Oracle® Fusion Middleware

Upgrading Oracle WebLogic Server

14c (14.1.2.0.0)

F54637-02

January 2025

Oracle Fusion Middleware Upgrading Oracle WebLogic Server, 14c (14.1.2.0.0)

F54637-02

Contents

Preface

1 Introduction

2 Roadmap for Upgrading Your Application Environment

3 Reconfiguring WebLogic Domains

Preface

This document describes how to upgrade an application environment from an earlier version of Oracle WebLogic Server to Oracle WebLogic Server 14c (14.1.2.0.0).

Documentation Accessibility

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

Access to Oracle Support

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

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.

Conventions

The following text conventions are used in this document:

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

1
Introduction

You can upgrade WebLogic servers and domains from a supported version of WebLogic Server to WebLogic Server 14c (14.1.2.0.0). You can also update an existing application to run on Oracle WebLogic Server 14.1.2.0.0.

While upgrading to version 14.1.2.0.0, you might want to change your application or you might have to change the application. However, this document focuses only on issues that you should consider when moving an application to WebLogic Server 14.1.2.0.0 without making any application changes.

The instructions in this document are for the following supported upgrade scenarios:

- Upgrading standalone WebLogic Server 14.1.1 to WebLogic Server 14.1.2.0.0
- Upgrading from WebLogic Server 12.2.1.4 to WebLogic Server 14.1.2.0.0

> **Note:**
>
> If your preupgrade enviroment version is pre-12.2.1.4.0, then you will need to first upgrade to a supported starting point before you can begin an upgrade to 14.1.2.0.0. You cannot upgrade directly to 14.1.2.0.0 from an unsupported version. Use the upgrade procedures in the 12c (12.2.1.4.0) upgrade documentation library for information on upgrading to a supported starting point.

WebLogic Server generally supports high levels of upgrade capability across WebLogic Server versions. This document is intended to provide WebLogic Server upgrade support and identify issues that may surface during an upgrade so that they can be easily resolved.

> **Note:**
>
> If you are upgrading from version 12.2.1.4 to version 14.1.2.0.0, the Reconfiguration Wizard only needs to be run when the location of the JDK or the Oracle Home is changed as part of the upgrade. If the Oracle Home and the JDK binaries are in the same location, running Reconfiguration Wizard is not required.

Version Compatibility

Before you upgrade WebLogic Server, review the WebLogic Server and domain compatibility requirements for WebLogic Server 14.1.2.0.0.

See *Compatibility Within a Domain* in *Understanding Oracle WebLogic Server*.

Important Terminology

The documentation for upgrading WebLogic Server uses various terms when describing its features and functionality. It is important that you have a good understanding of these terms.

- **Upgrade**—In this document, the term upgrade refers to the process of upgrading WebLogic Server and moving an existing application, unchanged, to a new (upgraded) WebLogic Server version.
- **Reconfiguration**—The process of upgrading a domain that was created with a previous WebLogic Server version so that it is compatible with the WebLogic Server version to which you have upgraded. This can be done using either the Reconfiguration Wizard or WLST.
- **Application Environment**—An application environment includes applications and the WebLogic domains in which they are deployed. It also includes any application data associated with the domain, and may include resources such as database servers, firewalls, load balancers, and LDAP servers.
- **Migrate**—To move an application or domain configuration from a third-party product to an Oracle product.
- **Interoperability**—(1) The ability of an application deployed in one WebLogic Server version to communicate with another application that is deployed in a different WebLogic Server version. (2) The ability of Oracle product components to communicate with third-party software using standard protocols.
- **Compatibility**—The capability of an application built using one WebLogic Server release to run in another WebLogic Server release, regardless of whether the application was rebuilt.

Overview of the Upgrade Process

You can upgrade all WebLogic Server applications and domains simultaneously, upgrade them in a well-defined sequence, or upgrade some applications and domains while leaving other applications and domains on older WebLogic Server versions.

The process required to upgrade an application environment depends on the scope of the application. An *application environment* includes a WebLogic domain and any applications and application data associated with the domain. It may also include external resources, such as firewalls, load balancers, and LDAP servers. Figure 1-1 shows an example of a WebLogic application environment.

Figure 1-1 Example WebLogic Application Environment

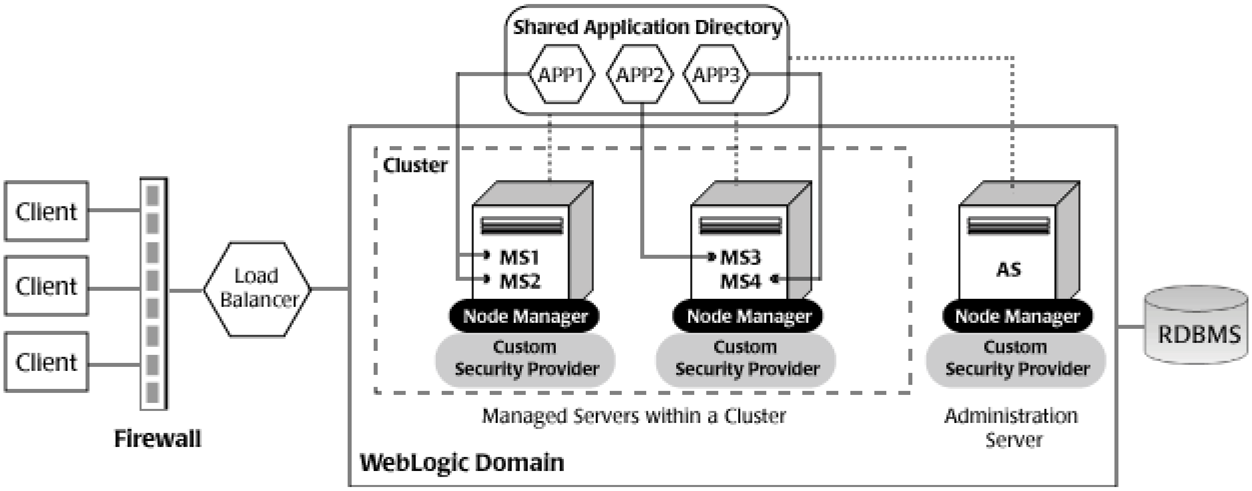

Table 1-1 lists the components of the WebLogic application environment shown in Figure 1-1 and the upgrade requirements for each.

Table 1-1 Upgrade Requirements for Components in Example WebLogic Application Environment

Component	Description	Upgrade Requirements
WebLogic domain	Includes the Administration Server (AS) and optionally one or more Managed Servers (for example, MS1, MS2, MS3, and MS4). The servers in a domain may span multiple machines. Furthermore, you can group Managed Servers into clusters to support load balancing and failover protection for critical applications. For more information about WebLogic domains, see Understanding Oracle WebLogic domains in *Understanding Domain Configuration for Oracle WebLogic Server*.	Upgrade the domain directory on each computer in the domain.
Applications	Any Java EE applications, including Web applications, EJBs, and so on. Typically, applications are deployed to one or more Managed Servers in a domain. Depending on the deployment strategy, applications may reside locally on a computer or be accessible using a shared directory. In addition, external client applications may access the application environment from outside a firewall.	Most WebLogic Server applications can be run without modifications in the new WebLogic Server 14.1.2.0.0 application environment.
External resources	Software components, such as databases for storing domain and application data, load balancers, and firewalls.	Verify that all external resources are compatible with WebLogic Server 14.1.2.0.0. See Oracle Fusion Middleware Supported System Configurations.

Upgrading business applications that are deployed to WebLogic Server may involve upgrading multiple WebLogic Server applications, and in some cases domains, in a coordinated fashion to:

- maintain consistency in the WebLogic Server versions being used
- use the same supported configurations environment across the entire installation
- meet specific interoperability requirements.

Before You Begin

Before you upgrade WebLogic Server, verify that your machine is set up to meet the requirements to upgrade and run WebLogic Server. You must also consider the scope of the environments that you are upgrading and which applications are upgraded in which sequence.

As covering all the permutations of an upgrade is beyond the scope of this document, consider the following items prior to planning your upgrade. These items focus on upgrades that involve a single application running in a single domain. Perform any applicable tasks before you begin the upgrade. Failure to complete the required tasks may result in a failed upgrade or extended system downtime.

- When upgrading a domain to 14c (14.1.2.0.0) from a prior release, if there is no explicit secure mode setting, then the Reconfiguration Wizard will explicitly set secure mode to “disabled” in the upgraded domain. This is to preserve the behavior that was present in the original domain. If there is an explicit secure mode setting, it will be preserved in the upgraded domain.
- Oracle recommends that you upgrade an application in development environments and use a standard QA, testing, and staging process to move upgraded applications to a production environment.
- You typically upgrade an application either by upgrading an existing domain or by creating a new domain, from which you can run the application on the new WebLogic Server version. Sometimes, you prefer to create new domains using the **Fusion Middleware Configuration Wizard** or other configuration tools (such as WLST) to test the applications that you are upgrading.
- When planning a WebLogic Server version upgrade, you should review the Oracle Fusion Middleware Supported Systems Configurations page to ensure that your upgraded environment is supported by Oracle, in particular:
  - current and planned JVM and JDK versions
  - operating system versions
  - database versions
  - Web services versions
  - versions of other products that interoperate with or run on WebLogic Server, to ensure that the upgraded environment is supported by Oracle or other vendors' products that you are using with WebLogic Server.
- On an ongoing basis, Oracle documents APIs and features that have been deprecated (that is, planned for removal in a future release). This is intended to inform you that you should avoid using these APIs and features to ensure upgradability. Oracle also documents the APIs and features that have actually been removed in the current release so that if you are upgrading from prior versions, you can determine if your applications will be affected by an upgrade. See Identifying Unused APIs

  When upgrading, you should review all documentation of deprecated and removed features for all applicable WebLogic Server versions.
- You should consider the impact (if any) that the upgrade process may have on any automation (such as WLST scripts) that you are using to configure, deploy, start/stop, or monitor your WebLogic Server applications. You may need to upgrade such automation along with the applications and domains you are upgrading.
- You should consider the potential impact that may result from the use of third-party libraries in your applications, as they may conflict with different versions of those same libraries that

are embedded in WebLogic Server. In particular, new versions of WebLogic Server may change the version of open source libraries that are embedded in WebLogic Server. Applications that may run successfully on earlier WebLogic Server versions may encounter new class conflicts after upgrade.

If you are upgrading an application that contains embedded third-party libraries, you should consider using the Classloader Analysis Tool, and filtering classloaders when upgrading WebLogic Server applications to WebLogic Server 14.1.2.0.0. This tool enables you to identify, diagnose and resolve such conflicts, and may simplify the upgrade process.

- If you are running applications on prior versions of WebLogic Server, and are using WebLogic Server patches or bug fixes, you should investigate whether or not those patches or bug fixes have been incorporated into the version of WebLogic Server to which you are upgrading.

2

Roadmap for Upgrading Your Application Environment

Use the upgrade roadmap to identify the procedure required to upgrade your Oracle WebLogic Server application environment. An upgrade of WebLogic application environment is complete when you upgrade, configure, and deploy your WebLogic application environments.

- Plan for an Upgrade
- Prepare for the Upgrade
- Upgrade an Application Environment
- Reconfiguring WebLogic Domains

Plan for an Upgrade

Before upgrading your WebLogic application environment, plan the upgrade path. Planning the upgrade path includes generating an inventory of the application environment, verifying the supported system configurations, reviewing the compatibility information of application environment, and creating an upgrade plan.

To ensure that your plan addresses all the aspects of upgrading that are necessary for your environment, complete the following steps:

- Step 1: Inventory the Application Environment
- Step 2: Verify Supported Configuration Information
- Step 3: Review the Compatibility Information
- Step 4: Create an Upgrade Plan

Step 1: Inventory the Application Environment

Generate an inventory of the application environment by identifying the following components:

- Administration Server and the computer on which it resides
- Managed Servers and the computer(s) on which they reside
- Location of the applications (including all external client applications)
- External resources, for example:
  - Databases used to store persisted and application data
  - Firewalls
  - Load balancers
- Tools, scripts, templates, and source code used for automating the tasks required to create the application environment

You can view a sample application environment in Overview of the Upgrade Process.

Step 2: Verify Supported Configuration Information

Supported configurations (for example, JDK versions, Operating System versions, Web server versions, and database versions) have changed for WebLogic Server 14.1.2.0.0. You may be required to upgrade your environments to the supported versions of these and other products.

For information about supported configurations, see *Oracle Fusion Middleware Supported System Configurations* on Oracle Technology Network (OTN).

For databases, note that:

- The evaluation database available from the installation program that is provided for use by the sample applications and code examples, and as a demonstration database, is Derby. Derby is an open source relational database management system based on Java, JDBC, and SQL standards. For more information about Derby, see `http://db.apache.org/derby/`.
- The Oracle Thin Drivers are included as part of the WebLogic Server installation.
- If you are using an Oracle OCI database driver and want to change to use a Thin database driver, you must remove the `server` property (as illustrated below) from the generated JDBC module. For example:

  ```
  <property>
    <name>server</name>
    <value>servername</value>
  </property>
  ```

- The Oracle Thin Drivers are installed with WebLogic Server and are ready for use. For more information about using these drivers, see JDBC Drivers Installed with WebLogic Server in *Administering JDBC Data Sources for Oracle WebLogic Server*.

Step 3: Review the Compatibility Information

Most existing WebLogic Server applications can be run without modification in the new WebLogic Server 14.1.2.0.0 application environment.

Step 4: Create an Upgrade Plan

Using the information gathered in the preceding steps, create a plan for upgrading your application environment.

Identify the scope and timing of the upgrade process, based on your business needs. Note the following points:

- Oracle does not recommend upgrading an application environment that is currently deployed in production. Instead, you should upgrade your application environment while it is under development or test and execute standard procedures for quality assurance and performance tuning before promoting the upgraded environment to production.
- If you start a WebLogic Server after performing a significant upgrade, then the WebLogic Server, the products layered on top of the WebLogic Server, and/or your applications can make irreversible changes to existing data files and database tables. For example, the default file stores may change such that the older versions will no longer be able to load them. Significant upgrade includes upgrade from the major versions, minor versions, and patch sets, or any specific patch/application that changes the files or database formats.

- If your application is complex, for example, if it includes multiple clustered domains and a large number of deployed applications, you may choose to upgrade the components of the application environment in stages.
- You may consider limiting the number of WebLogic Server versions used in any single application environment to minimize the diversity and cost of systems being administered.
- If you plan to use the RDBMS security store in a WebLogic domain, Oracle recommends that you create a new domain in which the RDBMS security store is configured. If you have an existing domain in which you want to use the RDBMS security store, you should create the new domain, and then migrate your security realm to it. Oracle does not recommend "retrofitting" the RDBMS security store to an existing domain. See Managing the RDBMS Security Store in *Administering Security for Oracle WebLogic Server*.

Prepare for the Upgrade

Before you start the upgrade process, you should verify whether there are any upgrade compatibility issues that apply to your applications. You then shut down all running server instances and back up the application components in your domain.

Review the following steps to prepare your system for an upgrade.

Step 1: Check Your Applications (Undeploy If Necessary)

It is not necessary for WebLogic Server applications to be undeployed before upgrading the domain. In most cases, WebLogic Server applications can be run without modifications in the new WebLogic Server 14.1.2.0.0 application environment. Note that if you use deprecated or removed APIs in the application, you might encounter warnings or exceptions at run time.

WebLogic provides tools that can help simplify the application migration process by identifying unused APIs or activating recipes that perform migration tasks for you automatically.

- The **WebLogic Migration Analysis Tool** is a command line utility that identifies APIs in a WebLogic application that have been removed or are no longer used in WebLogic Server. For more information about using the WebLogic Migration Analysis Tool, see Identifying Unused APIs .
- The **WebLogic OpenRewrite Recipes** project uses a standard OpenRewrite module that is designed to be used by WebLogic Server application users who need to migrate to a new WebLogic version or an updated JDK. For more information, see Using OpenRewrite Recipes to Migrate WebLogic Applications.

Using OpenRewrite Recipes to Migrate WebLogic Applications

Use the OpenRewrite recipes to simplify migrating your WebLogic applications to a new version of WebLogic and Java.

The Oracle WebLogic OpenRewrite recipes implement a Rewrite module that performs common tasks when migrating your WebLogic applications to a new version of WebLogic and Java.

For more information about activating these recipes for your own WebLogic applications, see the Getting Started Readme located in the public WebLogic Rewrite GitHub repository at WebLogic OpenRewrite Project .

Identifying Unused APIs

It is important to know which classes and APIs are no longer in the new version and to address any changes required for the application to successfully deploy. Use the WebLogic Migration Analysis Tool to identify these classes and APIs.

The WebLogic Migration Analysis Tool is a command line utility that identifies APIs in a WebLogic application that have been removed or are no longer used in WebLogic Server 14.1.2.0.0 and generates a report. The report is meant to show only those deletions that are on the classpath of a running WebLogic Server (such as weblogic.jar). The reports are not intended to report on all jar files that are missing in a class - just those associated with the WebLogic applications.

> **Note:**
>
> This report will not catch all potential problems associated with missing or deprecated APIs or classes. For example, if you are using reflection, this report will not detect it. Similarly, this report may state that a library has been removed and could cause and issue, but you may have your own copy of that library somewhere else.

To run the Migration Analysis Tool, execute the following:

```
java -jar $WL_HOME/server/lib/weblogic.migration-analysis-tool.jar <archive-
file-name> <archive-2> <archive-3>
```

Where <*archive-file-name*> parameters are the files you want to analyze. For example:

```
java -jar $WL_HOME/server/lib/weblogic.migration-analysis-tool.jar /tmp/
em_example.war
```

Step 2: Shut Down Servers in the Application Environment

Before you upgrade, you must shut down all servers in the application environment.

Step 3: Back Up the Application Environment

Oracle recommends that before upgrading your application environment, you manually back up the components defined in Table 2-1. You should back up the relevant information on all machines in the domain.

Table 2-1 Recommendations for Backing Up the Application Environment

Component	Recommendations
Domain directory	Back up the Administration Server and any remote Managed Server domain directories that are defined in the application environment. **Note:** The Domain Upgrade Wizard, which automatically backed up the domain being upgraded, is no longer provided with WebLogic Server. You must manually back up your domain directory prior to upgrading the domain.

Table 2-1 (Cont.) Recommendations for Backing Up the Application Environment

Component	Recommendations
Applications and application-persisted data	Back up any applications and data that reside outside of the domain directory.
Log files	If it is important for you to maintain a record of all messages that are logged, back up the log files. As log files can be large, you may want to delete them to conserve disk space if it is not important to retain them.

Step 4: Install Required Oracle Products

Before upgrading your application environment, you must install the Oracle WebLogic Server 14.1.2.0.0 products that you require on each computer in the domain. For more information about installing Oracle WebLogic products, see *Installing and Configuring Oracle WebLogic Server and Coherence*.

Step 5: Set Up the Environment

To set up the environment for an upgrade:

1. Open an MS-DOS command prompt window (on Windows) or a command shell (on UNIX).
2. Add the WebLogic Server classes to the CLASSPATH environment variable and `WL_HOME\server\bin` to the PATH environment variable, where `WL_HOME` refers to the top-level installation directory for WebLogic Server.

   You can perform this step by running the `WL_HOME\server\bin\setWLSEnv` script.

   > **Note:**
   >
   > On UNIX operating systems, the `setWLSEnv.sh` command does not set the environment variables in all command shells. Oracle recommends that you execute this command using the Korn shell or bash shell.

Upgrade an Application Environment

To upgrade your application environment to the latest version of WebLogic Server, back up the domain, upgrade the Administration Server host machine, configure the Node Manager, and upgrade each Managed Server instance.

The table below summarizes the steps for updating an application environment. Each step that is performed must be done on every machine in the domain and in the given sequence shown in this table.

Table 2-2 Procedure for Upgrading an Application Environment

Task	Description
Back up the domain	Before upgrading the domain on the Administration Server, ensure to backup the domain. See Backing Up the Domain.

Table 2-2 (Cont.) Procedure for Upgrading an Application Environment

Task	Description
Upgrade WebLogic domain (Administration Server)	Run the Reconfiguration Wizard to upgrade the WebLogic domain on the computer that hosts the Administration Server. **Notes:** Oracle recommends that you completely upgrade the domain on the Administration Server before upgrading the domain on the Managed Servers. Depending on the Node Manager configuration of the original domain and the desired Node Manager configuration of the upgraded domain, you may be able to upgrade Node Manager by using the Reconfiguration Wizard. See Determining Node Manager Upgrade Procedure.
Complete Node Manager configuration	This is needed only if your existing domain is using a per host Node Manager configuration and you want to continue using a per domain Node Manager in the upgraded domain. See Completing the Node Manager Configuration.
Back up the domain on each Managed Server.	Prior to upgrading the domain on a Managed Server, make a backup copy of the domain.
Upgrade WebLogic domain (remote Managed Servers)	Use the `pack` and `unpack` commands or the WLST `writeTemplate()` command in online mode to upgrade the WebLogic domain on every computer that hosts any Managed Servers. **See Also**: • Updating a Managed Server Domain on a Remote Machine • *Creating Templates and Domains Using the Pack and Unpack Commands* **Note:** • The `unpack` command works only with the same version used to `pack` the WebLogic domain. • Managed Servers that reside on the same computer as the Administration Server do not require additional upgrade steps.

Troubleshooting an Upgrade

If the upgrade process fails at any step, the Reconfiguration Wizard displays a message indicating the reason for the failure, and then terminates the upgrade process.

To proceed with the upgrade process, perform the following steps:

1. Restore the application environment to its original state using the backup files you created in Step 3: Back Up the Application Environment.
2. Correct the failure reported by the Reconfiguration Wizard.
3. Run the Reconfiguration Wizard again to upgrade the domain.

If you encounter any issues during the upgrade or post upgrade, refer to the Known Issues and Workarounds in *Release Notes for Oracle WebLogic Server* to troubleshoot the issue.

3 Reconfiguring WebLogic Domains

You can use the Reconfiguration Wizard to upgrade any WebLogic domain that was created with Oracle WebLogic Server 12.2.1.4 or later.

When you use the Reconfiguration Wizard to reconfigure a WebLogic Server domain, the following items are automatically updated, depending on the applications in the domain:

- WLS core infrastructure
- Domain version

> **Note:**
>
> The Reconfiguration Wizard does not update any of your applications that are included in the domain. For information about how to upgrade your applications, see WebLogic Server 14.1.2.0.0 Compatibility with Previous Releases.

Learn how to use the Reconfiguration Wizard to reconfigure WebLogic Server domains.

- Before You Begin
- Updating a Managed Server Domain on a Remote Machine
- Important Notes About the Domain Upgrade Process

Before You Begin

The reconfiguration may require you to perform additional tasks, such as configuring the `CONFIG_JVM_ARGS` environment variable, backing up the domain, and choosing the Node Manager configuration that you want to use with the upgraded domain.

Setting CONFIG_JVM_ARGS on UNIX and Linux Systems

Prior to running the Reconfiguration Wizard to reconfigure a domain on a UNIX or Linux operating system, if you have not already done so, set the `CONFIG_JVM_ARGS` environment variable to the following value to use the operating system's random number generator:

```
-Djava.security.egd=file:/dev/./urandom
```

This decreases the amount of time it takes for the Reconfiguration Wizard to reconfigure a domain.

Backing Up the Domain

Prior to running the Reconfiguration Wizard, make a backup copy of the domain directory. For example, copy C:\domains\mydomain to C:\domains\mydomain_backup.

Prior to updating the domain on each remote Managed Server, make a backup copy of the domain directory on each remote machine.

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 prior to reconfiguration.

Determining Node Manager Upgrade Procedure

A Node Manager default configuration is a per domain Node Manager configuration. That is, the Node Manager configuration is specific to a given domain. This configuration allows multiple domains on a given machine to have different Node Manager configurations. See Default Node Manager Configuration in *Administering Node Manager for Oracle WebLogic Server*.

Table 3-1 shows the supported Node Manager upgrade paths when upgrading WebLogic Server from version 12.2.1.4.0 or later to the current version. Per host in this context means any Node Manager configuration that is outside of your per domain Node Manager configurations.

Table 3-1 Supported Node Manager Upgrade Paths

Node Manager Upgrade Paths	From WebLogic Server 12.2.1.4 or later
Per domain to per domain	Supported
Per domain to per host	Not supported
Per host to per domain	Supported
Per host to per host	Manual configuration

Table 3-2 shows the Node Manager upgrade details for each supported upgrade path.

Table 3-2 Node Manager Upgrade Details

Per Domain to Per Domain	Per Host to Per Domain	Per Host to Per Host
This is an automatic upgrade for all WebLogic Server 12.2.1.4.0 or later releases that are already configured for per domain Node Manager. The environment is updated to standard settings and can be customized later. The upgrade is automatic whether you are using the Reconfiguration Wizard or WLST to upgrade the domain.	In this case, the Reconfiguration Wizard provides a Node Manager screen during domain reconfiguration. Use this screen to select the Node Manager configuration to use for the reconfigured domain. The resulting configuration depends on the combination of options that you select for **Node Manager Type** and **Node Manager Configuration**. You can also use WLST to upgrade the domain and Node Manager configuration as desired. See Reconfiguring a WebLogic Domain Using WebLogic Scripting Tool. If multiple per domain Node Managers run on the same machine, see Configuring Multiple Per Domain Node Managers on the Same Machine. Click the **Help** button on the screen on the Fusion Middleware Reconfiguration Wizard window to see the *Reconfiguration Wizard Context-Sensitve Help*.	Node Manager configuration must be completed manually as described in Completing the Node Manager Configuration.

Configuring Multiple Per Domain Node Managers on the Same Machine

If you have multiple domains on the same machine using a Per Domain Node Manager configuration, when running the Reconfiguration Wizard, do the following:

- On the Advanced Configuration screen, select **Managed Servers, Clusters, and Coherence** to reconfigure the existing machines for the 14c Node Manager.
- No changes are needed on the Managed Servers and Clusters screens. When the Machines screen appears, ensure that you use a unique Node Manager port for each domain. For example, if you have three per domain Node Managers running on the machine, use port 5556 for Domain 1, port 5557 for Domain 2, and port 5558 for Domain 3.

  Click the **Help** button on the screen on the Fusion Middleware Reconfiguration Wizard window to see the *Reconfiguration Wizard Context-Sensitve Help* .

Running Two Per Host Node Managers on the Same Machine

If all the following items apply to your upgrade scenario, extra steps are needed during the reconfiguration process to create a second Node Manager for the 14c domains:

- You want to upgrade only some of your existing domains to 14c.
- You want to continue using a per host Node Manager for the 14c domains.
- Your existing per host Node Managers and 14c per host Node Managers are running on the same machine.

When running the Reconfiguration Wizard:

- On the Node Manager screen, select **Manual Node Manager Setup**. This option keeps the Node Manager configuration as a per host Node Manager for the 14c domain being upgraded.
- On the Advanced Configuration screen, select **Managed Servers, Clusters, and Coherence** to reconfigure the existing machines for the 14c Node Manager. In addition, select **Deployments and Services** to check machine assignments for your deployments and services.
- No changes are needed on the Managed Servers and Clusters screens. When the Machines screen appears, change the name of each machine to something other than the name that is being used for the 12c domains. In addition, enter a Node Manager port number that is different than the Node Manager port number that is being used for the exisitng Node Manager. Use the same port number for each 14c machine in this domain.
- Verify that your deployments and services are assigned to the new machine names.

Reconfiguring a WebLogic Domain

Oracle provides a choice of two tools for reconfiguring a WebLogic domain: the graphical Fusion Middleware Reconfiguration Wizard or the WebLogic Scripting Tool (WLST).

Caution:

Once the domain reconfiguration process starts, it is irreversible. Before using the Reconfiguration Wizard or WLST to upgrade the domain, ensure that you have backed up the domain as described in Backing Up the Domain. If an error or other interruption occurs during the reconfiguration process, you must restore the domain by copying the files and directories from the backup location to the original domain directory. This workaround is the only way to ensure that the domain has been returned to its original state before reconfiguration.

When you reconfigure a domain:

- The domain version number in the config.xml file for the domain is updated to the Administration Server's installed WebLogic Server version major and minor version number (for example, 14.1.2.0).
- As of WebLogic Server 14.1.2.0.0, when you select production mode, it automatically enables the stricter settings of secured production mode. If you truly want to implement the more moderate settings of production mode, you must explicitly disable secured production mode after the upgrade. For more information, see Changing Domain Mode Post Upgrade.
- Reconfiguration templates for all installed Oracle products are automatically selected and applied to the domain. These templates define any reconfiguration tasks that are required to make the WebLogic domain compatible with the current WebLogic Server version.
- Start scripts are updated.
- After reconfiguring the domain on the Administration Server, you must port the reconfigured domain to all remote Managed Servers in the domain. See Updating a Managed Server Domain on a Remote Machine.
- After reconfiguring a domain to a per host Node Manager by using either WLST or the Reconfiguration Wizard, you must take additional steps to complete the Node Manager configuration. See Completing the Node Manager Configuration and Completing the Node Manager Configuration (Two Per Host Node Managers).

Reconfiguring a WebLogic Domain in Graphical Mode

To reconfigure a domain using the Reconfiguration Wizard, you first launch it from a DOS command prompt or UNIX shell, and then provide the required upgrade details in a sequence of screens that are displayed.

Note:

If you cannot run the Reconfiguration Wizard in GUI mode, Oracle recommends that you use a WLST script to reconfigure your domain. See Reconfiguring a WebLogic Domain Using WebLogic Scripting Tool.

To start the Reconfiguration Wizard in graphical mode from a Windows command prompt or on UNIX systems:

1. Log in to the system on which the domain resides.

2. Open an MS-DOS command prompt window (on Windows) or a command shell (on UNIX).
3. Go to the following directory, where `ORACLE_HOME` is your Oracle home directory:

   On Windows: `ORACLE_HOME\oracle_common\common\bin`

   On UNIX: `ORACLE_HOME/oracle_common/common/bin`
4. Run the following commands:

   On Windows: `reconfig.cmd`

   On UNIX: `sh reconfig.sh`

   > **Note:**
   >
   > When you run the `reconfig.cmd` or `reconfig.sh` command, the following error message appears if the default cache directory is not valid:
   >
   > ```
   > *sys-package-mgr*: can't create package cache dir
   > ```
   >
   > You can change the cache directory by including the `-Dpython.cachedir=`*`valid_directory`* option in the command.
   >
   > To create a log file of the Reconfiguration Wizard session, include the `-log=reconfig.log -log_priority=debug` parameter in the command. You can specify any file name for the log file, such as `config_today.log`. The log file is stored in the *`logs`* directory of the Oracle Home directory. Other valid values for `log_priority` are `OFF`, `SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER`, `FINEST`, and `ALL`.

   The Select Domain screen appears.

The Reconfiguration Wizard displays a sequence of screens in the order listed in Table 3-3.

> **Note:**
>
> Depending on the applications in your domain and other factors, extra configuration screens appear in addition to the screens shown in the following table. For information on these screens, click the **Help** button on the screen.
>
> If the Advanced Configuration screen appears during the reconfiguration process, do not select any options to skip all advanced configuration. If necessary, you can use WLST, the Configuration Wizard, or the WebLogic Server Administration Console later to perform advanced configuration such as adding more servers and clusters or changing deployment targeting.

Table 3-3 Reconfiguring an Existing WebLogic Domain

Screen	When Does This Screen Appear?	Perform the Following Action
Select Domain	Always	Enter the full path to the domain directory or click **Browse** to navigate to and select the domain directory. Click **Next** to continue.
Reconfiguration Setup Progress	Always	Shows the progress of the application of reconfiguration templates. When the process completes, click **Next** to continue.
Reconfiguration Summary	Always	Displays the information about the reconfiguration process for all the reconfigured templates. Click **Next** to continue.
Domain Mode and JDK	Always	Domain mode cannot be changed during the reconfiguration. You can change domain mode after the upgrade. **Note:** If you upgrade from WebLogic Server 14.1.1.0.0 or earlier and your domain was in production mode, it will remain in production mode with secured production mode now explicitly disabled in the domain configuration file. However, if you upgrade and then switch to production mode, secured production mode will be enabled. Select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use. Click **Next** to continue.
Additional domain configuration screens may appear at this point	Additional screens depend on the domain configuration	Click the **Help** button on the screen or refer to Reconfiguration Wizard Screens, which describes all the screens in the order in which they are displayed.

Table 3-3 (Cont.) Reconfiguring an Existing WebLogic Domain

Screen	When Does This Screen Appear?	Perform the Following Action
Advanced Configuration	Always	Select the check box for each category (if any) for which you want to perform advanced configuration tasks. The available check boxes depend on the domain configuration. Click **Next** to continue.
Configuration Summary	Always	Review the configuration. Click the **Back** button to change the configuration or click the **Reconfig** button to complete the domain reconfiguration process.
Reconfiguration Success	Always	Shows the final status of the reconfiguration process. Click **Finish** to exit the Configuration Wizard.

Reconfiguring a WebLogic Domain Using WebLogic Scripting Tool

To reconfigure a domain using WLST, you use the `readDomainForUpgrade` command. You can also use this command to migrate an existing per host Node Manager configuration to a per domain configuration.

> **Note:**
>
> If the original domain is using a per domain Node Manager configuration, Node Manager is upgraded automatically and no additional steps are needed.
>
> If the original domain is using a per host Node Manager, and you want to continue using that configuration, you must manually reconfigure Node Manager as described in Completing the Node Manager Configuration.

Example 3-1 shows how to reconfigure a domain called `my_domain` using WLST offline.

Example 3-2 shows how to migrate an existing per host Node Manager configuration to a per domain configuration located in *DOMAIN_HOME*/nodemanager.

Example 3-3 shows how to migrate an existing per host configuration located in `/Oracle/Middleware/oracle_common/common/nodemanager` to a per domain configuration located in `/Oracle/Middleware/custom/nodemanager`.

For information about available Node Manager options for the `setOption` command, see setOption in *WLST Command Reference for Oracle WebLogic Server*. For information about available Node Manager WLST commands, see Node Manager Commands in *WLST Command Reference for Oracle WebLogic Server*.

Example 3-1 Reconfiguring a WebLogic Domain

```
# Open the domain for upgrade.
wls:/offline> readDomainForUpgrade('c:/domains/my_domain')
```

```
# Save the updated domain.
wls:/offline/my_domain> updateDomain()

# Close the domain.
wls:/offline/my_domain> closeDomain()
```

If your existing domain is using a per host Node Manager and you want to move to a per domain Node Manager configuration, you have several options:

- Create a per domain configuration in the default location (`DOMAIN_HOME/nodemanager`) by migrating an existing per host configuration.
- Create a per domain configuration in the default location (`DOMAIN_HOME/nodemanager`) with a new configuration based on Oracle-recommended defaults.
- Create a per domain configuration in a custom location by migrating an existing per host configuration.
- Create a per domain configuration in a custom location with a new configuration based on Oracle-recommended defaults.

Example 3-2 Creating a New Node Manager Configuration in the Default Location

```
#Read domain for reconfiguration
readDomainForUpgrade('domains/mydomain')

#Set Node Manager username and password.
cd('/')
cd('SecurityConfiguration/mydomain')
cmo.setNodeManagerUsername('username')
cmo.setNodeManagerPasswordEncrypted('password')

#Browse Node Manager properties
cd('/')
cd('NMProperties')

# Create per domain Node Manager with new default configuration. Existing
# Node Manager properties will not be migrated in this case.
setOption('NodeManagerType','PerDomainNodeManager')
setOption('NodeManagerUpgradeType','New')

# Update the domain to commit the changes.
updateDomain()
```

Example 3-3 Migrating an Existing Configuration to a Custom Location

```
#Read domain for reconfiguration
readDomainForUpgrade('/domains/mydomain')

#Set Node Manager username and password.
cd('/')
cd('SecurityConfiguration/mydomain')
cmo.setNodeManagerUsername('username')
cmo.setNodeManagerPasswordEncrypted('password')

#Browse node manager properties
cd('/')
cd('NMProperties')

# Create custom location Node Manager, migrating an existing Node Manager
# configuration with default values for Oracle-recommended default properties.
setOption('NodeManagerType','CustomLocationNodeManager')
setOption('NodeManagerHome','/Oracle/Middleware/custom/nodemanager/')
setOption('NodeManagerUpgradeType','Migrate')
setOption('OldNodeManagerHome','/Oracle/Middleware/Oracle_Home/oracle_common/
common/nodemanager')
setOption('NodeManagerUpgradeOverwriteDefault','true')

# Update the domain to commit the changes.
updateDomain()
```

Completing the Node Manager Configuration

If the domain you reconfigured was using a per host Node Manager configuration and you want to continue using a per host Node Manager for the domain, you must complete a set of configuration tasks for Node Manager.

1. In the new WebLogic Server installation, create the `nodemanager` directory in `ORACLE_HOME/oracle_common/common`.
2. Copy the `nodemanager.properties` and `nodemanager.domains` files from the `WL_HOME/common/nodemanager` directory of your previous WebLogic Server installation to the directory you created in Step 1.
3. If your previous installation includes an `nm_data.properties`, `SerializedSystemIni.data`, or `security/SerializedSystemIni.dat` file, copy it to the directory you created in Step 1. If copying SerializedSystemIni.dat, you must create a `security` directory under the `nodemanager` directory and store the file there.
4. Make the following edits to the `nodemanager.properties` file, where `ORACLE_HOME` is the Oracle home directory for your WebLogic Server installation:
   - Update `DomainsFile` to point to `ORACLE_HOME/oracle_common/common/nodemanager/nodemanager.domains` file.
   - Update `JavaHome` to point to the `jre` directory for the JDK that you are using for WebLogic Server. If the file also contains a `javaHome` property setting (lower-case `j`), remove it as it is not needed.
   - Update `NodeManagerHome` to point to `ORACLE_HOME/oracle_common/common/nodemanager`.
   - Update `LogFile` to point to `ORACLE_HOME/oracle_common/common/nodemanager/nodemanager.log`.
5. If you are using your own security certificates, verify that the location of those certificates is correct in `nodemanager.properties`. You may have to update the path if you moved the certificates to another location.

   If you were using the WebLogic Server demo certificate in your previous installation, you must run `CertGen` to create a demo keystore for this installation:

   a. Run `setWLSEnv`:

      `cd` *`WL_HOME`*`/server/bin`

      `. ./setWLSEnv.sh` (UNIX)

      `setWLSEnv.cmd` (Windows)

> **Note:**
>
> On UNIX operating systems, the `setWLSEnv.sh` command does not set the environment variables in all command shells. Oracle recommends that you execute this command using the Korn shell or bash shell.

b. Change to the `ORACLE_HOME/oracle_common/common/nodemanager/` directory and create a security directory if it does not exist.

c. Change to the security directory and enter the following command:

```
java utils.CertGen -certfile democert -keyfile demokey -keyfilepass
DemoIdentityPassPhrase
```

d. To generate the `DemoIdentity.jks` file, enter the following command:

```
java utils.ImportPrivateKey -certfile democert.pem -keyfile demokey.pem -
keyfilepass DemoIdentityPassPhrase -keystore DemoIdentity.jks -storepass
DemoIdentityKeyStorePassPhrase -alias demoidentity
```

6. From the `ORACLE_HOME/wlserver/server/bin` directory, run `startNodeManager.cmd` (Windows) or `startNodeManager.sh` (UNIX).
7. Verify that you can start servers using Node Manager. See Using Node Manager to Control Servers in *Administering Node Manager for Oracle WebLogic Server*. To ensure that your `permgen` settings are adequate for starting the servers, you can use any one of the following methods:
   - Start the Managed Servers using the `startManagedWebLogic` script.
   - Use a `setUserOverrides` script to specify `permgen` settings for server startup. See Customizing Domain Wide Server Parameters in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

Completing the Node Manager Configuration (Two Per Host Node Managers)

If the domain you reconfigured was using a per host Node Manager configuration, you can continue using a per host Node Manager for the 14c domain on a machine that already has a per host Node Manager for the existing domain.

Complete the following steps on each machine in the domain:

> **Note:**
>
> Prior to performing the steps in this section, ensure that you have unpacked the domain to each remote machine in the domain. Include the `–nodemanager_type=ManualNodeManagerSetup` and `-overwrite_domain=true` parameters in the command. For example:
>
> ```
> ./unpack.sh -domain=domain_home -template=template_jar -
> nodemanager_type=ManualNodeManagerSetup -overwrite_domain=true
> ```

1. In the new WebLogic Server installation, create the `nodemanager` directory in *`ORACLE_HOME`*`/oracle_common/common`.
2. Copy the `nodemanager.domains` and `nodemanager.properties` files from the *`WL_HOME`*`/common/nodemanager` directory of your previous WebLogic Server installation to the directory you created in Step 1. If any 12c domains are listed in the `nodemanager.domains` file, delete or comment out those lines.
3. Edit the `nodemanager.properties` file as appropriate on each machine. In particular:
   - Verify that the `SecureListener` is set to `true` if using an SSL Node Manager, or is set to `false` if using a Plain Node Manager.
   - Change `DomainsFile` to point to *`ORACLE_HOME`*`/oracle_common/common/nodemanager/nodemanager.domains`.
   - Change `PropertiesVersion` to 14.1.2.0.0.
   - Change `NodeManagerHome` to *`ORACLE_HOME`*`/oracle_common/common/nodemanager`.
   - Change `JavaHome` to point to the `jre` directory for the Java installation that you are using for WebLogic Server 14.1.2.0.0.
   - Remove the `javaHome` line as it is not needed in 14c.
   - Change `ListenPort` to the value you specified on the Machines screen of the Configuration Wizard.
   - Change `LogFile` to the desired location and file name. The recommended value is *`ORACLE_HOME`*`/oracle_common/common/nodemanager/nodemanager.log`.
4. If you are using your own security certificates, verify that the location of those certificates is correct in `nodemanager.properties`. If you moved the certificates to another location, you have to update the path.

   If you used the WebLogic Server demo certificate in your previous installation, you must run `CertGen` to create a demo keystore for this installation:

   a. Run `setWLSEnv`:

      `cd` *`WL_HOME`*`/server/bin`

      `. ./setWLSEnv.sh` (UNIX)

      `setWLSEnv.cmd` (Windows)

      > **Note:**
      >
      > On UNIX operating systems, the `setWLSEnv.sh` command does not set the environment variables in all command shells. Oracle recommends that you execute this command using the Korn shell or bash shell.

   b. Change to the *`ORACLE_HOME`*`/oracle_common/common/nodemanager/` directory and create a security directory if it does not exist.

   c. Change to the security directory and enter the following command:

      ```
      java utils.CertGen -certfile democert -keyfile demokey -keyfilepass
      DemoIdentityPassPhrase
      ```

   d. To generate the `DemoIdentity.jks` file, enter the following command:

```
java utils.ImportPrivateKey -certfile democert.pem -keyfile demokey.pem -
keyfilepass DemoIdentityPassPhrase -keystore DemoIdentity.jks -storepass
DemoIdentityKeyStorePassPhrase -alias demoidentity
```

5. From the *ORACLE_HOME*/wlserver/server/bin directory, start Node Manager.
6. If the Administration Server is running, restart the Administration Server.
7. Verify that you can start servers using Node Manager. See Using Node Manager to Control Servers in *Administering Node Manager for Oracle WebLogic Server*. To ensure that your permgen settings are adequate for starting the servers, you can use any one of the following methods:
   - Start the Managed Servers using the startManagedWebLogic script.
   - Set permgen space
   - Use a setUserOverrides script to specify permgen settings for server startup. See Customizing Domain Wide Server Parameters in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

Updating a Managed Server Domain on a Remote Machine

If your WebLogic domain contains multiple Managed Servers, and each Managed Server domain directory is located on a machine that is remote to the Administration Server host machine, you can use one of two methods to update the domain on the remote machine.

- Use the pack command to generate the domain template JAR. Ensure that you include the -managed=true argument in the pack command. Move the JAR to the remote machine and then use the unpack command on the remote machine to create the Managed Server domain. See Creating Templates and Domains Using the Pack and Unpack Commands.
- Use the WLST writeTemplate command in online mode. When you execute the writeTemplate command while connected to the Administration Server from a remote machine, it dynamically packs the domain on the Administration Server into a template JAR file and transfers the template JAR to the specified directory.

  The following sample WLST script demonstrates how to use writeTemplate to create or update a Managed Server domain on a remote machine. Run the script on each remote machine in the domain. This script does the following tasks:
  - logs in to the Administration Server
  - packs the Administration Server domain into a JAR file and writes it to the specified template directory on the remote machine
  - disconnects from the Administration Server
  - reads the template JAR
  - creates the domain on the remote machine

```
import os

wlsHome = os.getenv('WL_HOME')
mwHome = os.path.join(wlsHome, '..')

#Substitute the administrator user name and password values below as needed
connect('adminuser','adminpassword','admin_server_url')

#Create the path on the local machine where the template will be stored,
#The specified template JAR should not already exist. The timeout value
#specifies the number of milliseconds to elapse before the connection between
```

```
#the Administration Server and remote machine times out (default is 120000).
templatePath = '/user_templates/myTemplate.jar'
timeout = 180000

#get the packed template from the Administration Server
writeTemplate(templatePath, timeout)

#disconnect from online WLST connection to the Administration Server
disconnect()

#read the template that was downloaded from the Administration Server
readTemplate(templatePath)

#specify the domain directory where the domain needs to be created
domainPath = 'domains/myDomain'

#create the domain
writeDomain(domainPath)
```

Important Notes About the Domain Upgrade Process

Bear in mind several key notes about the domain upgrade process, such as whether it is necessary to undeploy WebLogic Server applications, the minimum set of files that must exist in the domain directory, and more.

- It is not always necessary to undeploy WebLogic Server applications. Usually, WebLogic Server applications can run without modifications in the new WebLogic Server 14.1.2.0.0 application environment. Run the WebLogic Migration Analysis Tool before an upgrade to identify classes and APIs are no longer in the new version and reports any that changes are required for the application to successfully deploy. For more information, see Identifying Unused APIs .
- At a minimum, the domain directory must contain the following files:
  - `config.xml`
  - Security-related files, including `SerializedSystemIni.dat`, `DefaultAuthenticatorInit.ldift`, `DefaultAuthorizerInit.ldift`, and `DefaultRoleMapperInit.ldift`

    If the security-related files are not available, the server fails to start and an authentication error message is logged.
  - Any transaction log (`.tlog`) files that reside in the domain. See Using Transaction Log Files to Recover Transactions in *Developing JTA Applications for Oracle WebLogic Server*.
- All contents of the domain directory on the target computer are updated during this process.
- You must upgrade the domain on every computer in the application environment.
- The Reconfiguration Wizard does not upgrade your own applications that may exist in the domain during a WebLogic domain upgrade.
- Domains that contain resources for WebLogic Liquid Data, or AquaLogic Data Services Platform cannot be upgraded to WebLogic Server 14.1.2.0.0.

Completing Post-Upgrade Tasks

After you upgrade the application environment, it may be necessary to perform tasks such re-applying customizations to startup scripts, verifying file permissions and remote server startup options, and more.

This section includes the following topics:

- Changing Domain Mode Post Upgrade
- Identifying Unused APIs
- Re-apply Customizations to Startup Scripts
- Verify File Permissions
- Verify Remote Server Startup Options
- Recreating the Windows Node Manager Service
- Promote the Application Environment to Production

Not all these steps are required for all situations. Review the sections to determine which, if any, of these steps are appropriate for your environment.

Changing Domain Mode Post Upgrade

After the upgrade, your domain retains its original pre-upgrade domain security mode settings. If you want to change the domain mode, to enable enhanced security, for example, you must explicitly change the settings using the WebLogic Remote Console or by modifying the `DomainMBean`.

If your domain is currently set to Production Mode, and you want to enable added security, then after the upgrade use the WebLogic Remote Console to change the domain mode and enable the Secured Production Mode. Change the Domain Mode in *Oracle WebLogic Remote Console Online Help*.

> **Caution:**
>
> Changes to the domain mode require a full domain restart - a rolling restart is not sufficient. You must stop all managed servers before you attempt to change the domain mode.

When upgrading a domain to 14c (14.1.2.0.0), if there is no explicit secure mode setting, then the Reconfiguration Wizard will explicitly set secure mode to *disabled* in the upgraded domain. This is to preserve the behavior that was present in the original domain. If there is an explicit secure mode setting, it will be preserved in the upgraded domain. For more information, see Understand How Domain Mode Affects the Default Security Configuration in *Securing a Production Environment for Oracle WebLogic Server*.

Note:

Secured Production Mode enforces more restrictive and stringent security settings to ensure less vulnerability to threats. To make sure that your domain is secure, after enabling Secured Production Mode, you will have to choose the security configuration options that are appropriate for the environment in which the domain runs, such as obtaining and storing certificates, protecting user accounts, and securing the network on which the domain runs. If these options are not properly configured, you will be blocked from using WebLogic Server.

After you have created your WebLogic domain, several key steps remain to ensure its integrity such as selecting appropriate security configurations. For more information, see Securing the Domain After You Have Created It in *Administering Security for Oracle WebLogic Server*.

Identifying Unused APIs

It is important to know which classes and APIs are no longer in the new version and to address any changes required for the application to successfully deploy. Use the WebLogic Migration Analysis Tool to identify these classes and APIs.

The WebLogic Migration Analysis Tool is a command line utility that identifies APIs in a WebLogic application that have been removed or are no longer used in WebLogic Server 14.1.2.0.0 and generates a report. The report is meant to show only those deletions that are on the classpath of a running WebLogic Server (such as weblogic.jar). The reports are not intended to report on all jar files that are missing in a class - just those associated with the WebLogic applications.

Note:

This report will not catch all potential problems associated with missing or deprecated APIs or classes. For example, if you are using reflection, this report will not detect it. Similarly, this report may state that a library has been removed and could cause and issue, but you may have your own copy of that library somewhere else.

To run the Migration Analysis Tool, execute the following:

```
java -jar $WL_HOME/server/lib/weblogic.migration-analysis-tool.jar <archive-
file-name> <archive-2> <archive-3>
```

Where <*archive-file-name*> parameters are the files you want to analyze. For example:

```
java -jar $WL_HOME/server/lib/weblogic.migration-analysis-tool.jar /tmp/
em_example.war
```

Re-apply Customizations to Startup Scripts

To complete the upgrade of your application environment to 14.1.2.0.0, it might be necessary to re-apply any customizations to startup scripts. The following sections describe how to customize the default startup scripts as well as any custom startup scripts.

Default Startup Scripts

The Reconfiguration Wizard does not carry forward any customizations that have been made to the default startup scripts, such as the setting of the `JAVA_OPTIONS` environment variable. After the upgrade process is complete, you must customize the default scripts again.

Custom Startup Scripts

To update custom startup scripts:

- Set the JDK version to the JDK that you are using with WebLogic Server.
- Update the CLASSPATH variable, as follows:
  - Add WebLogic Server 14.1.2.0.0 classes to the beginning of the variable.
  - Remove all *unused* WebLogic classes prior to version 10.3.

Verify File Permissions

Verify the file permissions, as follows:

- If you backed up the domain directory as part of the upgrade, you must make your backup files secure because they might contain confidential information.
- During the upgrade process, file permissions are not preserved. If nondefault file permissions are set on files, they must be verified and reset.
- On a UNIX system, ownership and permissions for any new files created during the upgrade process are assigned to the user performing the upgrade. For example, if the upgrade is performed by root, then root is assigned ownership of any new files. As a result, any user who subsequently wants to update these files in the domain must have root privileges. You may want to review or modify the permissions on files created during the upgrade process.

Verify Remote Server Startup Options

When you start the Administration Server, verify that the remote server start options, such as `JAVA_HOME`, `BEA_HOME`, and `CLASSPATH`, reference the WebLogic Server installation on the target Managed Server. This can be accomplished using the WebLogic Server Remote Console, as described in Configure startup arguments for Managed Servers in *Oracle WebLogic Server Remote Console Online Help*.

Note:

If the remote server startup options are not set correctly, when attempting to start a Managed Server using Node Manager, messages similar to the following may be written to the log file. Because these messages may be sent recursively, they may eventually consume all space available on the drive.

```
No config.xml was found.

Would you like the server to create a default configuration and boot? (y/n):
java.io.IOException: The handle is invalid
```

Recreating the Windows Node Manager Service

On Windows systems, if you were running Node Manager as a Windows service for your domain, you must reconfigure it if you want to continue using it.

For information about how to configure the Node Manager service for Windows, see Default Node Manager Configuration in *Administering Node Manager for Oracle WebLogic Server*.

Optionally, you can remove the Node Manager service from your installation by running `uninstallNodeMgrSrv.cmd`. See Default Node Manager Configuration in *Administering Node Manager for Oracle WebLogic Server*.

Promote the Application Environment to Production

Execute standard procedures for quality assurance and performance tuning before promoting an application environment to production. You should test the execution of your applications (including external client applications) in your test application environment. If your applications use APIs that have been deprecated or removed, then you may encounter warnings or exceptions at run time. If you do, you can make any required modifications before promoting your applications to production.

When all test criteria have been met, you can promote the application environment to production, as outlined in your upgrade plan (defined previously in Step 4: Create an Upgrade Plan).

When the new 14.1.2.0.0 application environment is deployed into production, you can start redirecting requests to the new environment from the existing environment. Gradually, you can bring the existing environment to a safe state for shutdown. This might be accomplished using a load balancer, for example.

Maintain FIPS 140-2 Compliance

As of WebLogic Server 14.1.2.0.0, the FIPS 140-2 compliant implementation of WebLogic Server changed to rely on the Jipher JCE and SunJSSE providers rather than the Dell JCE and Dell BSAFE JSSE providers (previously known as RSA JCE and RSA BSAFE JSSE).

If you wish to continue using FIPS mode following your upgrade, then you must update your environment to use the Jipher JCE and SunJSSE providers. See Enabling FIPS Mode with Jipher JCE and Sun JSSE Providers in *Administering Security for Oracle WebLogic Server*.

Additionally, you should remove the Dell JCE and Dell BSAFE JSSE providers from your WebLogic Server environment. See Removing Dell JCE and Dell BSAFE JSSE Providers in *Administering Security for Oracle WebLogic Server*.

4

Upgrading WebLogic Web Services

Learn the procedures for upgrading WebLogic and RESTful web services from Oracle WebLogic Server 10.*x* to 14c.
This chapter includes the following sections:

- Upgrading a 10.3.*x* RESTful Web Service (JAX-RS) to 14c
- Upgrading a 10.x WebLogic Web Service (JAX-WS) to 14c
- Upgrading a WebLogic JAX-RPC Web Service to the WebLogic JAX-WS Stack

> **Note:**
>
> - The WebLogic web services (JAX-WS) 10.3.*x* will continue to run without any changes on WebLogic Server 14c, as the associated web services run time is still supported in this release, although they are deprecated and will be removed from the product in future releases. For this reason, Oracle highly recommends that you follow the instructions in this chapter to upgrade your 10.3.*x* web services to 14c.
> - Support for JAX_RPC has been removed as of 14.1.2.0.0,

Upgrading a 10.3.*x* RESTful Web Service (JAX-RS) to 14c

In 10.3.*x*, a set of pre-built shared libraries were delivered with WebLogic Server to support Jersey 1.9 and 1.1.5.1 Java API for RESTful Web Services (JAX-RS) Reference Implementations (RIs). In 14c, WebLogic Server supports Jersey 2.29 (JAX-RS 2.1 RI) by default.To use the pre-built shared libraries of 10.3.*x*, you needed to register them with the WebLogic Server instance, and modify the `web.xml` and `weblogic.xml` deployment descriptors to use the Jersey servlet and reference the shared libraries, respectively.
In 14c, as WebLogic Server supports Jersey 2.29 (JAX-RS 2.1 RI) by default, registration as a shared library with WebLogic Server is no longer required.

To use the Jersey 2.29 (JAX-RS 2.1 RI), you need to modify your 10.3.*x* RESTful Web service applications as follows:

1. Update your application deployment descriptors to reference the Jersey 2.*x* container. See Servlet-based Deployment in *Jersey 2.29 User Guide*.

   > **Note:**
   >
   > For backward compatibility, references to `com.sun.jersey.spi.container.servlet.ServletContainer`, as shown in the following example, continues to work. However, Oracle recommends that you update your application deployment descriptors to reference the Jersey 2.*x* container instead.

For example, replace `com.sun.jersey.spi.container.servlet.ServletContainer` with `org.glassfish.jersey.servlet.ServletContainer` in the following `<web-app>` content:

```
<web-app>
    <servlet>
        <display-name>My Jersey Application</display-name>
        <servlet-name>MyJerseyApp</servlet-name>
        <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</
servlet-class>
        <init-param>
            <param-name>javax.ws.rs.Application</param-name>
            <param-value>myPackage.myJerseyApplication</param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>MyJerseyApp</servlet-name>
        <url-pattern>/*</url-pattern>
    </servlet-mapping>
</web-app>
```

For more advanced configuration options, see *Jersey 2.29 User Guide*.

2. If applicable, update all applications that use Jersey 1.*x* server APIs to use the corresponding standard JAX-RS 2.1 or Jersey 2.*x* APIs instead. Support for the Jersey 1.*x* (JAX-RS 1.1 RI) *server* APIs has been removed in this release and applications that reference them will not work.
3. Update your clients to use the `javax.ws.rs.client` API, as described in Developing RESTful Web Service Clients in *Developing and Securing RESTful Web Services for Oracle WebLogic Server*.

> **Note:**
>
> Support for the Jersey 1.18 client packages, including the `com.sun.jersey`package, its nested packages, and the `weblogic.jaxrs.api.client` package, is deprecated in this release of WebLogic Server, but are maintained for backward compatibility. However, many Fusion Middleware components, such as Oracle Web Services Manager, have been migrated to the standard JAX-RS 2.0 client API and are not compatible with the Jersey 1.*x* JAX-RS client APIs. Therefore, Oracle strongly recommends that you update your RESTful client applications as soon as possible to use the standard JAX-RS 2.0 API.
>
> The Jersey 1.*x* JAX-RS RI client APIs are not compatible with Jersey 2.*x* (JAX-RS 2.0 RI).

Upgrading a 10.x WebLogic Web Service (JAX-WS) to 14c

No steps are required to upgrade a 10.*x* WebLogic web service to 14c. You can redeploy the JAX-WS Web service to WebLogic Server 14c without making any changes or recompiling.

Upgrading a WebLogic JAX-RPC Web Service to the WebLogic JAX-WS Stack

The WebLogic JAX-WS run time is based on the JAX-WS (The Java API for XML-Based Web Services) 2.2 specification and the Web Services for Java EE v1.3 (JSR 109) specifications. Starting with JAX-WS 2.0, the JAX-WS technology has replaced JAX-RPC in the Java Platform and in WebLogic Server. JAX-RPC Web Services in WebLogic applications should be upgraded to JAX-WS.

> **Note:**
>
> The JAX-RPC API has been deprecated in 12.2.x and will be removed in a future release. Oracle does not recommend upgrading to the JAX-RPC stack.

This section summarizes how to upgrade a WebLogic JAX-RPC Web service to use the WebLogic JAX-WS stack.

Upgrading your WebLogic Server JAX-RPC Web service includes the following high-level tasks:

- Upgrade any Web service EJBs from 2.*x* to 3.*x*.

  JAX-WS supports EJB 3.0 and 3.*x*. It does not support EJB 2.*x*.
- Upgrade your JWS, mapping any proprietary JAX-RPC features to similar JAX-WS features.

  Note that there is not a one-to-one correspondence between WebLogic JAX-RPC Web service features and JAX-WS 12.*x* features.
- Update the Ant build script that builds the Web service to change the value of the `type` attribute on the `jws`, `wsdlc`, and `clientgen` tasks to be `"JAXWS"` (for example, `type="JAXWS"`).
- Generate new JAX-WS clients using the JAX-WS `clientgen` Ant task.

JAX-WS Upgrade Considerations

When upgrading to JAX-WS, you should consider the following:

- The JAX-WS specification supports the *document-literal* and *rpc-literal* styles, but not *rpc-encoded*.
- SOAP Arrays are not supported by JAX-WS.

See *Developing JAX-WS Web Services for Oracle WebLogic Server*.

5

WebLogic Server 14.1.2.0.0 Compatibility with Previous Releases

Learn about important compatibility information that you should consider before upgrading to Oracle WebLogic Server 14.1.2.0.0 from 14.1.1.0.0 (standalone WebLogic Server) or 12.2.1.4 release. Also learn about the feature changes in various Oracle WebLogic Server versions that may impact the applications you plan to run in the upgraded environment.
See also:

- WebLogic Server Compatibility in *Understanding Oracle WebLogic Server*. This section provides general information about WebLogic Server compatibility goals and how they apply to this WebLogic Server release.
- *What's New in Oracle WebLogic Server 14.1.2.0.0* for this and prior releases. These documents provide information about new features that are available to you as well as behavior changes that may impact your applications.

About WebLogic Server Cluster Messaging

In 12.2.1.4.0, WebLogic Server cluster messaging was enhanced. If all of the servers in a cluster are running the same installation version of WebLogic Server, no changes are required.

When you perform an upgrade from 12.2.1.4.0 or 14.1.1.0.0, the upgraded servers must be explicitly set, to temporarily allow the older protocol. You can do this by setting the system property `weblogic.upgradeExpirationDate` with an expiration date, which enables the upgraded server to allow communication on the cluster until that expiration date and time is reached. For example:

```
-Dweblogic.upgradeExpirationDate=2024-01-05T08:47
```

If you want the clusters that are at different versions, to continue to communicate for an extended period of time, you must set the value to a preferred future upgrade date.

> **Note:**
>
> The system property `-Dweblogic.upgradeExpirationDate` must be used in the Server Start arguments for each of the Managed Servers, and not in the `JAVA_OPTIONS` environment variable in the `startWebLogic.sh` or `startWebLogic.cmd` scripts.

Index