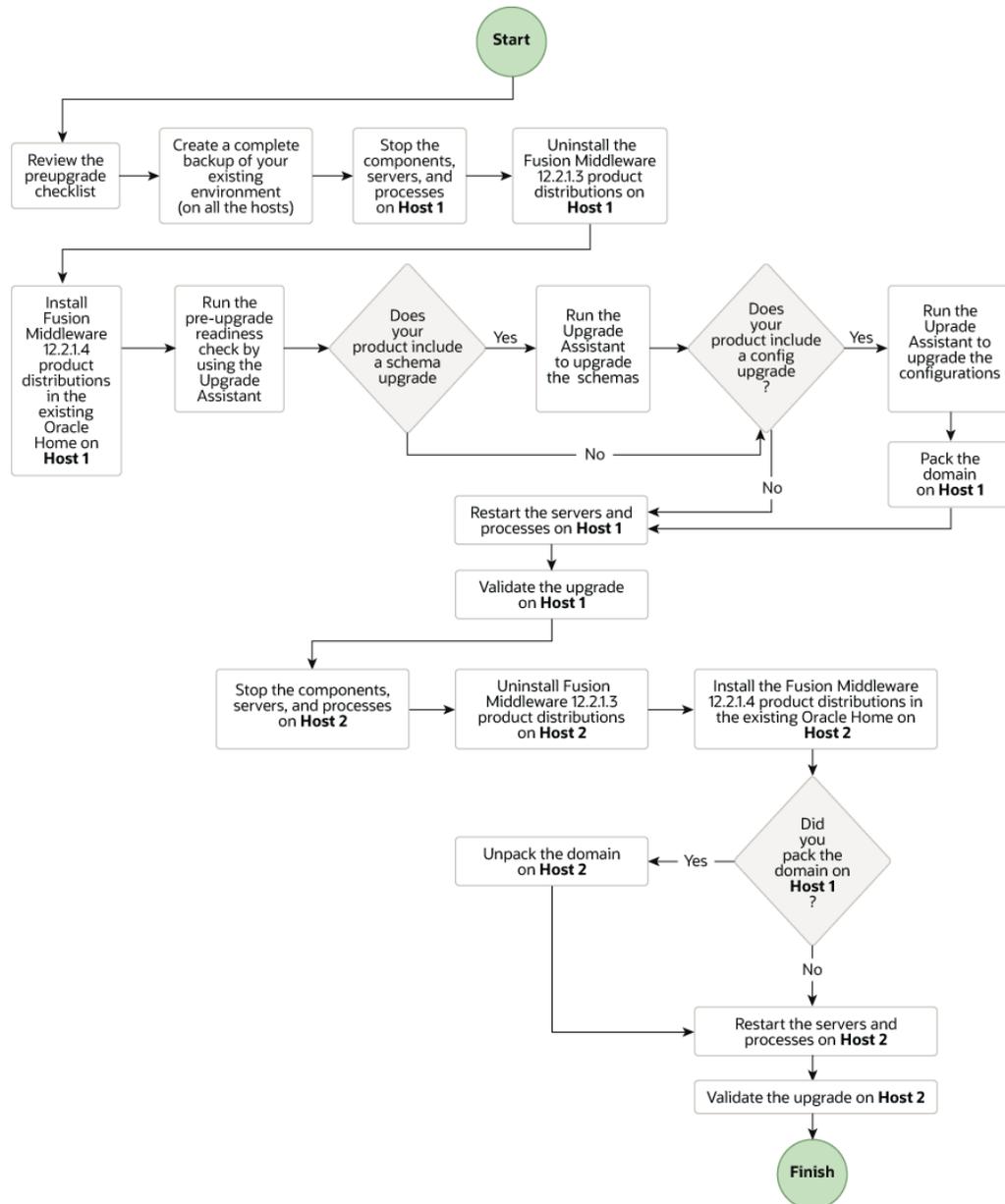


Table C-1 lists the high-level steps that you need to perform for a reduced downtime upgrade of the Oracle Fusion Middleware 12c (12.2.1.3.0) release. You must repeat these steps for each additional host.

Table C-1 Tasks for Performing a Reduced Downtime Upgrade of Oracle Fusion Middleware 12c (12.2.1.3.0) Release

Task	Description
<p>Required</p> <p>Before you begin a reduced downtime upgrade, you must complete the required pre-upgrade tasks.</p>	<p>The pre-upgrade tasks include reviewing the pre-upgrade checklists; backing up the Oracle home, domain directory, and component schemas; and using the appropriate JDK version.</p> <p>For a complete list of the pre-upgrade tasks, see Required Tasks that Must be Completed Before You Begin.</p>
<p>Required</p> <p>Create a complete backup of your existing environment, on all hosts.</p>	<p>See Creating a Complete Backup.</p>
<p>Required</p> <p>Stop the servers and processes on Host 1.</p>	<p>Before starting the upgrade process, stop all the servers, components, and processes on Host 1.</p> <p>See Stopping Components, Servers and Processes on Host 1.</p>
<p>Required</p> <p>Uninstall the Fusion Middleware 12c (12.2.1.3.0) product distributions on Host 1.</p>	<p>Uninstall the Fusion Middleware 12c (12.2.1.3.0) product distributions from the existing <code>ORACLE_HOME</code> so you can install the Fusion Middleware Infrastructure 12c (12.2.1.4.0) into the same directory.</p> <p>See Uninstalling the Software.</p>
<p>Required</p> <p>Install 12c (12.2.1.4.0) product distributions into the existing Oracle Home on Host 1.</p>	<p>Install Oracle Fusion Middleware Infrastructure 12c (12.2.1.4.0) by using the Oracle Universal Installer. You must install 12c (12.2.1.4.0) product distributions into the same <code>ORACLE_HOME</code>.</p> <p>See Installing the Product Distributions.</p>
<p>Optional</p> <p>Run the Readiness Check.</p>	<p>Running the Readiness Check by using the Upgrade Assistant helps you determine whether your pre-upgrade environment is ready for upgrade.</p> <p>See Running a Pre-Upgrade Readiness Check.</p>
<p>Required</p> <p>If applicable to your product, perform the schema and config upgrade separately on Host 1, by using the Upgrade Assistant.</p>	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Schema and configuration upgrade is not applicable for upgrading Oracle HTTPS Server.</p> </div>
<p>Required</p> <p>Restart the servers and processes on Host 1.</p>	<p>The upgrade process is complete. You can now restart the servers, components, and processes.</p> <p>See Restarting Node Manager, Administration Server, Managed Servers and Components on Host 1.</p>
<p>Required</p> <p>Validate the upgrade on Host 1.</p>	<p>After you complete the upgrade, perform the upgrade validation tasks.</p> <p>See Validating the Upgrade.</p>

Table C-1 (Cont.) Tasks for Performing a Reduced Downtime Upgrade of Oracle Fusion Middleware 12c (12.2.1.3.0) Release

Task	Description
<p>Required Stop the servers and processes on Host 2.</p>	<p>Before starting the upgrade, you must stop the system components, managed servers and node manager on Host 2. See Stopping the Components, Servers and Processes on Host 2.</p>
<p>Required Uninstall Fusion Middleware Infrastructure 12 c (12.2.1.3.0) on Host 2.</p>	<p>Uninstall Fusion Middleware Infrastructure 12c (12.2.1.3.0) from the existing ORACLE_HOME so you can install the Fusion Middleware Infrastructure 12c (12.2.1.4.0) into the same directory. See Uninstalling the Software.</p>
<p>Required Install Fusion Middleware Infrastructure 12c (12.2.1.4.0) and any other product distributions that you run in your domain on Host 2.</p>	<p>See Installing the Software into the Existing Middleware Home</p>
<p>Required Restart the servers and processes on Host 2.</p>	<p>After the upgrade is complete, restart the servers and processes. See Restarting the Managed Servers and Processes.</p>
<p>Required Validate the upgrade on Host 2.</p>	<p>After restarting the servers and processes, perform the upgrade validation tasks. See Validating the Upgrade.</p>

- [Performing a Reduced Downtime Upgrade](#)
If you are upgrading from Fusion Middleware 12c (12.2.1.3) release, you can use this process to upgrade your multi-node domain without shutting down all of the servers at the same time.

Performing a Reduced Downtime Upgrade

If you are upgrading from Fusion Middleware 12c (12.2.1.3) release, you can use this process to upgrade your multi-node domain without shutting down all of the servers at the same time.

The procedures described in this section are based on the Oracle Fusion Middleware Standard Installation Topology (SIT) and require that you have a multi-node environment. The standard installation topology for Oracle Fusion Middleware Infrastructure has a standard WebLogic Server domain that contains an Administration Server and a cluster containing two Managed Servers. **Host 1** is used to describe the procedures performed on the host with the Administration server and **Host 2** is used to describe the procedures performed on the other managed server host(s). If you have more than two hosts in your environment, be sure to complete the procedures on each additional node.

Required Tasks that Must be Completed Before You Begin

Review the following before you begin a reduced downtime upgrade:

- Review the preupgrade checklists for the components in your deployment. The checklists are found in each of the component-specific upgrade guides. Some products may require additional steps before performing the upgrade.
- Create a complete backup of the Oracle home (on all of the nodes), the entire domain directory (on all of the nodes) and component schemas before performing the upgrade. In addition, Oracle recommends that you create a backup of UI customizations and the applications directory, in addition to domain directory. See [Creating a Complete Backup](#).
- Make sure that you are using the appropriate JDK version for this release. For this release the correct version is `jdk1.8.0_211`
- If you are upgrading a shared component directory, back up the contents of the shared directory before the upgrade. The configuration upgrade makes changes to these directories.
- Make sure that your backups include any modified scripts, such as `setStartupEnv.sh`, for example. The upgrade will overwrite any customized files and you will lose your changes.
- [Performing the Upgrade on Host 1](#)
- [Performing the Upgrade on Host 2](#)
- [Validating the Upgrade](#)

Performing the Upgrade on Host 1

Perform the following tasks on the machine that hosts the Administration server and serves as the primary machine for your deployment.

- [Stopping Components, Servers and Processes on Host 1](#)
You must shut down all of the system components, processes, servers (including the Administration Server and any managed servers), and the node manager (if running).
- [Uninstalling the Software](#)
When performing a rolling upgrade, an empty directory is required for installing the new binaries prior to upgrading.
- [Installing the Product Distributions](#)
Before starting your upgrade, uninstall the software from the existing Oracle home, then use the Oracle Universal Installer to install the 12c (12.2.1.4.0) product distributions into the same Oracle home on the target system. You must install the product distributions on each host during the upgrade.
- [Restarting Node Manager, Administration Server, Managed Servers and Components on Host 1](#)
After the upgrade, you must restart the components, servers, and processes in the correct order.

Stopping Components, Servers and Processes on Host 1

You must shut down all of the system components, processes, servers (including the Administration Server and any managed servers), and the node manager (if running).

Note:

The procedures in this section describe how to stop components, servers, and processes using the WLST command line or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console.

An Oracle Fusion Middleware environment can consist of an Oracle WebLogic Server domain, an Administration Server, multiple managed servers, Java components, system components such as Identity Management components, and a database used as a repository for metadata.

The components may be dependent on each other, so they must be stopped in the following order:

- System Components (if any)
- Managed Server(s)
- Administration Server
- Node Manager

Stopping System Components

To stop system components, such as Oracle HTTP Server, use the `stopComponent` script:

```
DOMAIN_HOME\bin  
  
./stopComponent.sh component_name
```

Stopping the Managed Servers

To stop a WebLogic Server Managed Server, use the `stopManagedWebLogic` script:

1. Enter the following command:

```
(UNIX) DOMAIN_HOME/bin/stopManagedWebLogic.sh managed_server_name  
admin_url
```

```
(Windows)DOMAIN_HOME\bin\stopManagedWebLogic.cmd managed_server_name  
admin_url
```

2. When prompted, enter your user name and password.

Stop SOA servers and processes in this order:

Stopping the Administration Server

When you stop the Administration Server, you also stop the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control.

To stop the Administration Server:

1. Enter the following command:

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

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

2. When prompted, enter your user name, password, and the URL of the Administration Server.

Stopping the Node Manager

To stop Node Manager, close the command shell/prompt in which it is running.

Alternatively, after having set the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down.

Uninstalling the Software

When performing a rolling upgrade, an empty directory is required for installing the new binaries prior to upgrading.

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.

Note:

You must deinstall the upperstack components first, then deinstall JRF. After deinstalling the JRF, back up any remaining files and then delete all files in the directory. The installation directory must be empty.

Follow the instructions in this section to remove the software from the existing `ORACLE_HOME`. You will reinstall the new software into this same directory.

To start the Oracle Universal Installer in deinstallation mode, execute the following command:

UNIX: `ORACLE_HOME/oui/bin/deinstall.sh`

Windows: `ORACLE_HOME\oui\bin\deinstall.cmd`

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*.

Installing the Product Distributions

Before starting your upgrade, uninstall the software from the existing Oracle home, then use the Oracle Universal Installer to install the 12c (12.2.1.4.0) product distributions into the same Oracle home on the target system. You must install the product distributions on each host during the upgrade.

Note:

When Infrastructure is required for the upgrade, you must install the Oracle Fusion Middleware distribution first before you install other Fusion Middleware products.

Before you begin, note the following:

- Make sure that you have uninstalled the existing product distributions before installing the 12c (12.2.1.4.0) distributions.
- Use the same install path (Oracle home) for the installation of the new distributions that you used for the previous distributions that you uninstalled.

To install the 12c (12.2.1.4.0) distributions:

1. Sign in to the target system.
2. Download the following from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle Fusion Middleware Infrastructure (fmw_12.2.1.4.0_infrastructure_generic.jar)
 - Oracle HTTP Server (UNIX: fmw_12.2.1.4.0_ohs_linux64.bin), (Windows: setup_fmw_12.2.1.4.0_ohs_win64.exe)
3. Check that your machines meet the following requirements:
 - Ensure that the system, patch, kernel, and other requirements are met as specified in *Installing and Configuring Oracle HTTP Server*.
 - Because Oracle HTTP Server is installed by default on port 7777, you must ensure that port 7777 is not used by any service on the nodes. To check if this port is in use, run the following command before you install Oracle HTTP Server. You must free the port if it is in use.

```
netstat -an | grep 7777
```

4. On UNIX platforms, if the `/etc/oraInst.loc` file exists, check that its contents are correct. Specifically, check that the inventory directory is correct and that you have write permissions for that directory.

If the `/etc/oraInst.loc` file does not exist, you can skip this step.

5. Change to the directory where you downloaded the 12c (12.2.1.4.0) product distribution.
6. Start the installation program for Oracle Fusion Middleware Infrastructure:
 - (UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.4.0_infrastructure.jar`
 - (Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.4.0_infrastructure.jar`

7. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an 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, and click **Next**.

 **Note:**

The Installation Inventory Setup screen does not appear on Windows operating systems.

8. On the Welcome screen, review the information to make sure that you have met all the prerequisites. Click **Next**.
9. On the Auto Updates screen, select an option:
 - **Skip Auto Updates:** If you do not want your system to check for software updates at this time.
 - **Select patches from directory:** To navigate to a local directory if you downloaded patch files.
 - **Search My Oracle Support for Updates:** To automatically download software updates if you have a My Oracle Support account. You must enter Oracle Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

10. On the Installation Location screen, specify the location for the Oracle home directory and click **Next**.

Since you are installing the standard installation topology for a collocated Oracle HTTP Server in a WebLogic Server domain, enter the path to an existing Oracle Fusion Middleware Infrastructure Oracle home.

See Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.

11. On the Installation Type screen, select the following:
 - For Infrastructure, select **Fusion Middleware Infrastructure**
 - For Oracle HTTP Server, select **Collocated HTTP Server (Managed through WebLogic Server)**

Click **Next**.

12. The Prerequisite Checks screen analyzes the host computer to ensure that the specific operating system prerequisites have been met.

To view the list of tasks that are 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).

13. On the Installation Summary screen, verify the installation options that 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.

Click **Install** to begin the installation.

14. On the Installation Progress screen, when the progress bar displays 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.
15. The Installation Complete screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.
16. After you have installed Oracle Fusion Middleware Infrastructure, enter the following command to start the installer for your product distribution and repeat the steps above to navigate through the installer screens:

(UNIX) `./fmw_12.2.1.4.0_ohs_linux64.bin`

(Windows) `setup_fmw_12.2.1.4.0_ohs_win64.exe`

You have installed the Oracle HTTP Server in a collocated mode.

Restarting Node Manager, Administration Server, Managed Servers and Components on Host 1

After the upgrade, you must restart the components, servers, and processes in the correct order.



Note:

The procedures in this section describe how to start servers and processes using the WLST command line or a script. You can also use the Oracle Fusion Middleware Control and the Oracle WebLogic Server Administration Console. See Starting and Starting Administration and Managed Servers and Node Manager in *Administering Oracle Fusion Middleware*.

The components must be started in the following order:

- Node Manager
- Administration Server
- Managed Server(s)
- System Components



Note:

If you are unable to successfully start any of the following components on Host 1, do not proceed with the upgrade on the remaining Hosts. You must first resolve the issues with the components on Host 1.

 **Note:**

Windows Users Only: When restarting the servers on a Windows operating system, the upgraded domain might fail with a parsing exception. To fix this parsing error, add the property "`-Doracle.xml.schema.Ignore_Duplicate_components=true`" to the server startup script `setDomainEnv.cmd`.

Starting the Node Manager

To start Node Manager:

- Enter the following command:

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

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

Starting the Administration Server

To start the Administration Server:

1. Enter the following command:

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

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

2. When prompted, enter your user name, password, and the URL of the Administration Server.

Starting the Managed Servers

To start the Managed Servers:

1. Enter the following command:

(UNIX) `DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name admin_url`

(Windows) `DOMAIN_HOME\bin\startManagedWebLogic.cmd managed_server_name admin_url`

2. When prompted, enter your user name and password.

Starting Component Processes

When you stopped the Administration Server, you also stopped the processes running in the Administration Server, including the WebLogic Server Administration Console and Fusion Middleware Control. After the upgrade, you will need to restart all of these processes.

1. Navigate to the `DOMAIN_HOME\bin` directory.
2. Execute the following script for each component

`/startComponent.sh component_name`

Performing the Upgrade on Host 2

Once you have completed the upgrade on host 1, perform the following steps on each additional host in your environment. Our standard topology example includes only two hosts, but you may have more.

- [Stopping the Components, Servers and Processes on Host 2](#)
You must stop the system components, managed servers and node manager running on Host 2.
- [Uninstalling the Software](#)
When performing a rolling upgrade, an empty directory is required for installing the new binaries prior to upgrading.
- [Installing the Software into the Existing Middleware Home](#)
Install the software into the empty Middleware home.
- [Restarting the Managed Servers and Processes](#)
After the upgrade is complete on Host 2, restart the managed servers.

Stopping the Components, Servers and Processes on Host 2

You must stop the system components, managed servers and node manager running on Host 2.

Stop the system components first, followed by the managed server(s) and then the node manager. Failure to stop the components in the correct order may result in a failed upgrade.

Stopping System Components

To stop system components, such as Oracle HTTP Server, use the `stopComponent` script:

```
DOMAIN_HOME\bin  
  
./stopComponent.sh component_name
```

Stopping the Managed Servers

To stop a WebLogic Server Managed Server, use the `stopManagedWebLogic` script:

1. Enter the following command:

```
(UNIX) DOMAIN_HOME/bin/stopManagedWebLogic.sh managed_server_name  
admin_url  
  
(Windows)DOMAIN_HOME\bin\stopManagedWebLogic.cmd managed_server_name  
admin_url
```

2. When prompted, enter your user name and password.

Stop SOA servers and processes in this order:

Stopping the Node Manager

To stop Node Manager, close the command shell/prompt in which it is running.

Alternatively, after having set the `nodemanager.properties` attribute `QuitEnabled` to `true` (the default is `false`), you can use WLST to connect to Node Manager and shut it down.

Uninstalling the Software

When performing a rolling upgrade, an empty directory is required for installing the new binaries prior to upgrading.

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.

Note:

You must deinstall the upperstack components first, then deinstall JRF. After deinstalling the JRF, back up any remaining files and then delete all files in the directory. The installation directory must be empty.

Follow the instructions in this section to remove the software from the existing `ORACLE_HOME`. You will reinstall the new software into this same directory.

To start the Oracle Universal Installer in deinstallation mode, execute the following command:

UNIX: `ORACLE_HOME/oui/bin/deinstall.sh`

Windows: `ORACLE_HOME\oui\bin\deinstall.cmd`

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*.

Installing the Software into the Existing Middleware Home

Install the software into the empty Middleware home.

Uninstall the software from the existing Oracle home before installing the 12c (12.2.1.4.0) product distributions. Install the software using the same process you used to install on Host 1.

Restarting the Managed Servers and Processes

After the upgrade is complete on Host 2, restart the managed servers.

To start the Managed Servers:

1. Enter the following command:

```
(UNIX) DOMAIN_HOME/bin/startManagedWebLogic.sh managed_server_name  
admin_url
```

```
(Windows) DOMAIN_HOME\bin\startManagedWebLogic.cmd  
managed_server_name admin_url
```

2. When prompted, enter your user name and password.

Validating the Upgrade

After you have completed the upgrade on all hosts, complete the standard upgrade verification tasks to ensure that all components will continue to work as expected.

See [Tasks to Perform After Upgrade](#).



Note:

Only perform those tasks that pertain to your environment, configuration and preferences. These tasks are meant to assist you in verifying that the upgrade was successful. You may need to perform additional testing based on your configuration.

Recovering from a Failed Upgrade

If your upgrade is unsuccessful, you will need to restore your environment from backup. Make sure that you include your backed up configuration and script files. Restore the backup of the Oracle home (on all of the nodes), the entire domain directory (on all of the nodes) and component schemas. In addition, you will need to restore any UI customizations and the applications directory, in addition to domain directory.