

Oracle® Fusion Middleware

Installing and Configuring Oracle Data Integrator



14c (14.1.2.0.0)

F78368-01

December 2024

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

ORACLE®

Oracle Fusion Middleware Installing and Configuring Oracle Data Integrator, 14c (14.1.2.0.0)

F78368-01

Copyright © 2015, 2024, Oracle and/or its affiliates.

Contributing Authors: 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, MySQL, and NetSuite 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	ix
Documentation Accessibility	ix
Diversity and Inclusion	ix
Related Documents	ix
Conventions	x

1 About the Oracle Data Integrator Installation

Using the Standard Installation Topology As a Starting Point	1-1
About the Oracle Data Integrator Standalone Agent Standard Installation Topology	1-1
About the Oracle Data Integrator Standalone Collocated Agent Standard Installation Topology	1-2
About the Oracle Data Integrator Java EE Agent Standard Installation Topology	1-3
About the Oracle Data Integrator Topology	1-5
About Oracle Data Integrator Runtime Agents	1-5
About Oracle Data Integrator Studio	1-6
Using This Document to Extend an Existing Domain	1-6

2 Preparing to Install and Configure Oracle Data Integrator

Roadmap for Installing and Configuring a Standard Installation Topology	2-1
Roadmap for Verifying Your System Environment	2-2
Verifying Certification, System, and Interoperability Requirements	2-3
Selecting an Installation User	2-3
About User Permissions	2-3
About Non-Default User Permissions on Linux or UNIX Operating Systems	2-5
Verifying that the Installation User has Administrator Privileges on Windows Operating Systems	2-6
About the Directories for Installation and Configuration	2-6
About the Recommended Directory Structure	2-6
About the Oracle Home Directory	2-7
About the Domain Home Directory	2-8
About the Application Home Directory	2-8

Installing Multiple Products in the Same Domain	2-9
Preparing for Shared Storage	2-10
About JDK Requirements for an Oracle Fusion Middleware Installation	2-10
About Database Requirements for an Oracle Fusion Middleware Installation	2-10
About Product Distributions	2-11
Obtaining the Product Distribution	2-11
Verifying Digital Signature and Integrity of Installation Archive Files	2-12

3 Installing the Oracle Data Integrator Software

Starting the Installation Program	3-1
Navigating the Installation Screens	3-1
Verifying the Installation	3-2
Reviewing the Installation Log Files	3-2
Checking the Directory Structure	3-3
Viewing the Contents of the Oracle Home	3-3
Installing Oracle Data Integrator in Silent Mode	3-3

4 Creating the Master and Work Repository Schemas

Installing and Configuring a Certified Database	4-1
Starting the Repository Creation Utility	4-1
Navigating the Repository Creation Utility Screens to Create Schemas	4-2
Introducing the RCU	4-2
Selecting a Method of Schema Creation	4-2
Providing Database Connection Details	4-2
Specifying a Custom Prefix and Selecting Schemas	4-3
Specifying Schema Passwords	4-3
Specifying Custom Variables	4-3
Completing Schema Creation	4-3

5 Configuring Oracle Data Integrator Studio

About Oracle Data Integrator Studio	5-1
Creating the Database Schemas	5-1
Setting the Same Time Zone for the Agent and Repository	5-1
Adding Libraries to ODI Studio	5-2
Setting Java Home for ODI Studio	5-3
Starting ODI Studio	5-3
Connecting to the Master Repository	5-3
Add Proxy Configuration for ODI Studio	5-5

6 Configuring the Domain for a Standalone Agent

Creating the Database Schemas	6-1
Creating an Agent in the Master Repository with ODI Studio	6-1
Configuring the Domain	6-2
Starting the Configuration Wizard	6-2
Navigating the Configuration Wizard Screens to Create and Configure the Domain	6-2
Selecting the Configuration Type and Domain Home Location	6-3
Selecting the Configuration Template	6-3
Specifying the Domain Mode and JDK	6-3
Specifying the Database Configuration Type	6-3
Specifying JDBC Component Schema Information	6-4
Testing the JDBC Connections	6-5
Creating a New ODI System Component	6-5
Configuring the ODI Server Access	6-5
Configuring Node Manager	6-6
Reviewing Your Configuration Specifications and Configuring the Domain	6-6
Writing Down Your Domain Home and Administration Server URL	6-6
Adding Libraries to a Standalone Agent	6-6
Starting a Standalone Agent Using Node Manager	6-7
Starting Node Manager	6-7
Starting a Standalone Agent	6-7
Starting a Standalone Agent Without Node Manager	6-8
Verifying the Configuration	6-8
Add Proxy Configuration for Standalone Agent	6-8

7 Configuring the Domain for a Standalone Collocated Agent

Creating the Database Schemas	7-1
Creating an Agent in the Master Repository with ODI Studio	7-1
Configuring the Domain	7-2
Starting the Configuration Wizard	7-2
Navigating the Configuration Wizard Screens to Create and Configure the Domain	7-2
Selecting the Configuration Type and Domain Home Location	7-3
Selecting the Configuration Template	7-3
Configuring the Administrator Account	7-3
Specifying the Domain Mode and JDK	7-3
Specifying the Database Configuration Type	7-4
Specifying JDBC Component Schema Information	7-5
Testing the JDBC Connections	7-6
Selecting Advanced Configuration	7-6
Configuring the Administration Server Listen Address	7-6

Configuring Node Manager	7-7
Creating a New ODI System Component	7-7
Configuring the ODI Server Access	7-7
Specifying the Node Manager Host Address	7-8
Assigning System Components to Machines	7-8
Reviewing Your Configuration Specifications and Configuring the Domain	7-8
Writing Down Your Domain Home and Administration Server URL	7-8
Adding Libraries to a Standalone Collocated Agent	7-9
Starting a Standalone Collocated Agent Using Node Manager	7-9
Starting Node Manager	7-9
Starting a Standalone Collocated Agent	7-10
Starting a Standalone Collocated Agent Without Node Manager	7-10
Verifying the Configuration	7-10

8 Configuring the Domain for a Java EE Agent

Creating the Database Schemas	8-1
Generating a Template for a Java EE Agent	8-1
Creating an Agent in the Master Repository with ODI Studio	8-2
Configuring the Domain	8-3
Starting the Configuration Wizard	8-3
Navigating the Configuration Wizard Screens to Create and Configure the Domain	8-3
Selecting the Configuration Type and Domain Home Location	8-4
Selecting the Configuration Template	8-4
Selecting the Application Home Location	8-5
Configuring the Administrator Account	8-5
Specifying the Domain Mode and JDK	8-5
Specifying the Database Configuration Type	8-6
Specifying JDBC Component Schema Information	8-7
Testing the JDBC Connections	8-7
Creating Credential Keys	8-8
Selecting Advanced Configuration	8-8
Configuring the Administration Server Listen Address	8-9
Configuring Node Manager	8-9
Configuring Managed Servers for Oracle Data Integrator	8-9
Configuring a Cluster for Oracle Data Integrator	8-10
Defining Server Templates	8-11
Configuring Dynamic Servers	8-11
Assigning Oracle Data Integrator Managed Servers to the Cluster	8-12
Configuring Coherence Clusters	8-12
Creating a New Oracle Data Integrator Machine	8-13
Assigning Servers to Oracle Data Integrator Machines	8-13

Reviewing Your Configuration Specifications and Configuring the Domain	8-14
Writing Down Your Domain Home and Administration Server URL	8-14
Adding Libraries to a Java EE Agent	8-14
Starting the Servers	8-15
Starting Node Manager	8-16
Starting the Administration Server	8-17
Starting the Managed Servers	8-18
Add Proxy Configuration for JEE Agent	8-18

9 Next Steps After Configuring the Domain

Performing Basic Administrative Tasks	9-1
Performing Additional Domain Configuration Tasks	9-2
Preparing Oracle Data Integrator for Development	9-2
Preparing Your Environment for High Availability	9-3

10 Uninstalling or Reinstalling Oracle Data Integrator

Stopping the Oracle Data Integrator Agents	10-1
AgentStop Command Parameters	10-1
Stopping Oracle Fusion Middleware	10-2
Removing Your Database Schemas	10-2
Uninstalling the Software	10-3
Starting the Uninstall Wizard	10-3
Selecting the Product to Uninstall	10-3
Navigating the Uninstall Wizard Screens	10-3
Removing the Oracle Home Directory Manually	10-4
Removing the Program Shortcuts on Windows Operating Systems	10-4
Removing the Domain and Application Data	10-4
Removing Oracle Data Integrator Wallet Information	10-5
Reinstalling the Software	10-5

A Troubleshooting Oracle Data Integrator

General Troubleshooting Tips	A-1
Troubleshooting Using Installation and Configuration Log Files	A-1
Need More Help?	A-2

B Secondary Topologies for Oracle Data Integrator

Oracle Data Integrator Topology with the Oracle Data Integrator Console	B-1
---	-----

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

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

Preface

This document describes how to use the Oracle Data Integrator installation program to install the Oracle Data Integrator software and configure a WebLogic domain.

If you are an existing Oracle Data Integrator user, your path to obtain the most recent version of the software will depend on the version of your existing software. For more information, refer to the [Oracle Fusion Middleware Download, Installation, and Configuration Readme Files](#) for the steps you need to take to update or upgrade your existing software.

Audience

This guide is intended for administrators who are responsible for installing and configuring components of Oracle Data Integrator. It is assumed that readers are comfortable running some system administration operations, such as creating users and groups, adding users to groups, and installing operating system patches on the computer where your products will be installed. Users in UNIX systems who are installing need `root` access to run some scripts.

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.

Related Documents

Refer to the Oracle Fusion Middleware Library for additional information.

- For Oracle Data Integrator information, see Oracle Data Integrator Documentation.

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

Conventions

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.

1

About the Oracle Data Integrator Installation

The standard installation for Oracle Data Integrator described in this guide creates the standard topology, which represents a sample starting topology for this product. This guide explains how to install Oracle Data Integrator (ODI) on your local system using your on-premises hardware.

Using the Standard Installation Topology As a Starting Point

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

The information in this guide helps you to create a standard installation topology for Oracle Data Integrator. If required, you can later extend the standard installation topology to create a secure and highly available production environment, see [Next Steps After Configuring the Domain](#).

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

About the Oracle Data Integrator Standalone Agent Standard Installation Topology

This topology represents a standalone agent configured in a standalone domain, managed by the WebLogic Management Framework. The domain is configured on a single host and requires a supported database where the required Fusion Middleware schemas are installed.

For configuration instructions, see [Configuring the Domain for a Standalone Agent](#).

[Table 1-1](#) describes all elements in this standard installation topology illustration.

Table 1-1 Description of the Elements in the Standalone Agent Standard Installation Topology

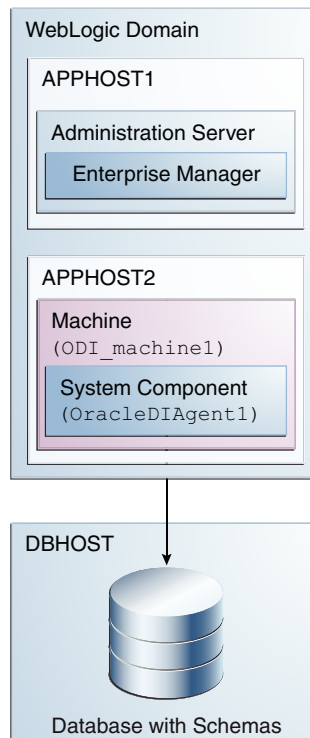
Element	Description and Links to Additional Documentation
APPHOST	Standard term used in Oracle documentation referring to the computer that is hosting the application tier.
DBHOST	Standard term used in Oracle documentation referring to the computer that is hosting the database.
Standalone Domain	A container for system components, such as Oracle HTTP Server or Oracle Data Integrator standalone agents. See <i>What Is a Standalone Domain?</i> in <i>Understanding Oracle Fusion Middleware</i> .
System Component	A manageable process that is not deployed in a Java application container. A system component corresponds to a standalone agent. See <i>What Is a System Component?</i> in <i>Understanding Oracle Fusion Middleware</i> .

About the Oracle Data Integrator Standalone Collocated Agent Standard Installation Topology

This installation topology represents a standalone agent configured in a WebLogic domain, managed by an Administration Server and Node Manager. The domain is configured on a single host and requires a supported database where the required Fusion Middleware schemas are installed.

Figure 1-1 shows the standard installation topology for the standalone collocated agent.

Figure 1-1 Standard Installation Topology for the Standalone Collocated Agent



For configuration instructions, see [Configuring the Domain for a Standalone Collocated Agent](#).

Table 1-2 describes all elements in this standard installation topology.

Table 1-2 Description of the Elements in the Standalone Collocated Agent Standard Installation Topology

Element	Description and Links to Additional Documentation
APPHOST	Standard term used in Oracle documentation referring to the computer that is hosting the application tier.
DBHOST	Standard term used in Oracle documentation referring to the computer that is hosting the database.

Table 1-2 (Cont.) Description of the Elements in the Standalone Collocated Agent Standard Installation Topology

Element	Description and Links to Additional Documentation
Administration Server	The central control entity of a domain which maintains the domain's configuration objects and distributes configuration changes to Managed Servers. See <i>What Is the Administration Server?</i> in <i>Understanding Oracle Fusion Middleware</i> .
Enterprise Manager	Oracle Enterprise Manager Fusion Middleware Control. See <i>Oracle Enterprise Manager Fusion Middleware Control</i> in <i>Understanding Oracle Fusion Middleware</i> .
Machine	Logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are also the logical glue between WebLogic Managed Servers and the Node Manager; in order to start or stop a Managed Server with Node Manager, the Managed Server must be associated with a machine.
System Component	A standalone process that is managed by the WebLogic Management Framework. A system component corresponds to a standalone agent. See <i>What Is a System Component?</i> in <i>Understanding Oracle Fusion Middleware</i> .

About the Oracle Data Integrator Java EE Agent Standard Installation Topology

This topology represents a standard WebLogic Server domain that contains an Administration Server and a Managed Server on which the Infrastructure and Java EE agent are deployed.

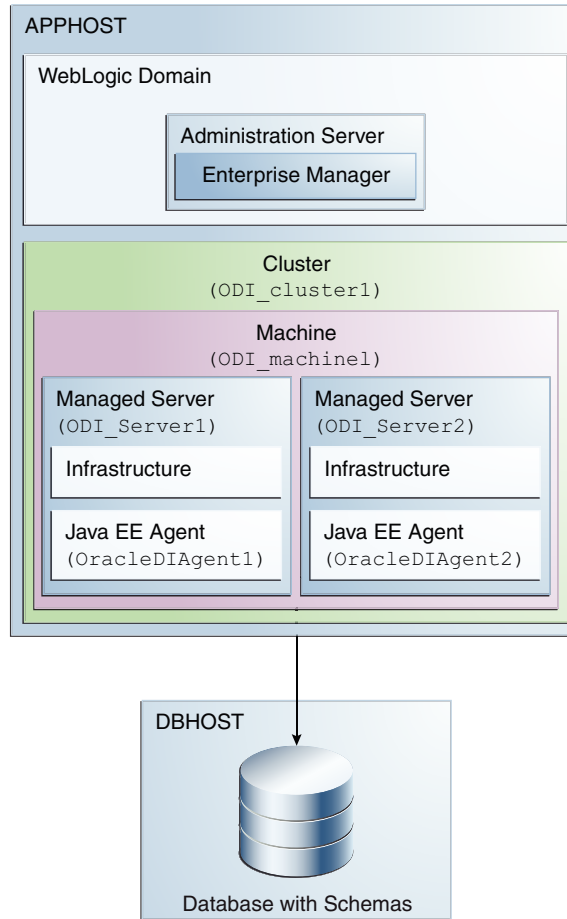
The Managed Server is targeted to a machine inside a cluster. The domain is configured on a single host and requires a supported database where the required Fusion Middleware schemas are installed. The Java EE agent is installed into an existing Oracle home containing Oracle Fusion Middleware Infrastructure. ODI Console, ODI Studio, and the plugin for Fusion Middleware Control are also installed in the topology.

Note:

The infrastructure can use more than one machine for scalability and high-availability considerations. See [Preparing Your Environment for High Availability](#) for more information.

Figure 1-2 shows the standard installation topology for an Oracle Data Integrator Java EE agent.

Figure 1-2 Standard Installation Topology for an ODI Java EE Agent



For configuration instructions, see [Configuring the Domain for a Java EE Agent](#).

[Table 1-3](#) describes all elements in this standard installation topology illustration.

Table 1-3 Description of the Elements in the Java EE Standard Installation Topology

Element	Description and Links to Additional Documentation
APPHOST	Standard term used in Oracle documentation referring to the computer that is hosting the application tier.
DBHOST	Standard term used in Oracle documentation referring to the computer that is hosting the database.
WebLogic Domain	A logically related group of Java components (in this case, the administration Server, Managed Servers, and other related software components). See <i>What Is an Oracle WebLogic Server Domain?</i> in <i>Understanding Oracle Fusion Middleware</i> .
Administration Server	The central control entity of a domain which maintains the domain's configuration objects and distributes configuration changes to Managed Servers. See <i>What Is the Administration Server?</i> in <i>Understanding Oracle Fusion Middleware</i> .

Table 1-3 (Cont.) Description of the Elements in the Java EE Standard Installation Topology

Element	Description and Links to Additional Documentation
Enterprise Manager	Oracle Enterprise Manager Fusion Middleware Control. See Oracle Enterprise Manager Fusion Middleware Control in <i>Understanding Oracle Fusion Middleware</i> .
Cluster	A collection of multiple WebLogic Server instances running simultaneously and working together. See Overview of Managed Servers and Managed Server Clusters in <i>Understanding Oracle Fusion Middleware</i> .
Machine	Logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are also the logical glue between WebLogic Managed Servers and the Node Manager; in order to start or stop a Managed Server with Node Manager, the Managed Server must be associated with a machine.
Managed Server	Host for your applications, application components, Web services, and their associated resources. See Overview of Managed Servers and Managed Server Clusters in <i>Understanding Oracle Fusion Middleware</i> .
Infrastructure	Collection of services that include the following: <ul style="list-style-type: none"> • Metadata repository (MDS). This contains metadata for Oracle Fusion Middleware components, such as the Oracle Application Developer Framework. See What Is the Metadata Repository? in <i>Understanding Oracle Fusion Middleware</i>. • Oracle Application Developer Framework (Oracle ADF) • Oracle Web Services Manager (OWSM)
Java EE Agent	The Java EE run-time agent.

About the Oracle Data Integrator Topology

The Oracle Data Integrator topology is the physical and logical representation of the Oracle Data Integrator architecture and components.

Before you can begin developing using Oracle Data Integrator, you must first set up an Oracle Data Integrator topology, which is the physical and logical representation of the Oracle Data Integrator architecture and components. The Oracle Data Integrator topology defines where to find the sources and targets of the data that you are integrating.

Note that this topology is not the same as the standard installation topology, which is the layout of files installed and configured on your computer by the Oracle Universal Installer and the configuration wizard.

See Setting up a Topology in *Developing Integration Projects with Oracle Data Integrator*.

About Oracle Data Integrator Runtime Agents

At design time, developers generate scenarios from the business rules that they have designed. The code of these scenarios is then retrieved from the repository by a runtime agent. This agent then connects to the data servers and orchestrates the code execution on these servers.

The types of run-time agents that can be configured in the ODI installation topology are:

- Standalone agent
- Standalone collocated agent
- Java EE agent

See Runtime Agent in *Understanding Oracle Data Integrator*.

About Oracle Data Integrator Studio

Oracle Data Integrator Studio is a developer's interface for configuring and managing ODI.

To learn how to use ODI Studio to configure and manage your ODI topology, see [Configuring Oracle Data Integrator Studio](#).

Using This Document to Extend an Existing Domain

The procedures in this guide describe how to create a new domain. The assumption is that no other Oracle Fusion Middleware products are installed on your system.

If you have installed and configured other Oracle Fusion Middleware products on your system (for example, Fusion Middleware Infrastructure, with a domain that is up and running) and wish to extend the same domain to include Oracle Data Integrator, see [Installing Multiple Products in the Same Domain](#).

2

Preparing to Install and Configure Oracle Data Integrator

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

Roadmap for Installing and Configuring a Standard Installation Topology

This roadmap provides all the steps required to install and configure a standard Oracle Data Integrator installation topology.

[Table 2-1](#) shows the high-level steps required to install and configure the topology.

Table 2-1 Oracle Data Integrator Installation Roadmap

Task	Description	Documentation
Verify your system environment	Before beginning the installation, verify that the minimum system and network requirements are met.	NA
Check for any mandatory patches that will be required before or after the installation	Review the Oracle Fusion Middleware Infrastructure release notes to see if there are any mandatory patches required for the software products you are installing.	See Install and Configure in <i>Release Notes for Oracle Fusion Middleware Infrastructure</i> .
Obtain the appropriate distribution	Obtain the Oracle Data Integrator product distribution for installation.	See About Product Distributions .
Determine your installation directories	Verify that the installer can access or create the required installer directories. Also, verify that the directories exist on systems that meet the minimum requirements.	See <i>What Are the Key Oracle Fusion Middleware Directories?</i> in <i>Understanding Oracle Fusion Middleware</i> .

Table 2-1 (Cont.) Oracle Data Integrator Installation Roadmap

Task	Description	Documentation
Install the software	Run the Oracle Universal Installer to install Oracle Data Integrator. Installing the software transfers the software to your system and creates the Oracle home directory.	See Installing Oracle Data Integrator .
Create the Master and Work Repository schemas.	Use the Repository Creation Utility (RCU) to create the Master and Work Repository database schemas.	See Creating the Master and Work Repository Schemas .
Create the physical domain	Use ODI Studio to create the physical agent.	See Configuring Oracle Data Integrator Studio to create the physical agent.
Create and configure your domain	Use the configuration wizard to create and configure your domain.	See Configuring the Domain for a Standalone Agent to create the topology for a standalone agent. See Configuring the Domain for a Standalone Collocated Agent to create the topology for a standalone collocated agent. See Configuring the Domain for a Java EE Agent to create the topology for a Java EE agent.
Administer and prepare your domain for high availability	Discover additional tools and resources to administer your domain. If you have configured a Java EE agent, you can also prepare and configure your domain to be highly available.	See Preparing Your Environment for High Availability .

Roadmap for Verifying Your System Environment

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

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

Table 2-2 Roadmap for Verifying Your System Environment

Task	Description	Documentation
Verify certification and system requirements.	Verify that your operating system is certified and configured for installation and configuration.	See Verifying Certification, System, and Interoperability Requirements .
Identify a proper installation user.	Verify that the installation user has the required permissions to install and configure the software.	See Selecting an Installation User .

Table 2-2 (Cont.) Roadmap for Verifying Your System Environment

Task	Description	Documentation
Select the installation and configuration directories on your system.	Verify that you can create the necessary directories to install and configure the software, according to the recommended directory structure. Select a new, empty Oracle Home directory.	See About the Directories for Installation and Configuration .
Install a certified JDK.	The installation program for the distribution requires a certified JDK present on your system.	See About JDK Requirements for an Oracle Fusion Middleware Installation .
Install and configure a database for mid-tier schemas.	To configure your WebLogic domain, you must have access to a certified database that is configured for the schemas required by Oracle Data Integrator.	See About Database Requirements for an Oracle Fusion Middleware Installation .

Verifying Certification, System, and Interoperability Requirements

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

1. Verifying that your environment meets certification requirements:

Ensure that you install your product on a supported hardware and software configuration.

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

2. Using the system requirements document to verify certification:

Oracle recommends that you use the *Oracle Fusion Middleware System Requirements and Specifications* document to verify that the certification requirements are met. System requirements can change in the future. Therefore, the system requirement documents are kept outside of the documentation libraries and are available on Oracle Technology Network.

3. Verifying interoperability among multiple products:

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

Selecting an Installation User

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

About User Permissions

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

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

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

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

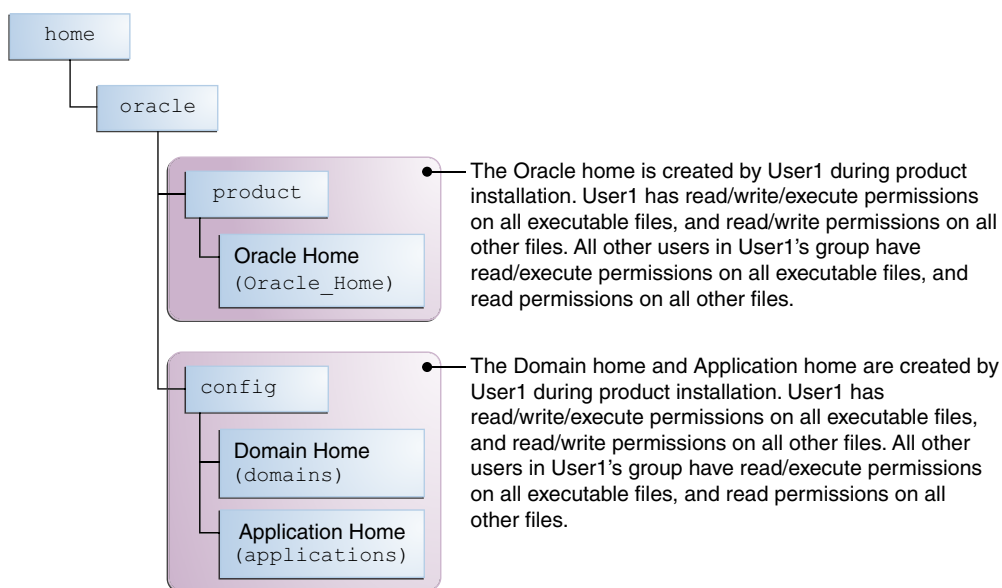
Consider the following examples:

- **Example 1: A Single User Installs the Software and Configures the Domain**

This example explains the file permissions where the same user installs the software and configures the domain.

To ensure proper permissions and privileges for all files, Oracle recommends that the same owner perform both tasks: install the Oracle Fusion Middleware product and configure the WebLogic Server domain by using the Configuration Wizard.

Figure 2-1 Directory Structure when a Single User Installs the Software and Configures the Domain

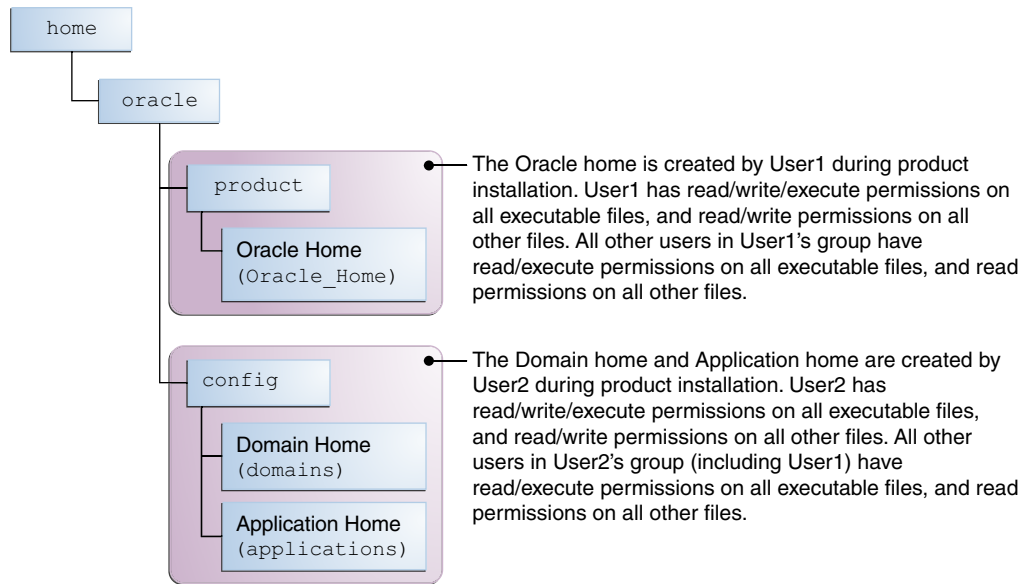


If the user who creates the domain is different than the user who installed the software, then both users must have the same privileges, as shown in the next example.

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

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

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



 **Note:**

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

Consider the following points before you run the installer:

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

```
umask 027
```

You must enter this command in the same terminal window from which you plan to run the product installer.

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

About Non-Default User Permissions on Linux or UNIX Operating Systems

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

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

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

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

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

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

To perform a task with elevated privileges:

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

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

Note:

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

3. Perform the desired task.

For example, to start the product installer:

For a jar file, enter:

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

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

```
distribution_name.exe
```

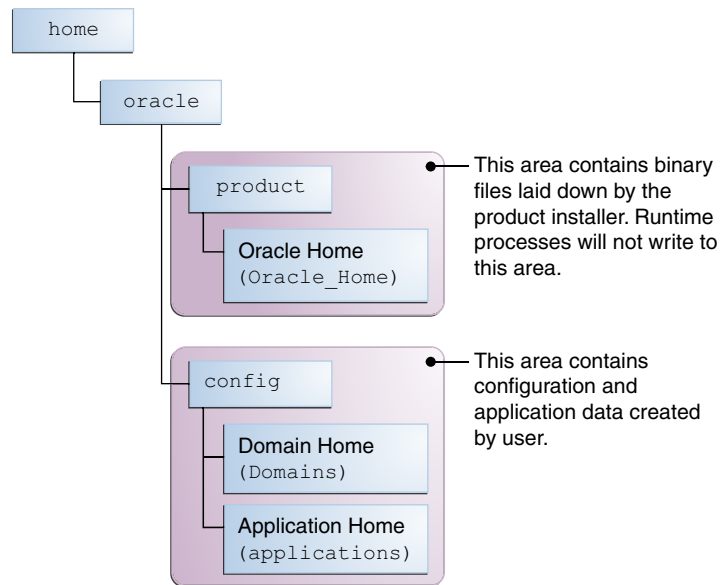
About the Directories for Installation and Configuration

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

About the Recommended Directory Structure

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

Oracle recommends a directory structure similar to the one shown in [Figure 2-3](#).

Figure 2-3 Recommended Oracle Fusion Middleware Directory Structure

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

Oracle recommends that you do not keep your configuration data in the Oracle home directory; if you upgrade your product to another major release, you are required to create an Oracle home for binary files. You must also make sure that your configuration data exists in a location where the binary files in the Oracle home have access.

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

About the Oracle Home Directory

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

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

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

Oracle Home Considerations

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

- Do not include spaces in the name of your Oracle home directory; the installer displays an error message if your Oracle home directory path contains spaces.

- You can install only one instance of each Oracle Fusion Middleware product in a single Oracle home directory. If you need to maintain separate versions of a product on the same machine, each version must be in its own Oracle home directory.

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

Multiple Home Directories

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

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



Note:

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

About the Domain Home Directory

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

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

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

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

About the Application Home Directory

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

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

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

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

Installing Multiple Products in the Same Domain

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

- **Method 1.**

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

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

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

1. Stop all servers to prevent any updates to the domain while you add the new product.
See *Starting and Stopping Oracle Fusion Middleware* in *Administering Oracle Fusion Middleware*.
2. Follow the instructions in the installation guide for Product B, including creating the necessary schemas.
3. Run the Configuration Wizard to configure the domain.
During configuration, the Configuration Wizard automatically detects the components that have been installed and offers you the option to extend the existing Product A domain to include Product B.

- **Method 2.**

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

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

 **See Also:**

- To update WebLogic domains, see Updating WebLogic Domains in *Creating WebLogic Domains Using the Configuration Wizard*.
- For important information regarding the ability of Oracle Fusion Middleware products to function with previous versions of other Oracle Fusion Middleware, Oracle, or third-party products, see Oracle Fusion Middleware 14c (14.1.2.0.0) Interoperability and Compatibility in *Understanding Interoperability and Compatibility*.

Preparing for Shared Storage

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

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

About JDK Requirements for an Oracle Fusion Middleware Installation

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

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

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

Always verify the required JDK version by reviewing the certification information on the *Oracle Fusion Middleware Supported System Configurations* page for Oracle Fusion Middleware.

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

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

About Database Requirements for an Oracle Fusion Middleware Installation

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

To find a certified database for your operating system, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page on *Technical Resources from Oracle*.

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

For Oracle Identity Management only:

- For OIG you need to run a few prerequisite `sql` files prior to creating Oracle Identity Management schemas. See Oracle Database requirements when using Oracle Identity Management in the *System Requirements and Specifications* document.
- Based on your deployment topology and the work load, it is recommended that you refer to the following note on My Oracle Support, and take appropriate actions for your deployment. See [Performance Tuning Guidelines and Diagnostics Collection for Oracle Identity Manager \(OIM\) \(Doc ID 1539554.1\)](#)

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

About Product Distributions

You create the initial Oracle Data Integrator domain using the Oracle Fusion Middleware Infrastructure distribution, which contains both Oracle WebLogic Server software and Oracle Java Required Files (JRF) software.

Oracle JRF software consists of:

- Oracle Web Services Manager
- Oracle Application Development Framework (Oracle ADF)
- Oracle Enterprise Manager Fusion Middleware Control
- Repository Creation Utility (RCU)
- Other libraries and technologies required to support Oracle Fusion Middleware products

Prerequisites:

- Install Oracle Fusion Middleware Infrastructure. For more information about installing Oracle Fusion Middleware Infrastructure, see *Installing the Infrastructure Software in the Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Note:

If you want to access public internet cloud data sources, you must have a direct network connection as connections via proxy servers are not supported.

Obtaining the Product Distribution

You can obtain the Oracle Fusion Middleware Infrastructure and Oracle Data Integrator distribution on *Technical Resources from Oracle*.

To prepare to install Oracle Fusion Middleware Infrastructure and Oracle Data Integrator:

1. Enter `java -version` on the command line to verify that a certified JDK is installed on your system. For 14c (14.1.2.0.0), the certified JDK is 17.0.12 and later.

See [About JDK Requirements for an Oracle Fusion Middleware Installation](#).

2. Locate and download the Oracle Fusion Middleware Infrastructure and Oracle Data Integrator software.

See Obtaining Product Distributions in *Planning an Installation of Oracle Fusion Middleware*.

To obtain the distribution for product evaluation, visit the [Oracle Software Delivery Cloud](#) page.

After preparing to install and configure the software, see [Installing the Oracle Data Integrator Software](#).

Verifying Digital Signature and Integrity of Installation Archive Files

Oracle digitally signs the installation archive files with Oracle certificates to ensure the integrity of the packages before you deploy them in your environments.

Use the Java utility `jarsigner` to verify the integrity of your installation archive files. You can verify the integrity of the installation archive files before you extract the installation files.

Quick Verification

To quickly verify the installation archive files, use the `jarsigner` command with the `-verify` option:

1. Go to the directory where you have downloaded the installation archive files.
2. Run this command to check your installation archive file:

```
jarsigner -verify installation_archive_file
```

For example, to check the Oracle Fusion Middleware Infrastructure archive:

```
jarsigner -verify fmw_14.1.2.0.0_infrastructure.jar
```

```
jar verified.
```

Detailed Certificate Information

If you want detailed certificate information, then use the `-verbose:summary` and `-certs` along with the `-verify` option.

1. Go to the directory where you have downloaded the installation archive files.
2. Run this command to check your installation archive file:

```
jarsigner -verify -verbose:summary -certs installation_archive_file
```

For example, to check the Oracle Fusion Middleware Infrastructure image:

```
jarsigner -verify -verbose:summary -certs fmw_14.1.2.0.0_infrastructure.jar
```

The output is similar to the following:

```
2237119 Fri Dec 6 07:02:30 UTC 2023 META-INF/MANIFEST.MF

>>> Signer
X.509, CN="Oracle America, Inc.", O="Oracle America, Inc.",
L=Redwood City, ST=California, C=US
[
Signature algorithm: SHA256withRSA, 3072-bit key
[certificate is valid from 12/19/24 12:00 AM to 12/19/25 11:59 PM]
X.509, CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1,
O="DigiCert, Inc.", C=US
[
Signature algorithm: SHA384withRSA, 4096-bit key
[certificate is valid from 4/29/24 12:00 AM to 4/28/36 11:59 PM]
X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
[
Signature algorithm: SHA384withRSA, 4096-bit key
[trusted certificate]
>>> TSA
X.509, CN=DigiCert Timestamp 2024 - 2, O=DigiCert, C=US
[
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 9/21/24 12:00 AM to 11/21/33 11:59 PM]
X.509, CN=DigiCert Trusted G4 RSA4096 SHA256 TimeStamping CA,
O="DigiCert, Inc.", C=US
[
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 3/23/24 12:00 AM to 3/22/37 11:59 PM]
X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
[
Signature algorithm: SHA384withRSA, 4096-bit key
[certificate is valid from 8/1/24 12:00 AM to 11/9/31 11:59 PM]

2237281 Fri Feb 17 07:02:32 UTC 2024 META-INF/ORACLE_C.SF (and 1
more)

(Signature related entries)

0 Fri Feb 17 05:41:24 UTC 2023 OPatch/ (and 1897 more)

(Directory entries)

2977 Tue Dec 20 08:02:16 UTC 2024 OPatch/README.txt (and 20199 more)

[entry was signed on 2/17/24 7:02 AM]
>>> Signer
X.509, CN="Oracle America, Inc.", O="Oracle America, Inc.",
L=Redwood City, ST=California, C=US
[
Signature algorithm: SHA256withRSA, 3072-bit key
[certificate is valid from 8/19/24 12:00 AM to 8/19/25 11:59 PM]
X.509, CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1,
O="DigiCert, Inc.", C=US
[
Signature algorithm: SHA384withRSA, 4096-bit key
```

```

[certificate is valid from 4/29/24 12:00 AM to 4/28/36 11:59 PM]
X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
[
Signature algorithm: SHA384withRSA, 4096-bit key
[trusted certificate]
>>> TSA
X.509, CN=DigiCert Timestamp 2024 - 2, O=DigiCert, C=US
[
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 9/21/24 12:00 AM to 11/21/33 11:59 PM]
X.509, CN=DigiCert Trusted G4 RSA4096 SHA256 TimeStamping CA,
O="DigiCert, Inc.", C=US
[
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 3/23/24 12:00 AM to 3/22/37 11:59 PM]
X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
[
Signature algorithm: SHA384withRSA, 4096-bit key
[certificate is valid from 8/1/24 12:00 AM to 11/9/31 11:59 PM]

s = signature was verified
m = entry is listed in manifest
k = at least one certificate was found in keystore
i = at least one certificate was found in identity scope

- Signed by "CN="Oracle America, Inc.", O="Oracle America, Inc.",
L=Redwood City, ST=California, C=US"
  Digest algorithm: SHA-256
  Signature algorithm: SHA256withRSA, 3072-bit key
  Timestamped by "CN=DigiCert Timestamp 2024 - 2, O=DigiCert, C=US" on Fri
Feb 17 07:02:33 UTC 2024
  Timestamp digest algorithm: SHA-256
  Timestamp signature algorithm: SHA256withRSA, 4096-bit key

jar verified.

The signer certificate will expire on 2025-12-19.
The timestamp will expire on 2031-11-09.

```

3

Installing the Oracle Data Integrator Software

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

Starting the Installation Program

Before running the installation program, you must verify the JDK and prerequisite software is installed.

To start the installation program:

1. Sign in to the host system.
2. Change to the directory where you downloaded the installation program.
3. Copy the installer `fmw_14.1.2.0.0_odi_generic.jar` file to the same directory.
4. You must have installed the Oracle Fusion Middleware Infrastructure 14c (14.1.2.0.0). For instructions, see *Installing the Infrastructure Software* in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
5. Start the installation program by running the `java` executable from the JDK directory.



Note:

You can also start the installer in silent mode using a saved response file instead of launching the installer screens. For more about silent or command line installation, see *Using the Oracle Universal Installer in Silent Mode* in *Installing Software with the Oracle Universal Installer*.

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

Navigating the Installation Screens

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

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

Table 3-1 Oracle Data Integrator Install Screens

Screen	Description
Installation Inventory Setup	On Linux or UNIX operating systems, this screen opens if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location. See About the Oracle Central Inventory in <i>Installing Software with the Oracle Universal Installer</i> . This screen does not appear on Windows operating systems.
Welcome	Review the information to make sure that you have met all the prerequisites, then click Next .
Auto Updates	Select to skip automatic updates, select patches, or search for the latest software updates, including important security updates, through your My Oracle Support account.
Installation Location	Specify your Oracle home directory location. <ul style="list-style-type: none"> The Java EE and standalone collocated agent must be installed into an existing Oracle Home directory containing Oracle Fusion Middleware Infrastructure. The standalone agent must be installed into a new, empty Oracle Home directory. If OID has to co-exist with other Oracle upper stack, like SOA; then ODI Enterprise needs to be installed to the existing FMW Oracle Home. You can click View to verify and ensure that you are installing Oracle Data Integrator in the correct Oracle home.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. To view the list of tasks that gets verified, select View Successful Tasks . To view log details, select View Log . If any prerequisite check fails, then an error message appears at the bottom of the screen. Fix the error and click Rerun to try again. To ignore the error or the warning message and continue with the installation, click Skip (not recommended).
Installation Progress	This screen shows the installation progress. When the progress bar reaches 100% complete, click Finish to dismiss the installer, or click Next to see a summary.
Installation Complete	This screen displays the Installation Location and the Feature Sets that are installed. Review this information and click Finish to close the installer.

Verifying the Installation

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

Reviewing the Installation Log Files

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

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

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

Checking the Directory Structure

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

See *What Are the Key Oracle Fusion Middleware Directories?* in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of the Oracle Home

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

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

Installing Oracle Data Integrator in Silent Mode

You can install Oracle Data Integrator in silent mode by using the `silent` flag.

To install Oracle Data Integrator in silent mode, use the `-silent` flag on the command line when you start the installer.

For more information about silent or command line installation, see *Using the Oracle Universal Installer in Silent Mode*.

Example Response Files

The following are response file examples when installing Oracle Data Integrator in Silent mode:

For an Enterprise installation:

```
[ENGINE]

#DO NOT CHANGE THIS.
Response File Version=1.0.0.0.0

[GENERIC]

#The oracle home location. This can be an existing Oracle Home or a new Oracle Home
ORACLE_HOME=/scratch/aganeriw/odi1412enterprise

#Set this variable value to the Installation Type selected. e.g. Standalone
Installation, Enterprise Installation.
INSTALL_TYPE=Enterprise Installation

#Type String (URL format) Indicates the OCM Repeater URL which should be of the format
[scheme[Http/Https]]://[repeater host]:[repeater port]
COLLECTOR_SUPPORTHUB_URL=
```

For a Standalone Installation:

```
[ENGINE]

#DO NOT CHANGE THIS.
Response File Version=1.0.0.0.0

[GENERIC]

#The oracle home location. This can be an existing Oracle Home or a new Oracle Home
```

```
ORACLE_HOME=/scratch/aganeriw/odi1412standalone

#Set this variable value to the Installation Type selected. e.g. Standalone
Installation, Enterprise Installation.
INSTALL_TYPE=Standalone Installation

#Provide the My Oracle Support Username. If you wish to ignore Oracle Configuration
Manager configuration provide empty string for user name.
MYORACLESUPPORT_USERNAME=

#Provide the My Oracle Support Password
MYORACLESUPPORT_PASSWORD=<SECURE VALUE>

#Set this to true if you wish to decline the security updates. Setting this to true and
providing empty string for My Oracle Support username will ignore the Oracle
Configuration Manager configuration
DECLINE_SECURITY_UPDATES=true

#Set this to true if My Oracle Support Password is specified
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false

#Provide the Proxy Host
PROXY_HOST=

#Provide the Proxy Port
PROXY_PORT=

#Provide the Proxy Username
PROXY_USER=

#Provide the Proxy Password
PROXY_PWD=<SECURE VALUE>

#Type String (URL format) Indicates the OCM Repeater URL which should be of the format
[scheme[Http/Https]]://[repeater host]:[repeater port]
COLLECTOR_SUPPORTHUB_URL=
```

4

Creating the Master and Work Repository Schemas

Before you can configure an Oracle Data Integrator domain, you must create the Master and Work Repository (`ODI_REPO`) schemas on a certified database for use with this release of Oracle Fusion Middleware.

Note:

As of Oracle Fusion Middleware 14c (14.1.2.0.0), new schemas are created with editions-based redefinition (EBR) views enabled by default. When EBR is enabled, the schema objects can be upgraded online to a future Fusion Middleware release without any downtime. For more information about using editions-based redefinition, see [Using Edition-based Redefinition](#).

Installing and Configuring a Certified Database

Before you create the database schemas, you must install and configure a certified database, and verify that the database is up and running.

See [About Database Requirements for an Oracle Fusion Middleware Installation](#).

Starting the Repository Creation Utility

Start the Repository Creation Utility (RCU) after you verify that a certified JDK is installed on your system.

To start the RCU:

1. Verify that a certified JDK already exists on your system by running `java -version` from the command line. For 14c (14.1.2.0.0), the certified JDK is 17.0.12 and later.
See [About JDK Requirements for an Oracle Fusion Middleware Installation](#).
2. Ensure that the `JAVA_HOME` environment variable is set to the location of the certified JDK.
3. Change to the following directory:
 - (UNIX) `ORACLE_HOME/oracle_common/bin`
 - (Windows) `ORACLE_HOME\oracle_common\bin`
4. Enter the following command:
 - (UNIX) `./rcu`
 - (Windows) `rcu.bat`

Navigating the Repository Creation Utility Screens to Create Schemas

Enter required information in the RCU screens to create the database schemas.

Introducing the RCU

The Welcome screen is the first screen that appears when you start the RCU.

Click **Next**.

Selecting a Method of Schema Creation

Use the Create Repository screen to select a method to create and load component schemas into the database.

On the Create Repository screen:

- If you have the necessary permissions and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have SYSDBA privileges.
- If you do *not* have the necessary permissions or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script that you can give to your database administrator. See About System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.
- If the DBA has already run the SQL script for System Load, select **Perform Product Load**.

Providing Database Connection Details

On the Database Connection Details screen, provide the database connection details for the RCU to connect to your database.

To provide the database connection details:

1. On the Database Connection Details screen, provide the database connection details.

For example:

```
Database Type: Oracle EBR Database
Connection String Format: Connection Parameters or Connection String
Connection String:
examplehost.exampledomain.com:1521:Orcl.exampledomain.com
Host Name: examplehost.exampledomain.com
Port: 1521
Service Name: Orcl.exampledomain.com
Username: sys
Password: *****
Role: SYSDBA
```

2. Click **Next** to proceed, then click **OK** in the dialog window that confirms a successful database connection.

For information about specifying connection credentials when connecting to an Oracle database, see *Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition*.

Specifying a Custom Prefix and Selecting Schemas

Use the Database Connection Details screen to specify a product database schema with a custom prefix.

The custom prefix logically groups these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported. See, *Understanding Custom Prefixes and Planning Your Schema Creation in Creating Schemas with the Repository Creation Utility*.

To specify custom prefix with schema:

1. On the **Select Components** screen, specify a custom prefix and select the product database schema.
2. Select **Create new prefix**, specify a custom prefix, then select the **Oracle Data Integrator**. The dependent schemas are automatically selected.

Make a note of the custom prefix you choose to enter here; you will need this later on during the domain creation process.

3. Click **Next** to proceed, then click **OK** to confirm that prerequisite checking for schema creation was successful.

Specifying Schema Passwords

On the Schema Passwords screen, specify how you want to set the schema passwords on your database, then enter and confirm your passwords.

You must make a note of the passwords you set on this screen; you will need them later on during the domain creation process.

Click **Next**.

Specifying Custom Variables

On the Custom Variables screen, specify the custom variables for the Master and Work Repository.

For more information about the options on this screen, see *Custom Variables in Creating Schemas with the Repository Creation Utility*.

Completing Schema Creation

Navigate through the remaining RCU screens to complete schema creation.

On the Map Tablespaces screen, the Encrypt Tablespace check box appears *only* if you enabled Transparent Data Encryption (TDE) in the database (Oracle or Oracle EBR) when you start the RCU.

To complete schema creation:

1. On the Map Tablespaces screen, select **Encrypt Tablespace** if you want to encrypt all new tablespaces that the RCU creates.

2. In the Completion Summary screen, click **Close** to dismiss the RCU.

5

Configuring Oracle Data Integrator Studio

Oracle Data Integrator Studio (ODI Studio) is the interface you use to manage and configure Oracle Data Integrator (ODI).

About Oracle Data Integrator Studio

Oracle Data Integrator Studio (ODI Studio) is the developer's interface that you use to configure and manage Oracle Data Integrator (ODI).

ODI Studio is installed as part of standalone or enterprise option under the specified `ORACLE_HOME` directory. ODI Studio provides four graphical Navigators for managing ODI artifacts: Designer, Operator, Topology, and Security. ODI Studio can also directly execute jobs on demand, and can be used for development and initial testing.

Creating the Database Schemas

Before you can configure an Oracle Data Integrator domain, you must create the Master and Work Repository (`ODI_REPO`) schemas on a certified database for use with this release of Oracle Fusion Middleware.

Follow the instructions in [Creating the Master and Work Repository Schemas](#) to create the required schemas.

Setting the Same Time Zone for the Agent and Repository

Each database has its way of identifying the configured time zone. To ensure that time zone differences do not have an impact on schedule, ensure that the time zones for the ODI Studio and the agent are the same as the time zone for the repository before starting them.

Perform the following steps to configure the same time zone:

1. Configure the required time zone information in the `SNP_LOC_REP.CONFIG_TZ_ID` column in Oracle Data Integrator repository.

Example: PST8PDT, Asia/Calcutta, America/New_York, and so on.

For information about valid time zone IDs, see <https://docs.oracle.com/javase/8/docs/api/java/time/ZonedDateTime.html>.

Note:

PST8PDT is the default time zone, which the scheduler will automatically select if you do not configure the time zone.

2. Match the configured time zone in the ODI Studio as shown below:

On the UNIX system:

```
$MW_HOME/odi/studio/bin/odi.conf
AddVMOption -Duser.timezone=<Repository Timezone>
```

- Match the configured time zone in the ODI Agent as shown below:

On the UNIX system:

Start `bash` and then run the following:

```
$DOMAIN_HOME/bin>source setDomainEnv.sh
JAVA_OPTIONS="{JAVA_OPTIONS} -Duser.timezone=<Repository Timezone>"
export JAVA_OPTIONS
```

 **Note:**

You might have more than one domain configured. So, ensure to check `setDomainEnv.sh` and match the configured time zone in all domains.

While retrieving the schedule plan, the scheduler process will convert the schedule plan from the configured time zone to the local zone.

Adding Libraries to ODI Studio

You can add third-party libraries to ODI Studio.

All files required by ODI Studio, such as third-party JDBC drivers, Open Tools, and libraries, should be deployed into the `userlib` directory. The location of this directory is:

- On UNIX systems: `$HOME/.odi/oracledi/userlib`
- On Windows systems: `%APPDATA%\odi\oracledi\userlib`

where `%APPDATA%` is `C:\Documents and Settings\Application Data` (in Windows XP), and `C:\Users\myUserName\AppData\Roaming` (on Windows 7 and Windows 8).

The `userlib` directory contains the `additional_path.txt` file, where you can declare additional files or folders outside of the `userlib` directory from which ODI Studio acquires other libraries and drivers, as illustrated in the following example:

```
; Additional paths file
; You can add here paths to additional libraries
; Examples:
; C:\ java\libs\myjar.jar
; C:\ java\libs\myzip.zip
; C:\java\libs\*.jar will add all jars contained in the C:\java\libs\
directory
; C:\java\libs\**\*.jar will add all jars contained in the
C:\java\libs\ directory and subdirectories

D:\java\libs\*.jar
D:\ODI\JDBC_drivers\sqljdbc_1.2\enu\sqljdbc.jar
```


You must restart ODI Studio after editing the `additional_path.txt` file for the changes to take effect.

Setting Java Home for ODI Studio

Set the `JAVA_HOME` environment variable to the location of the certified JDK before you start ODI Studio.

To set `JAVA_HOME` for ODI Studio:

1. Go to the `ORACLE_HOME/odi/studio/bin` directory.
2. Open the `odi.conf` file in an editor.
3. Add a line at the bottom of the file specifying the location of a certified JDK. . For 14c (14.1.2.0.0), the certified JDK is 17.0.12 and later. For example:
 - (UNIX) `setenv JAVA_HOME/home/Oracle/Java/jdk17.0.12`
 - (Windows) `set JAVA_HOME=C:\home\Oracle\Java\jdk17.0.12`
4. Save the file.

Starting ODI Studio

Start ODI Studio with the `odi` command.

To start ODI Studio:

1. Go to the `ORACLE_HOME/odi/studio` directory.
2. Enter the following command:
 - (UNIX) `./odi.sh`
 - (Windows) `odi.exe`

Note:

On Windows operating systems *only* the user who installed ODI can start ODI Studio. No other user has privileges to start ODI Studio.

Connecting to the Master Repository

Connect to the master repository before creating an agent.

Before you create an agent, you must connect to the master repository.

 **Note:**

If you have installed any previous version of Oracle Data Integrator (ODI) on the same computer you are currently using, you may be asked whether or not you want to import preferences and settings from those previous installations into Oracle Data Integrator Studio (ODI Studio). The procedure assumes that no previous versions of ODI exist on your computer.

For more information, refer to the topics in the ODI Studio **Help** menu.

To connect to the master repository:

1. From the ODI Studio menu, select **File**, then select **New**.

On the New gallery screen, select **Create a New ODI Repository Login**, then click **OK**.

2. On the Oracle Data Integrator Login screen, click the plus sign (+) icon to create a new login. On the Repository Connection Information screen:
 - Oracle Data Integrator Connection section:
 - Login Name: Specify a custom login name.
 - User: Specify `SUPERVISOR` (all capital letters).
 - Password: Specify the supervisor password of RCU Custom Variable screen.
 - Database Connection (Master Repository) section
 - User: Specify the schema user name for the Master repository. This should be `prefix_ODI_REPO` as specified on the Select Components screen in RCU.
 - Password: Specify the schema password for the Master repository. This was specified on the Schema Passwords screen in RCU.
 - Driver List: Select the appropriate driver for your database from the drop-down list.
 - URL: Specify the connection URL. Click on the magnifying glass icon for more information about the connection details for your driver.
 - In the Work Repository section, select **Master Repository Only**.

Click **Test** to test the connection, and fix any errors. After the test is successful, click **OK** to create the connection.

3. Specify and confirm a new wallet password on the New Wallet Password screen.
4. After you have successfully created a new login, you are returned to ODI Studio.

Select **Connect to Repository** and, when prompted, provide your new wallet password.

After providing your wallet password, the Oracle Data Integrator Login screen appears. Provide the following information to log in:

- a. In the drop-down menu in the Login Name field, select the name of the new login you just created.
- b. Specify `SUPERVISOR` as the user name.
- c. Provide the password for the Supervisor user.

When you are logged in, follow the instructions in [Creating an Agent in the Master Repository with ODI Studio](#) to create the agent.

Add Proxy Configuration for ODI Studio

You can add proxy configuration for ODI studio.

Perform the following steps to add proxy configuration for ODI studio:

1. Navigate to **Studio > Tools > Preferences > Web Browser and Proxy > Proxy Settings > Manual Proxy Settings**.
2. Add the proxy host and port number in the respective fields.
3. Click **Save**.

6

Configuring the Domain for a Standalone Agent

Create and configure a standalone domain for a standalone agent in the standard installation topology.

Creating the Database Schemas

Before you can configure an Oracle Data Integrator domain, you must create the Master and Work Repository (ODI_REPO) schemas on a certified database for use with this release of Oracle Fusion Middleware.

Follow the instructions in [Creating the Master and Work Repository Schemas](#) to create the required schemas.

Creating an Agent in the Master Repository with ODI Studio

A physical agent corresponds to a single standalone agent or a Java EE agent and should have a unique name in the topology. You create an ODI agent in the master repository with ODI Studio.

As part of its startup sequence, an agent connects to the master repository to verify that there is a physical agent defined in the master repository with its name. If it finds its entry, the agent continues with startup and, once started, it reads all the scheduled jobs of itself from the repository and starts processing them. If it does not find a physical agent entry for an agent, then the startup fails.

Before you can create an ODI agent, you must first start ODI Studio and connected to the master repository in ODI Studio. See:

- [Starting ODI Studio](#)
- [Connecting to the Master Repository](#)

To create a physical agent in the master repository with ODI Studio:

1. In the navigation pane on the left side, select the **Topology** tab, then right-click **Agents**, then select **New Agent**.
2. Specify the new agent details as summarized in the following table:

Field	Description
Name	Specify <code>OracleDIAgent1</code> as the agent name. This name must match the system component name.
Host	Enter the IP address or name of the machine where the agent is running. Do not use <code>localhost</code> .
Port	Enter the port number on which the agent listens.

 **Note:**

If your environment includes a cluster with Oracle HTTP Server acting as a front-end to the cluster, provide the host name and port of the Oracle HTTP Server host instead. The name should be `OracleDIAgent`, not `OracleDIAgent1`.

3. From the ODI Studio menu, select **File** then **Save All** to save your changes.

Configuring the Domain

Use the Configuration Wizard to create and configure a domain.

For information on other methods to create domains, see Additional Tools for Creating, Extending, and Managing WebLogic Domains in *Creating WebLogic Domains Using the Configuration Wizard*.

Starting the Configuration Wizard

Start the Configuration Wizard to begin configuring a domain.

To start the Configuration Wizard:

1. Change to the following directory:

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

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

where `ORACLE_HOME` is your 14c (14.1.2.0.0) Oracle home.

2. Enter the following command:

(UNIX) `./config.sh`

(Windows) `config.cmd`

Navigating the Configuration Wizard Screens to Create and Configure the Domain

Enter required information in the Configuration Wizard screens to create and configure the domain for the topology.

 **Note:**

You can use this procedure to extend an existing domain. If your needs do not match the instructions in the procedure, be sure to make your selections accordingly, or see the supporting documentation for more details.

Selecting the Configuration Type and Domain Home Location

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

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

To specify the Domain type and Domain home directory:

1. On the Configuration Type screen, select **Create a new domain**.
2. In the Domain Location field, specify your Domain home directory.

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

Selecting the Configuration Template

Use the Templates screen to select the templates you require.

On this screen, select the Oracle Data Integrator - Standalone Agent template.

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

Specifying the Domain Mode and JDK

Use the JDK Selection screen to specify the Java Development Kit (JDK).

On the JDK Selection screen:

- Select the JDK on your system that is certified for use with Oracle Fusion Middleware. For 14c (14.1.2.0.0), the certified JDK is 17.0.12 and later.

Specifying the Database Configuration Type

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

On the Database Configuration type screen, select **RCU Data**. This option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

 **Note:**

If you select **Manual Configuration** on this screen, you must manually fill in parameters for your schema on the next screen.

After selecting **RCU Data**, specify details in the following fields:

Field	Description
Host Name	Enter the name of the server hosting the database. Example: examplehost.exampledomain.com
DBMS/Service	Enter the database DBMS name, or service name if you selected a service type driver. Example: orcl.exampledomain.com
Port	Enter the port number on which the database listens. Example: 1521
Schema Owner Schema Password	Enter the username and password for connecting to the database's Service Table schema. This is the schema username and password entered for the Service Table component on the Schema Passwords screen in the RCU (see Specifying Schema Passwords). The default username is <code>prefix_STB</code> , where <code>prefix</code> is the custom prefix that you defined in the RCU.

For an Autonomous Transaction Processing database (both Autonomous Transaction Processing-Dedicated (ATP-D) and Autonomous Transaction Processing Shared (ATP-S)), specify the connection credentials using only the **Connection URL String** option, and enter the connect string in the following format described in Connection Credentials for an Autonomous Transaction Processing Database.

Click **Get RCU Configuration** when you finish specifying the database connection information. The following output in the Connection Result Log indicates that the operation succeeded:

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

Successfully Done.
```

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

See Database Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Specifying JDBC Component Schema Information

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

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

For an Autonomous Transaction Processing database (both Autonomous Transaction Processing-Dedicated (ATP-D) and Autonomous Transaction Processing Shared (ATP-S)), specify the connection credentials using only the **Connection URL String** option, and enter the connect string in the following format:

```
@TNS_alias?TNS_ADMIN=<path of the wallet files, ojdbc.properties, and
tnsnames.ora>
```

In the connect string, you must pass `TNS_alias` as the database service name found in `tnsnames.ora`, and `TNS_ADMIN` property to the location of the wallet files, `ojdbc.properties`, and `tnsnames.ora`.

Example connect string for Autonomous Transaction Processing-Dedicated (ATP-D) database:

```
@dbname_tp?TNS_ADMIN=/users/test/wallet_dbname/
```

Example connect string for Autonomous Transaction Processing Shared (ATP-S) database:

```
@dbname_tp?TNS_ADMIN=/users/test/wallet_dbname/
```

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

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

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

Testing the JDBC Connections

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

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

By default, the schema password for each schema component is the password you specified while creating your schemas.

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

Creating a New ODI System Component

Use the System Component screen to create a new ODI system component.

On the System Component screen, click **Add** then:

1. Under **System Component**, either accept the default ODI component (`OracleDIAgent1`) name, or specify a name of your choice.
2. Under **Component Type**, ensure that **ODI** is selected from the drop-down list.

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

Configuring the ODI Server Access

Use the ODI Server Configuration screen to configure access to your ODI server.

Set ODI Server configuration values as specified in the following table.

Field	Description
System Component	Select the ODI system component (<code>OracleDIAgent1</code>).
Server Listen Address	Specify the IP address of the server where the ODI server should listen. Do not use <code>localhost</code> .

Field	Description
Server Listen Port	Enter the port number on which ODI server listens.
Supervisor Username	Specify <code>SUPERVISOR</code> as the user name.
Supervisor Password	Specify the Supervisor user password, as specified during schema creation.
Preferred Datasource	Use the default <code>OdiMasterRepository</code> ; do not change this field.

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

Configuring Node Manager

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

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

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

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

Reviewing Your Configuration Specifications and Configuring the Domain

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

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

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

Writing Down Your Domain Home and Administration Server URL

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

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

- Domain Location
- Administration Server URL

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

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

Adding Libraries to a Standalone Agent

After configuring a standalone agent, you can specify additional libraries if required.

To add libraries to a ODI standalone agent:

1. Set the environment variable `ODI_ADDITIONAL_CLASSPATH` to locate additional jars before starting the agent. For example:

```
ODI_ADDITIONAL_CLASSPATH=/share/libs/mytool.jar:/share/libs/drivers/  
mydriver.jar
```

2. Do one of the following:
 - Copy the additional libraries to the `DOMAIN_HOME/lib` directory. The ODI standalone agents will automatically add these libraries to the agent's classpath.
 - Edit the `DOMAIN_HOME/config/fmwconfig/components/ODI/Agent Name/bin/instance.sh/cmd` command to add the libraries to the `ODI_POST_CLASSPATH` variable.

Starting a Standalone Agent Using Node Manager

Start Node Manager, then start the standalone agent.

Note:

If you have not yet configured the physical agent in the master repository, go to [Creating an Agent in the Master Repository with ODI Studio](#).

Starting Node Manager

To start the per-domain Node Manager:

1.
 - (UNIX) Go to the `DOMAIN_HOME/bin` directory.
 - (Windows) Go to the `DOMAIN_HOME\bin` directory.
2. Enter the following command:
 - (UNIX) Using `nohup` and `nm.out` as an example output file:

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

where `LOG_DIR` is the location of directory in which you want to store the log files.
 - (Windows) `startNodeManager.cmd`

Note:

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

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

Starting a Standalone Agent

To start an ODI standalone agent when using Node Manager:

1. Go to the `DOMAIN_HOME/bin` directory.

2. Start the agent:
 - (UNIX) `./startComponent.sh OracleDIAgent1`
 - (Windows) `startComponent.cmd OracleDIAgent1`
3. At the prompt, enter the Administrator password for the domain.
To verify that your agent was started successfully, go to the following URL in your browser:
`http://examplehost.exampledomain.com:port/oraclediagent`

Starting a Standalone Agent Without Node Manager

Start a standalone agent without Node Manager by using the `agent` command.

To start a standalone agent without Node Manager:

1. Go to the `DOMAIN_HOME/bin` directory.
2. Start the agent:
 - (UNIX) `./agent.sh -NAME=OracleDIAgent1`
 - (Windows) `agent.cmd -NAME=OracleDIAgent1`where `-NAME` must be entered in all capital letters, and the name of the agent (in this example, `OracleDIAgent1`) is case-sensitive.

Verifying the Configuration

After completing all configuration steps, you can perform additional steps to verify that your domain is properly configured.

To verify that the domain is configured properly, see [Performing Additional Domain Configuration Tasks](#).



Note:

If your work repository name is other than the default name `WORKREP`, you should edit the value of the parameter `ODI_SECU_WORK_REP` to include the actual work repository name, in the `instance.properties` file. This additional configuration is required for both standalone agent and standalone collocated agent.

Add Proxy Configuration for Standalone Agent

You can add proxy configuration for a standalone agent.

Perform the following steps to add proxy configuration for standalone agent:

1. Open the instance file.
 - **Linux:** `instance.sh`
 - **Windows:** `instance.cmd`

The instance file is located under `<DOMAIN>/config/fmwconfig/components/ODI/OracleDIAgent1/bin/`

2. Edit the file and add or modify the ODI_INSTANCE_JAVA_OPTIONS parameter.

For example:

```
ODI_INSTANCE_JAVA_OPTIONS="$ODI_ADDITIONAL_JAVA_OPTIONS $ODI_SSL_PROPERTIES -Doracle.odi.standalone.agent.useauthenticator=false -Dhttps.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port> - https.proxyHost=<proxy host> -Dhttps.proxyPort=<proxy port> -Dhttp.nonProxyHosts=<non proxy host>"
```

3. Restart the agent.

7

Configuring the Domain for a Standalone Collocated Agent

Create and configure a WebLogic domain for a standalone collocated agent in the standard installation topology.

Creating the Database Schemas

Before you can configure an Oracle Data Integrator domain, you must create the Master and Work Repository (ODI_REPO) schemas on a certified database for use with this release of Oracle Fusion Middleware.

Follow the instructions in [Creating the Master and Work Repository Schemas](#) to create the required schemas.

Creating an Agent in the Master Repository with ODI Studio

A physical agent corresponds to a single standalone agent or a Java EE agent and should have a unique name in the topology. You create an ODI agent in the master repository with ODI Studio.

As part of its startup sequence, an agent connects to the master repository to verify that there is a physical agent defined in the master repository with its name. If it finds its entry, the agent continues with startup and, once started, it reads all the scheduled jobs of itself from the repository and starts processing them. If it does not find a physical agent entry for an agent, then the startup fails.

Before you can create an ODI agent, you must first start ODI Studio and connected to the master repository in ODI Studio. See:

- [Starting ODI Studio](#)
- [Connecting to the Master Repository](#)

To create a physical agent in the master repository with ODI Studio:

1. In the navigation pane on the left side, select the **Topology** tab, then right-click **Agents**, then select **New Agent**.
2. Specify the new agent details as summarized in the following table:

Field	Description
Name	Specify <code>OracleDIAgent1</code> as the agent name. This name must match the system component name.
Host	Enter the IP address or name of the machine where the agent is running. Do not use <code>localhost</code> .
Port	Enter the port number on which the agent listens.

 **Note:**

If your environment includes a cluster with Oracle HTTP Server acting as a front-end to the cluster, provide the host name and port of the Oracle HTTP Server host instead. The name should be `OracleDIAgent`, not `OracleDIAgent1`.

3. From the ODI Studio menu, select **File** then **Save All** to save your changes.

Configuring the Domain

Use the Configuration Wizard to create and configure a domain.

For information on other methods to create domains, see Additional Tools for Creating, Extending, and Managing WebLogic Domains in *Creating WebLogic Domains Using the Configuration Wizard*.

Starting the Configuration Wizard

Start the Configuration Wizard to begin configuring a domain.

To start the Configuration Wizard:

1. Change to the following directory:

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

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

where `ORACLE_HOME` is your 14c (14.1.2.0.0) Oracle home.

2. Enter the following command:

(UNIX) `./config.sh`

(Windows) `config.cmd`

Navigating the Configuration Wizard Screens to Create and Configure the Domain

Enter required information in the Configuration Wizard screens to create and configure the domain for the topology.

 **Note:**

You can use this procedure to extend an existing domain. If your needs do not match the instructions in the procedure, be sure to make your selections accordingly, or see the supporting documentation for more details.

Selecting the Configuration Type and Domain Home Location

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

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

To specify the Domain type and Domain home directory:

1. On the Configuration Type screen, select **Create a new domain**.
2. In the Domain Location field, specify your Domain home directory.

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

Selecting the Configuration Template

Use the Templates screen to select the templates you require.

On the Templates screen, select the following template:

- Oracle Data Integrator - Standalone Collocated Agent

Selecting this template automatically selects the following as dependencies:

- Oracle JRF
- Oracle Coherence Cluster Extension

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

Configuring the Administrator Account

Use the Administrator Account screen to specify the username and password for the default WebLogic Administrator account for the domain.

Oracle recommends that you make a note of the username and password that you enter on this screen; you need these credentials later to boot and connect to the domain's Administration Server.

Specifying the Domain Mode and JDK

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

On the Domain Mode and JDK screen:

- Select **Production** in the **Domain Mode** field.

 **Note:**

As of WebLogic Server 14.1.2.0.0, when you select **Production** mode, WebLogic Server automatically sets some of the security configurations of **Secured Production** to more secure values. However, there are certain security configurations (such as SSL/TLS) that require manual configuration. See Using Secured Production Mode in *Administering Security for Oracle WebLogic Server*.

If you want to disable the more secure default settings, then you may select **Disable Secure Mode**. This will enable the non-SSL listen ports.

If you want to retain the more secure default settings of **Secured Production** mode in general, but want to change which ports (listen ports, SSL listen ports, or administration ports) will be enabled by default in your domain, then you may:

- Leave **Disable Secure Mode** unselected, and
- Change the default port selections under **Enable or Disable Default Ports for Your Domain**

For more information, see Understand How Domain Mode Affects the Default Security Configuration in *Securing a Production Environment for Oracle WebLogic Server*.

- Select the **Oracle HotSpot JDK** in the **JDK** field.

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

Specifying the Database Configuration Type

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

On the Database Configuration type screen, select **RCU Data**. This option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

 **Note:**

If you select **Manual Configuration** on this screen, you must manually fill in parameters for your schema on the next screen.

After selecting **RCU Data**, specify details in the following fields:

Field	Description
Host Name	Enter the name of the server hosting the database. Example: examplehost.exampledomain.com
DBMS/Service	Enter the database DBMS name, or service name if you selected a service type driver. Example: orcl.exampledomain.com

Field	Description
Port	Enter the port number on which the database listens. Example: 1521
Schema Owner Schema Password	Enter the username and password for connecting to the database's Service Table schema. This is the schema username and password entered for the Service Table component on the Schema Passwords screen in the RCU (see Specifying Schema Passwords). The default username is <code>prefix_STB</code> , where <code>prefix</code> is the custom prefix that you defined in the RCU.

For an Autonomous Transaction Processing database (both Autonomous Transaction Processing-Dedicated (ATP-D) and Autonomous Transaction Processing Shared (ATP-S)), specify the connection credentials using only the **Connection URL String** option, and enter the connect string in the following format described in Connection Credentials for an Autonomous Transaction Processing Database.

Click **Get RCU Configuration** when you finish specifying the database connection information. The following output in the Connection Result Log indicates that the operation succeeded:

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

Successfully Done.
```

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

See Database Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Specifying JDBC Component Schema Information

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

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

For an Autonomous Transaction Processing database (both Autonomous Transaction Processing-Dedicated (ATP-D) and Autonomous Transaction Processing Shared (ATP-S)), specify the connection credentials using only the **Connection URL String** option, and enter the connect string in the following format:

```
@TNS_alias?TNS_ADMIN=<path of the wallet files, ojdbc.properties, and
tnsnames.ora>
```

In the connect string, you must pass `TNS_alias` as the database service name found in `tnsnames.ora`, and `TNS_ADMIN` property to the location of the wallet files, `ojdbc.properties`, and `tnsnames.ora`.

Example connect string for Autonomous Transaction Processing-Dedicated (ATP-D) database:

```
@dbname_tp?TNS_ADMIN=/users/test/wallet_dbname/
```

Example connect string for Autonomous Transaction Processing Shared (ATP-S) database:

```
@dbname_tp?TNS_ADMIN=/users/test/wallet_dbname/
```

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

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

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

Testing the JDBC Connections

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

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

By default, the schema password for each schema component is the password you specified while creating your schemas.

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

Selecting Advanced Configuration

Use the Advanced Configuration screen to complete the domain configuration.

On the Advanced Configuration screen, select:

- Administration Server — Required to properly configure the listen address of the Administration Server.
- Node Manager — Required to configure Node Manager.
- System Components — Required to configure the standalone agent.

Optionally, select other available options as required for your desired installation environment. The steps in this guide describe a standard installation topology, but you may choose to follow a different path. If your installation requirements extend to additional options outside the scope of this guide, you may be presented with additional screens to configure those options. For information about all Configuration Wizard screens, see Configuration Wizard Screens in *Creating WebLogic Domains Using the Configuration Wizard*.

Configuring the Administration Server Listen Address

Use the Administration Server screen to select the Listen Address and configure the Administration Server ports.

Note:

The default port values will vary depending on how you configured your domain. The Enable SSL Listen Port is enabled by default, but the default values may change. For a list of default values, see Port Numbers by Product and Component.

1. Provide a name for the Administration Server. The name field must not be null or empty and cannot contain any special characters.
2. Select the drop-down list next to **Listen Address** and select the IP address of the host where the Administration Server will reside or use the system name or DNS name that maps to a single IP address. Do *not* use `All Local Addresses`.
3. Verify the port settings. When the domain type is set to Production, then the **Enable SSL Listen Port** option is enabled by default. Do *not* specify any server groups for the Administration Server.

 **Note:**

You can change the port values as needed, but **they must be unique**. If the same port numbers are used for different ports, you will not be able to navigate to the next step in the Configuration Wizard.

For more information, see *Specifying the Listen Address in Creating WebLogic Domains Using the Configuration Wizard*.

Configuring Node Manager

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

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

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

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

Creating a New ODI System Component

Use the System Component screen to create a new ODI system component.

On the System Component screen, click **Add** then:

1. Under **System Component**, either accept the default ODI component (`OracleDIAgent1`) name, or specify a name of your choice.
2. Under **Component Type**, ensure that **ODI** is selected from the drop-down list.

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

Configuring the ODI Server Access

Use the ODI Server Configuration screen to configure access to your ODI server.

Set ODI Server configuration values as specified in the following table.

Field	Description
System Component	Select the ODI system component (<code>OracleDIAgent1</code>).

Field	Description
Server Listen Address	Specify the IP address of the server where the ODI server should listen. Do not use <code>localhost</code> .
Server Listen Port	Enter the port number on which ODI server listens.
Supervisor Username	Specify <code>SUPERVISOR</code> as the user name.
Supervisor Password	Specify the Supervisor user password, as specified during schema creation.
Preferred Datasource	Use the default <code>OdiMasterRepository</code> ; do not change this field.

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

Specifying the Node Manager Host Address

Use the **Machines** screen to select the IP address of the host running Node Manager.

The system component you just created is automatically assigned to a machine called `LocalODIMachine`. The **Machines** screen shows that this machine is already created.

In the **Node Manager Listen Address** field, select the IP address of the host running Node Manager. Do not use `localhost`.

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

Assigning System Components to Machines

Use the **Assign System Components to Machines** screen to assign the system component to the ODI machine.

On the **Assign System Components to Machines** screen, verify that the system component `OracleDIAgent1` is assigned to the `LocalODIMachine`.

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

Reviewing Your Configuration Specifications and Configuring the Domain

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

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

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

Writing Down Your Domain Home and Administration Server URL

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

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

- Domain Location

- Administration Server URL

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

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

Adding Libraries to a Standalone Collocated Agent

After configuring a standalone collocated agent, you can specify additional libraries, if required.

To add libraries to a ODI standalone collocated agent:

1. Set the environment variable `ODI_ADDITIONAL_CLASSPATH` to locate additional jars before starting the agent. For example:

```
ODI_ADDITIONAL_CLASSPATH=/share/libs/mytool.jar:/share/libs/drivers/  
mydriver.jar
```

2. Do one of the following:
 - Copy the additional libraries to the `DOMAIN_HOME/lib` directory. The ODI standalone collocated agents will automatically add these libraries to the agent's classpath.
 - Edit the `DOMAIN_HOME/config/fmwconfig/components/ODI/Agent Name/bin/instance.sh/cmd` command to add the libraries to the `ODI_POST_CLASSPATH` variable.

Starting a Standalone Collocated Agent Using Node Manager

Start Node Manager, then start the standalone collated agent.

Note:

If you have not yet configured the physical agent in the master repository, go to [Creating an Agent in the Master Repository with ODI Studio](#).

Starting Node Manager

To start the per-domain Node Manager:

1.
 - (UNIX) Go to the `DOMAIN_HOME/bin` directory.
 - (Windows) Go to the `DOMAIN_HOME\bin` directory.
2. Enter the following command:
 - (UNIX) Using `nohup` and `nm.out` as an example output file:

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

where `LOG_DIR` is the location of directory in which you want to store the log files.
 - (Windows) `startNodeManager.cmd`

 **Note:**

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

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

Starting a Standalone Collocated Agent

To start an ODI standalone collated agent when using Node Manager:

1. Go to the `DOMAIN_HOME/bin` directory.
2. Start the agent:
 - (UNIX) `./startComponent.sh OracleDIAgent1`
 - (Windows) `startComponent.cmd OracleDIAgent1`
3. At the prompt, enter the Administrator password for the domain.

To verify that your agent was started successfully, go to the following URL in your browser:

`http://examplehost.exampledomain.com:port/oraclediagent`

Starting a Standalone Collocated Agent Without Node Manager

Start a standalone collated agent without Node Manager by using the `agent` command.

To start a standalone agent without Node Manager:

1. Go to the `DOMAIN_HOME/bin` directory.
2. Start the agent:
 - (UNIX) `./agent.sh -NAME=OracleDIAgent1`
 - (Windows) `agent.cmd -NAME=OracleDIAgent1`

where `-NAME` must be entered in all capital letters, and the name of the agent (in this example, `OracleDIAgent1`) is case-sensitive.

Verifying the Configuration

After completing all configuration steps, you can perform additional steps to verify that your domain is properly configured.

To verify that the domain is configured properly, see [Performing Additional Domain Configuration Tasks](#).

 **Note:**

If your work repository name is other than the default name `WORKREP`, you should edit the value of the parameter `ODI_SECU_WORK_REP` to include the actual work repository name, in the `instance.properties` file. This additional configuration is required for both standalone agent and standalone collocated agent.

8

Configuring the Domain for a Java EE Agent

Create and configure a WebLogic domain for a Java EE agent in the standard installation topology.

Creating the Database Schemas

Before you can configure an Oracle Data Integrator domain, you must create the Master and Work Repository (ODI_REPO) schemas on a certified database for use with this release of Oracle Fusion Middleware.

Follow the instructions in [Creating the Master and Work Repository Schemas](#) to create the required schemas.

Generating a Template for a Java EE Agent

Use the Server Template Generation wizard to generate the non-default template for deploying a Java EE agent defined in the ODI master repository.

Use this procedure if you require changes to the default template that deploys a Java EE agent with default name and settings. To create a Java EE agent with the Server Template Generation wizard:

1. From the **Physical Agent Editor** menu, select **Generate Server Template** to start the Template Generation wizard.
2. In the Agent Information step, review the agent information and modify the default configuration as appropriate. This information includes the following parameters: the **Agent Name**, the **Datasource JNDI Name**, the **Connection Retry Count**, the **Retry Delay**, and the **Supervisor Key**.
 - The **Connection Retry Count** specifies the number of retry attempts if the agent loses the connection to the repository. Note that setting this parameter to a non-zero value enables a high availability connection retry feature if the ODI repository resides on an Oracle RAC database. If this feature is enabled, the agent can continue to execute sessions without interruptions even if one or more Oracle RAC nodes become unavailable.
 - The **Retry Delay** specifies the number of milliseconds between connection retry attempts.
 - The **Supervisor Key** specifies the name of the key in the application server credential store that contains the login and the password of an ODI user with Supervisor privileges. The agent uses these credentials to connect to the repository.
3. Click **Next**.
4. To add domain and shared libraries for the agent, navigate to the **Libraries and Drivers** page, and then check the libraries you want to deploy with your agent. Libraries can be any JAR or ZIP files required by the agent. Select any additional JDBC drivers or libraries for accessing the source and target data servers in this page. Note that library name validation is limited within the newly added libraries, not against already deployed shared libraries.

- Domain libraries are the libraries deployed in a domain that are available to all applications running in the domain. This approach to adding libraries is typically used when adding infrequently changed JAR files. Note that when using this approach, you must reboot all servers in the domain for the change to take effect.
- Shared libraries are libraries you specify in the agent deployment descriptor before deploying the agent to the domain. In this way, multiple applications can use the libraries resources and duplication is avoided. This approach to adding libraries is used for libraries that may need modifications or upgrade. Shared libraries allow you to change libraries without restarting the server, but you may need to restart the application.

 **Note:**

Name field is mandatory for shared libraries.

5. Click **Next**.
6. In the Datasources step, select the correct data source that has to be deployed with this agent.
7. Click **Next**.
8. In **Template Target and Summary** step, enter the **Target Template Path** that specifies the location where the server template is generated.
9. Click **Finish** to close the wizard and generate the server template with your specifications.

The generated template can be used to deploy the agent in an Oracle WebLogic Server domain.

Creating an Agent in the Master Repository with ODI Studio

A physical agent corresponds to a single standalone agent or a Java EE agent and should have a unique name in the topology. You create an ODI agent in the master repository with ODI Studio.

As part of its startup sequence, an agent connects to the master repository to verify that there is a physical agent defined in the master repository with its name. If it finds its entry, the agent continues with startup and, once started, it reads all the scheduled jobs of itself from the repository and starts processing them. If it does not find a physical agent entry for an agent, then the startup fails.

Before you can create an ODI agent, you must first start ODI Studio and connected to the master repository in ODI Studio. See:

- [Starting ODI Studio](#)
- [Connecting to the Master Repository](#)

To create a physical agent in the master repository with ODI Studio:

1. In the navigation pane on the left side, select the **Topology** tab, then right-click **Agents**, then select **New Agent**.
2. Specify the new agent details as summarized in the following table:

Field	Description
Name	Specify <code>OracleDIAgent1</code> as the agent name. This name must match the system component name.
Host	Enter the IP address or name of the machine where the agent is running. Do not use <code>localhost</code> .
Port	Enter the port number on which the agent listens.

 **Note:**

If your environment includes a cluster with Oracle HTTP Server acting as a front-end to the cluster, provide the host name and port of the Oracle HTTP Server host instead. The name should be `OracleDIAgent`, not `OracleDIAgent1`.

- From the ODI Studio menu, select **File** then **Save All** to save your changes.

Configuring the Domain

Use the Configuration Wizard to create and configure a domain.

For information on other methods to create domains, see Additional Tools for Creating, Extending, and Managing WebLogic Domains in *Creating WebLogic Domains Using the Configuration Wizard*.

Starting the Configuration Wizard

Start the Configuration Wizard to begin configuring a domain.

To start the Configuration Wizard:

- Change to the following directory:

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

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

where `ORACLE_HOME` is your 14c (14.1.2.0.0) Oracle home.

- Enter the following command:

(UNIX) `./config.sh`

(Windows) `config.cmd`

Navigating the Configuration Wizard Screens to Create and Configure the Domain

Enter required information in the Configuration Wizard screens to create and configure the domain for the topology.

**Note:**

You can use this procedure to extend an existing domain. If your needs do not match the instructions in the procedure, be sure to make your selections accordingly, or see the supporting documentation for more details.

Selecting the Configuration Type and Domain Home Location

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

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

To specify the Domain type and Domain home directory:

1. On the Configuration Type screen, select **Create a new domain**.
2. In the Domain Location field, specify your Domain home directory.

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

Selecting the Configuration Template

Use the Templates screen to select the templates you require.

On the Templates screen, select the following template:

- Oracle Enterprise Manager Plug-in for ODI
Selecting this template automatically selects the following as dependencies:
 - Oracle Enterprise Manager
 - Oracle JRF
 - WebLogic Coherence Cluster Extension
- Oracle Data Integrator - Agent
Selecting this template automatically selects the following as dependencies:
 - Oracle Data Integrator - Agent Libraries
 - Oracle Data Integrator SDK Shared Library TemplateIn addition, select the following templates:
 - Oracle Data Integrator - JRF Async Web Services
 - Oracle Data Integrator - REST Services
- Oracle Data Integrator - Console

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

Selecting the Application Home Location

Use the Application Location screen to select the location to store applications associated with your domain, also known as the *Application home* directory.

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

For more about the Application home directory, see [About the Application Home Directory](#).

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

Configuring the Administrator Account

Use the Administrator Account screen to specify the username and password for the default WebLogic Administrator account for the domain.

Oracle recommends that you make a note of the username and password that you enter on this screen; you need these credentials later to boot and connect to the domain's Administration Server.

Specifying the Domain Mode and JDK

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

On the Domain Mode and JDK screen:

- Select **Production** in the **Domain Mode** field.

Note:

As of WebLogic Server 14.1.2.0.0, when you select **Production** mode, WebLogic Server automatically sets some of the security configurations of **Secured Production** to more secure values. However, there are certain security configurations (such as SSL/TLS) that require manual configuration. See *Using Secured Production Mode in Administering Security for Oracle WebLogic Server*.

If you want to disable the more secure default settings, then you may select **Disable Secure Mode**. This will enable the non-SSL listen ports.

If you want to retain the more secure default settings of **Secured Production** mode in general, but want to change which ports (listen ports, SSL listen ports, or administration ports) will be enabled by default in your domain, then you may:

- Leave **Disable Secure Mode** unselected, and
- Change the default port selections under **Enable or Disable Default Ports for Your Domain**

For more information, see *Understand How Domain Mode Affects the Default Security Configuration* in *Securing a Production Environment for Oracle WebLogic Server*.

- Select the **Oracle HotSpot JDK** in the **JDK** field.

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

Specifying the Database Configuration Type

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

On the Database Configuration type screen, select **RCU Data**. This option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

Note:

If you select **Manual Configuration** on this screen, you must manually fill in parameters for your schema on the next screen.

After selecting **RCU Data**, specify details in the following fields:

Field	Description
Host Name	Enter the name of the server hosting the database. Example: examplehost.exampledomain.com
DBMS/Service	Enter the database DBMS name, or service name if you selected a service type driver. Example: orcl.exampledomain.com
Port	Enter the port number on which the database listens. Example: 1521
Schema Owner Schema Password	Enter the username and password for connecting to the database's Service Table schema. This is the schema username and password entered for the Service Table component on the Schema Passwords screen in the RCU (see Specifying Schema Passwords). The default username is <code>prefix_STB</code> , where <code>prefix</code> is the custom prefix that you defined in the RCU.

For an Autonomous Transaction Processing database (both Autonomous Transaction Processing-Dedicated (ATP-D) and Autonomous Transaction Processing Shared (ATP-S)), specify the connection credentials using only the **Connection URL String** option, and enter the connect string in the following format described in Connection Credentials for an Autonomous Transaction Processing Database.

Click **Get RCU Configuration** when you finish specifying the database connection information. The following output in the Connection Result Log indicates that the operation succeeded:

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

Successfully Done.

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

See Database Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Specifying JDBC Component Schema Information

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

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

For an Autonomous Transaction Processing database (both Autonomous Transaction Processing-Dedicated (ATP-D) and Autonomous Transaction Processing Shared (ATP-S)), specify the connection credentials using only the **Connection URL String** option, and enter the connect string in the following format:

```
@TNS_alias?TNS_ADMIN=<path of the wallet files, ojdbc.properties, and  
tnsnames.ora>
```

In the connect string, you must pass `TNS_alias` as the database service name found in `tnsnames.ora`, and `TNS_ADMIN` property to the location of the wallet files, `ojdbc.properties`, and `tnsnames.ora`.

Example connect string for Autonomous Transaction Processing-Dedicated (ATP-D) database:

```
@dbname_tp?TNS_ADMIN=/users/test/wallet_dbname/
```

Example connect string for Autonomous Transaction Processing Shared (ATP-S) database:

```
@dbname_tp?TNS_ADMIN=/users/test/wallet_dbname/
```

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

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

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

Testing the JDBC Connections

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

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

By default, the schema password for each schema component is the password you specified while creating your schemas.

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

Creating Credential Keys

Use the Credentials screen to create two keys.

On the Credentials screen:

1. A key for the Supervisor user already exists. Modify the credential as follows:
 - a. Specify SUPERVISOR (all CAPS) as the user name. This is the default name initially assigned to the SUPERVISOR account and cannot be changed unless you create a new SUPERVISOR account.
 - b. The password must be the same password specified on the Custom Variables screen in RCU during schema creation.
2. Create a second credential:
 - a. Click the plus sign (+) icon to add a new credential.
 - b. In the **Key Name** field, enter the name of this domain as the key.
 - c. In the **Username** and **Password** fields, provide the Administrator user's user name and password.
 - d. Select `oracle.odi.credmap` as the store name.

 **Note:**

If the repository is configured in external authentication mode, then the SUPERVISOR user name and password should be entered according to your external authentication settings.

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

Selecting Advanced Configuration

Use the Advanced Configuration screen to complete the domain configuration.

On the Advanced Configuration screen, select:

- Administration Server
Required to properly configure the listen address of the Administration Server.
- Node Manager
Required to configure Node Manager.
- Topology
Required to configure the Oracle Data Integrator Managed Server.

Optionally, select other available options as required for your desired installation environment. The steps in this guide describe a standard installation topology, but you may choose to follow a different path. If your installation requirements extend to additional options outside the scope of this guide, you may be presented with additional screens to configure those options. For information about all Configuration Wizard screens, see Configuration Wizard Screens in *Creating WebLogic Domains Using the Configuration Wizard*.

Configuring the Administration Server Listen Address

Use the Administration Server screen to select the Listen Address and configure the Administration Server ports.

 **Note:**

The default port values will vary depending on how you configured your domain. The Enable SSL Listen Port is enabled by default, but the default values may change. For a list of default values, see Port Numbers by Product and Component.

1. Provide a name for the Administration Server. The name field must not be null or empty and cannot contain any special characters.
2. Select the drop-down list next to **Listen Address** and select the IP address of the host where the Administration Server will reside or use the system name or DNS name that maps to a single IP address. Do *not* use All Local Addresses.
3. Verify the port settings. When the domain type is set to Production, then the **Enable SSL Listen Port** option is enabled by default. Do *not* specify any server groups for the Administration Server.

 **Note:**

You can change the port values as needed, but **they must be unique**. If the same port numbers are used for different ports, you will not be able to navigate to the next step in the Configuration Wizard.

For more information, see Specifying the Listen Address in *Creating WebLogic Domains Using the Configuration Wizard*.

Configuring Node Manager

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

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

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

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

Configuring Managed Servers for Oracle Data Integrator

Use the Managed Servers screen to configure Managed Servers.

On the Managed Servers screen, a Managed Server named `ODI_server1` is automatically created by default.

To configure Managed Servers for Oracle Data Integrator:

1. In the Listen Address drop-down list, select the IP address of the host on which the Managed Server will reside. Do not use `All Local Addresses`.
2. In the Server Groups drop-down list, make sure that **ODI-MGD-SVRS** is selected. This server group ensures that the Oracle JRF services are targeted to the Managed Servers you are creating.

Server groups target Fusion Middleware applications and services to one or more servers by mapping defined application service groups to each defined server group. A given application service group may be mapped to multiple server groups if needed. Any application services that are mapped to a given server group are automatically targeted to all servers that are assigned to that group. See *Application Service Groups, Server Groups, and Application Service Mappings* in *Domain Template Reference*.

These server names are referenced in examples throughout this document; if you choose different names be sure to replace them as needed.

 **Note:**

You must make a note of the IP address and port number for the Managed Server. You will need this information when you configure the ODI agent.

Port numbers must be unique when creating multiple servers.

3. Configuring a second Managed Server is one of the steps needed to configure the standard topology for high availability. If you are not creating a highly available environment, then this step is optional.

Click **Add** and select **ODI-MGD-SVRS** to create a second Managed Server named `ODI_server2`.

For more information about the high availability standard topology, see *Understanding the Fusion Middleware Standard HA Topology* in *High Availability Guide*.

For more information about the next steps to prepare for high availability after your domain is configured, see [Preparing Your Environment for High Availability](#).

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

Configuring a Cluster for Oracle Data Integrator

Use the Clusters screen to create a new cluster.

On the Clusters screen:

1. Click **Add**.
2. Specify `ODI_cluster1` in the Cluster Name field.
3. Leave the Cluster Address field blank.

By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, see *Considerations for Choosing Unicast or Multicast* in *Administering Clusters for Oracle WebLogic Server*.

You can also create clusters using Fusion Middleware Control. In this case, you can configure cluster communication (unicast or multicast) when you create the new cluster. See *Create and configure clusters* in *Oracle WebLogic Server Administration Console Online Help*.

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

Defining Server Templates

If you are creating dynamic clusters for a high availability setup, use the Server Templates screen to define one or more server templates for the domain.

To add Server Templates:

Note:

The default port values will vary depending on how you configured your domain. The Enable SSL Listen Port is enabled by default, but the default values may change. For a list of default values, see Port Numbers by Product and Component.

1. Click **Add** to create `new_ServerTemplate_1`. The server template name will increment automatically when an additional server template is added (`new_ServerTemplate_2`).
2. For Secure Production Mode, verify that the **Enable SSL Port** option is selected. The default SSL Listen Port does not increment automatically when a new server template is added. You can change the default to Enable Listen Port, but Oracle recommends that retain the default to enable SSL. Enabling Listen Port disables SSL Listen Port.

Note:

You can change the port values as needed using an integer in the range of 1 and 65535, but they must be unique. If the same port numbers are used for different ports, you will receive a port conflict error and you will not be able to start the server.

3. The Administration Port does not increment when an additional server template is added.

Note:

If the Listen ports are disabled, then instead of seeing a number you will see Disabled.

For steps to create a dynamic cluster for a high availability setup, see Using Dynamic Clusters in *High Availability Guide*.

Configuring Dynamic Servers

If you are creating dynamic clusters for a high availability setup, use the Dynamic Servers screen to configure the dynamic servers.

If you are *not* configuring a dynamic cluster, click **Next** to continue configuring the domain.

 **Note:**

When you create dynamic clusters, keep in mind that after you assign the **Machine Name Match Expression**, you do not need to create machines for your dynamic cluster.

To create a dynamic cluster for a high availability setup, see Using Dynamic Clusters in *High Availability Guide*.

Assigning Oracle Data Integrator Managed Servers to the Cluster

Use the Assign Servers to Clusters screen to assign Managed Servers to a new *configured cluster*. A configured cluster is a cluster you configure manually. You do not use this screen if you are configuring a *dynamic cluster*, a cluster that contains one or more generated server instances that are based on a server template.

For more on configured cluster and dynamic cluster terms, see About Dynamic Clusters in *Understanding Oracle WebLogic Server*.

On the Assign Servers to Clusters screen:

1. In the Clusters pane, select the cluster to which you want to assign the Managed Servers; in this case, `ODI_cluster1`.
2. In the Servers pane, assign `ODI_server1` to `ODI_cluster1` by doing one of the following:
 - Click once on `ODI_server1` to select it, then click the right arrow to move it beneath the selected cluster (`ODI_cluster1`) in the Clusters pane.
 - Double-click on `ODI_server1` to move it beneath the selected cluster (`ODI_cluster1`) in the Clusters pane.
3. Repeat to assign `ODI_server2` to `ODI_cluster1`.

The following image shows a generic example of the Clusters pane after Managed Servers are assigned to clusters.

For more information about this screen, see Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster.

Leave the default port number as the Coherence cluster listen port. After configuration, the Coherence cluster is automatically added to the domain.

 **Note:**

Setting the unicast listen port to 0 creates an offset for the Managed Server port numbers. The offset is 5000, meaning the maximum allowed value that you can assign to a Managed Server port number is 60535, instead of 65535.

For Coherence licensing information, see Oracle Coherence Products in *Licensing Information*.

Creating a New Oracle Data Integrator Machine

Use the Machines screen to create new machines in the domain. A machine is required so that Node Manager can start and stop servers.

If you plan to create a high availability environment and know the list of machines your target topology requires, you can follow the instructions in this section to create all the machines at this time. For more about scale out steps, see *Optional Scale Out Procedure* in *High Availability Guide*.

To create a new Oracle Data Integrator machine so that Node Manager can start and stop servers:

1. Select the Machine tab (for Windows) or the UNIX Machine tab (for UNIX), then click **Add** to create a new machine.
2. In the Name field, specify a machine name, such as `ODI_machine1`.
3. In the Node Manager Listen Address field, select the IP address of the machine in which the Managed Servers are being configured.

You must select a specific interface and not `localhost`. This allows Coherence cluster addresses to be dynamically calculated.

4. Verify the port in the Node Manager Listen Port field.
5. Repeat these steps to add more machines, if required.



Note:

If you are extending an existing domain, you can assign servers to any existing machine. It is not necessary to create a new machine unless your situation requires it.

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

Assigning Servers to Oracle Data Integrator Machines

Use the Assign Servers to Machines screen to assign the Administration Server and Managed Servers to the new machine you just created.

On the Assign Servers to Machines screen:

1. In the Machines pane, select the machine to which you want to assign the servers; in this case, `ODI_machine1`.
2. In the Servers pane, assign `AdminServer` to `ODI_machine1` by doing one of the following:
 - Click once on `AdminServer` to select it, then click the right arrow to move it beneath the selected machine (`ODI_machine1`) in the Machines pane.
 - Double-click on `AdminServer` to move it beneath the selected machine (`ODI_machine1`) in the Machines pane.
3. Repeat these steps to assign all Managed Servers to their respective machines.

For more information about this screen, see *Assign Servers to Machines in Creating WebLogic Domains Using the Configuration Wizard*.

Reviewing Your Configuration Specifications and Configuring the Domain

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

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

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

Writing Down Your Domain Home and Administration Server URL

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

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

- Domain Location
- Administration Server URL

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

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

Adding Libraries to a Java EE Agent

After configuring a Java EE agent, you can specify additional libraries if required.

To add libraries to a Java EE agent:

1. Set the environment variable `ODI_ADDITIONAL_CLASSPATH` to locate additional jars before starting the agent. For example:

```
ODI_ADDITIONAL_CLASSPATH=/share/libs/mytool.jar:/share/libs/drivers/  
mydriver.jar
```

2. Do one the following:
 - Copy the additional libraries to the `DOMAIN_HOME/lib` directory. The Java EE will automatically add these libraries to the agent's classpath.
 - Follow the steps below to use the `generate_agent_wls_template.sh` script or ODI Studio to generate an agent template and then create/extend a WebLogic domain with that template.

To generate a template for a Java EE agent using the `generate_agent_wls_template.sh` script, specify domain and shared libraries with the `-domainLibraries` and `-sharedLibraries` arguments as illustrated in the following example:

```
./generate_agent_wls_template.sh  
-agentName sample  
-agentContextName context  
-odiUser SUPERVISOR  
-destinationJar /home/userName/NewFormaterJar.jar
```

```
-retryCount 10
-retryDelayMS 1
-createDataSourceForMasterRepository false
-masterDataSource jdbc/
odiMasterRepository,oracle.jdbc.OracleDriver,jdbc:oracle:thin:@mycompany.com-
isESSDependency false
-addToPath /home/userName
-domainLibraries /home/userName/odi/myDomain/userlib/myLib.jar
-sharedLibraries MessageBox
```

To generate a template for a Java EE agent and specify additional libraries for the agent using ODI Studio:

1. Start ODI Studio.
2. Select **Generate Server Template**, to start the Template Generation wizard.
3. In the **Agent Information** page, enter the agent information, such as the Java EE agent name, the datasource JNDI name, the connection retry count, and the Supervisor key.
4. In the **Libraries and Drivers** page, select the libraries you want to deploy with your Java EE agent from the list of external libraries. Note that library name validation is limited within the newly added libraries, not against already deployed shared libraries. You can select a domain library or a shared library. **Domain** libraries are the libraries deployed in the WebLogic domain and are available to a Java EE agent, since they are available to all applications running in the domain. Domain libraries are typically used when adding infrequently changed JAR files. Note that when using this approach, you must reboot all servers in the domain for the change to take effect. **Shared** libraries are libraries you specify in the agent deployment descriptor, before deploying the agent to the domain. With shared libraries, multiple applications can use the libraries resources and duplication is avoided. Shared libraries are used for libraries that may need modifications or upgrade. These libraries allow you to change libraries without restarting the server, but you may need to restart the application
5. Click **Next** to display the **Datasources** page.
6. In that page, select the datasource definitions that you want to include in the agent template. You can only select datasources using this wizard. To add datasources use the **Data Sources** tab.
7. Click **Next** to display the **Template Target and Summary** page.
8. In that page, specify where the template should be generated in the **Target Template Path** box.
9. Click **Finish** to close the wizard and generate the agent template.

Starting the Servers

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

 **Note:**

Depending on your existing security settings, you may need to perform additional configuration before you can manage a domain with secured production mode enabled. For more information, see [Connecting to the Administration Server using WebLogic Remote Console](#)

Windows Users Must Modify the `setDomainEnv.cmd` Before Starting Servers

Before starting the servers, Windows operating system users will need to modify the `setDomainEnv.cmd` file.

1. Add the following parameters:

```
-
Dweblogic.security.SSL.trustedCAKeyStore=C:\myfiles\install\14120\wlserver\
server\lib\trust.p12
-
Djavax.net.ssl.trustStore=C:\myfiles\install\14120\wlserver\server\lib\trus
t.p12
-Djavax.net.ssl.trustStorePassword=trustKeyStorePassword
```

2. Remove `Demotrust.jks` entry from the `EXTRA_JAVA_PROPERTIES` section of `setDomainEnv.cmd`.
3. Save the file.

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

For more information about Domain Mode, see [Understand How Domain Mode Affects the Default Security Configuration](#).

Starting Node Manager

To start the per-domain Node Manager:

1.
 - (UNIX) Go to the `DOMAIN_HOME/bin` directory.
 - (Windows) Go to the `DOMAIN_HOME\bin` directory.
2. Enter the following command:
 - (UNIX) Using `nohup` and `nm.out` as an example output file:

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

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

- (Windows) `startNodeManager.cmd`

 **Note:**


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

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

Starting the Administration Server

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

To start the Administration Server:

 **Note:**

When using secured production mode, you must provide additional parameters to start the Administration Server. See Connecting to the Administration Server using WLST in *Administering Security for Oracle WebLogic Server*.

1. **(Optional)** When using **Production Mode**, you can create a *boot.properties* file before starting the Administration Server and provide necessary permissions. This file can be created to bypass the need to provide a username and password when starting the Administration Server. For more information, see Creating a Boot Identity File for an Administration Server in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.
2. Go to the `DOMAIN_HOME/bin` directory.
3. Enter the following command:

- (UNIX)
`./startWebLogic.sh`
- (Windows)
`startWebLogic.cmd`

If you selected **Production Mode** on the Domain Mode and JDK screen when you created the domain, and you did not create the optional *boot.properties* file, you see a prompt for the Administrator user login credentials as provided on the Administrator Account screen.

4. Open a browser and verify that the Administration Server is up and running. The default port values will vary depending on how you configured your domain. The Enable SSL Listen Port is enabled by default, but the default values may change. For a list of default values, see Port Numbers by Product and Component.

`https://<Host_Name>:<port>`

5. Verify that all servers in the domain have unique port values. From the WebLogic Remote Console, you can review the **Local Administration Port Override** fields for each managed server and verify that each has a unique value. If one or more ports is using the same value, then you must change them before starting the managed servers. For more information about changing port values, see *Connect to an Administration Server* in the Oracle WebLogic Remote Console.

 **Note:**

The WebLogic Server Administration Console has been removed. For comparable functionality, you should use the WebLogic Remote Console. For more information, see Oracle WebLogic Remote Console.

Starting the Managed Servers

To start a WebLogic Managed Server that is not set to secure product mode, you can use the `startManagedWebLogic` script:

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

 **Note:**

When using secured production mode, you must provide additional parameters to start the Managed Servers. See *Starting Managed Servers using a Start Script in Administering Security for Oracle WebLogic Server*.

Add Proxy Configuration for JEE Agent

You can add proxy configuration for JEE agent.

Perform the following steps to add proxy configuration for JEE agent.

1. Open the `setDomainEnv` file.
 - **Linux:** `setDomainEnv.sh`
 - **Windows:** `setDomainEnv.cmd`

The `setDomainEnv` file is located under `<DOMAIN>/bin`

2. Edit the file and add or modify the `JAVA_OPTIONS` parameters.

For example:

```
JAVA_OPTIONS="{JAVA_OPTIONS} {JAVA_PROPERTIES} -Dhttp.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port> -Dhttps.proxyHost=<proxy host> -Dhttps.proxyPort=<proxy port>" export JAVA_OPTIONS
```

or

```
EXTRA_JAVA_PROPERTIES="-Dhttp.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port> -Dhttps.proxyHost=<proxy host> -Dhttps.proxyPort=< proxy port> $ {EXTRA_JAVA_PROPERTIES}" export EXTRA_JAVA_PROPERTIES
```

3. Restart the agent.

9


Next Steps After Configuring the Domain

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

Performing Basic Administrative Tasks

Review the administrative tasks you will likely want to perform on a new domain.

Table 9-1 Basic Administration Tasks for a New Domain

Task	Description	More Information
Getting familiar with Fusion Middleware administration tools	Get familiar with various tools that you can use to manage your environment.	See Overview of Oracle Fusion Middleware Administration Tools in <i>Administering Oracle Fusion Middleware</i> .
	 Note: The WebLogic Server Administration Console has been removed. For comparable functionality, you will use the WebLogic Remote Console.	
Starting and stopping products and servers	Learn how to start and stop Oracle Fusion Middleware, including the Administration Server, Managed Servers, and components.	See Starting and Stopping Oracle Fusion Middleware in <i>Administering Oracle Fusion Middleware</i> .
Configuring Secure Sockets Layer (SSL)	Learn how to set up secure communications between Oracle Fusion Middleware components using SSL.	See Configuring SSL in Oracle Fusion Middleware in <i>Administering Oracle Fusion Middleware</i> .
Monitoring Oracle Fusion Middleware	Learn how to keep track of the status of Oracle Fusion Middleware components.	See Monitoring Oracle Fusion Middleware in <i>Administering Oracle Fusion Middleware</i> .
Understanding Backup and Recovery Procedures	Learn the recommended backup and recovery procedures for Oracle Fusion Middleware.	See Introduction to Backup and Recovery in <i>Administering Oracle Fusion Middleware</i> .

Performing Additional Domain Configuration Tasks

Review additional configuration tasks you will likely want to perform on a new domain.

Table 9-2 Additional Domain Configuration Tasks

Task	Description	More Information
Deploying Applications	Learn how to deploy your applications to Oracle Fusion Middleware.	See Deploying Applications in <i>Administering Oracle Fusion Middleware</i> .
Adding a Web Tier front-end to your domain	Oracle Web Tier hosts Web pages (static and dynamic), provides security and high performance along with built-in clustering, load balancing, and failover features. In particular, the Web Tier contains Oracle HTTP Server.	To install and configure Oracle HTTP Server in the WebLogic Server domain, see Configuring Oracle HTTP Server in a WebLogic Server Domain in <i>Installing and Configuring Oracle HTTP Server</i> . See also Installing Multiple Products in the Same Domain for important information.
Tuning and configuring Coherence for your topology	The standard installation topology includes a Coherence cluster that contains storage-enabled Managed Coherence Servers. This configuration is a good starting point for using Coherence, but depending upon your specific requirements, consider tuning and reconfiguring Coherence to improve performance in a production environment.	For more information about Coherence clusters, see Configuring and Managing Coherence Clusters in <i>Administering Clusters for Oracle WebLogic Server</i> . For information on tuning Coherence, see Performance Tuning in <i>Administering Oracle Coherence</i> . For information on storing HTTP session data in Coherence, see Using Coherence*Web with WebLogic Server in <i>Administering HTTP Session Management with Oracle Coherence*Web</i> . For more about creating and deploying Coherence applications, see Getting Started in <i>Developing Oracle Coherence Applications for Oracle WebLogic Server</i> .

Preparing Oracle Data Integrator for Development

Review tasks you can perform to prepare Oracle Data Integrator for development.

[Table 9-3](#) lists tasks that prepare Oracle Data Integrator for development.

Table 9-3 Tasks for Preparing Oracle Data Integrator for Development

Task	Description	More Information
Creating a logical agent	Multiple physical agents having identical roles in different environments can be grouped under the same logical agent. A logical agent is related to physical agents through contexts. When starting an execution, you indicate the logical agent and the context. Oracle Data Integrator will translate this information into a single physical agent that will receive the execution request.	Creating a Logical Agent
Managing the agents	Once your agents are defined in the topology, you can start, deploy, and stop your agents as needed.	Managing Agents
Administer your repositories	Additional tasks are available to create and connect to new master and work repositories, changing passwords, and other administrative tasks to help you customize your environment.	Administering Repositories
Create a server template for your Java EE agent	Oracle Data Integrator provides a Server Template Generation wizard to help you create a server template for a run-time agent.	Creating a Server Template for the Java EE Agent
Include libraries in classpath	Libraries that enable usage of ODI SDK APIs.	Client Libraries for SDK APIs

Preparing Your Environment for High Availability

Scaling out for high availability requires additional steps.

[Table 9-4](#) provides a list of tasks to perform if you want to scale out your standard installation environment for high availability.



See Also:

See Roadmap for Setting Up a High Availability Topology in *Administering Oracle Data Integrator*.

Table 9-4 Tasks Required to Prepare Your Environment for High Availability

Task	Description	More Information
Scaling out to multiple host computers	To enable high availability, it is important to provide failover capabilities to another host computer. That way, if one computer goes down, your environment can continue to serve the consumers of your deployed applications.	See Scaling Out a Topology (Machine Scale Out) in the <i>High Availability Guide</i> .

Table 9-4 (Cont.) Tasks Required to Prepare Your Environment for High Availability

Task	Description	More Information
Configuring high availability for your Web Tier components.	If you have added a Web tier front-end, then you must configure the Web Tier for high availability, as well as the WebLogic Server software.	See Configuring High Availability for Web Tier Components in <i>High Availability Guide</i> .
Setting up a front-end load balancer	A load balancer can be used to distribute requests across servers more evenly.	See Server Load Balancing in a High Availability Environment and Configuring Load Balancer Virtual Server Names and Ports in <i>High Availability Guide</i> .
Configuring Node Manager	Node Manager enables you to start, shut down, and restart the Administration Server and Managed Server instances from a remote location. This document assumes you have configured a per-domain Node Manager. Review the Node Manager documentation, for information on advanced Node Manager configuration options and features.	See Advanced Node Manager Configuration in <i>Administering Node Manager for Oracle WebLogic Server</i> .

10

Uninstalling or Reinstalling Oracle Data Integrator

Follow the instructions in this section to uninstall or reinstall Oracle Data Integrator.

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

Stopping the Oracle Data Integrator Agents

Before running the uninstaller, stop all running components/processes.

Stopping a Standalone Agent with Node Manager

To stop a standalone agent when using Node Manager:

1. Go to the `DOMAIN_HOME/bin` directory.
2. Enter the following command:
 - (UNIX) `./stopComponent.sh OracleDIAgent1`
 - (Windows) `stopComponent.cmd OracleDIAgent1`
3. At the prompt, enter the Administrator password for the domain.

Stopping a Standalone Agent Without Node Manager

To stop a standalone agent without Node Manager by using the `agent` command:

1. Go to the `DOMAIN_HOME/bin` directory.
2. Enter the following command:
 - (UNIX) `./agentstop.sh -NAME=OracleDIAgent1`
 - (Windows) `agentstop.cmd -NAME=OracleDIAgent1`

where, `-NAME` must be entered in capital letters, and the name of the agent (in this example, `OracleDIAgent1`) is case-sensitive.

AgentStop Command Parameters

The table below lists the different parameters for the command to stop the agent. The parameters are preceded by the "-" character and the possible values are preceded by the "=" character. When entering the command, consider the operating system specific syntax of the delimiters.

Table 10-1 Different Parameters for the Command to Stop the Agent.

Parameters	Description
<code>-PORT=<port></code>	This parameter is deprecated. It is used to stop a standalone agent on the same machine. It is a shortcut to <code>-AGENT_URL=http://localhost:<port>/oraclediagent</code> . The default port is 20910.
<code>-AGENT_URL=<agent_url></code>	URL of the standalone agent to stop. This parameter has precedence over the <code>AGENT_NAME</code> and <code>PORT</code> parameter is deprecated.
<code>-NAME=<agent name></code>	If this parameter is specified, the physical agent whose name is provided is terminated. This agent may be a local or a remote agent, and must be declared in the master repository. This parameter has precedence over the <code>PORT</code> parameter.
<code>-IMMEDIATE=<true false (default)></code>	If this parameter is set to 'true', the agent is terminated without waiting for completion of its running sessions. If it is set to 'false', the agent is terminated after all its running sessions reach completion or after the <code>MAX_WAIT</code> timeout is reached. The default value is 'false'.
<code>-MAX_WAIT=<stop timeout in millis></code>	This parameter can be used when <code>IMMEDIATE</code> is set to 'false'. It defines a timeout in milliseconds after which the agent is terminated regardless of the running sessions. Default is 0, meaning no timeout and the agent is terminated after all its running sessions reach completion.

Stopping Oracle Fusion Middleware

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

See Stopping an Oracle Fusion Middleware Environment in *Administering Oracle Fusion Middleware*.

Removing Your Database Schemas

Before you remove the Oracle home, Oracle recommends that you run the Repository Creation Utility (RCU) to remove database schemas associated with this domain.

Each domain has its own set of schemas, uniquely identified by a custom prefix. For more information about custom prefixes, see About Custom Prefixes in *Creating Schemas with the Repository Creation Utility*. This set of schemas cannot be shared with any other domain. For more information about creating schemas with the RCU, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

If there are multiple sets of schemas on your database, be sure to identify the schema prefix associated with the domain that you are removing.

For schema removal steps, see Dropping Schemas in *Creating Schemas with the Repository Creation Utility*.

Uninstalling the Software

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

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

Starting the Uninstall Wizard

To start the Uninstall Wizard:

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

Selecting the Product to Uninstall

Because multiple products exist in the Oracle home, ensure that you are uninstalling the correct product.

After you run the Uninstall Wizard, the Distribution to Uninstall screen opens. From the dropdown menu, select the product you want to remove and click **Uninstall**. The uninstallation program shows the screens listed in [Navigating the Uninstall Wizard Screens](#).

 **Note:**

You can uninstall Oracle Fusion Middleware Infrastructure after you uninstall Oracle Data Integrator software by running the Uninstall Wizard again. Before doing so, make sure that there are no other products using the Infrastructure; those products will no longer function once the Infrastructure is removed. You will not encounter the Distribution to Uninstall screen if no other software depends on Oracle Fusion Middleware Infrastructure. See [Uninstalling Oracle Fusion Middleware Infrastructure in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*](#).

Navigating the Uninstall Wizard Screens

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

[Table 10-2](#) describes the screens in the Uninstall Wizard. For information, click **Help** on the screen.

Table 10-2 Uninstall Wizard Screens and Descriptions

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

Removing the Oracle Home Directory Manually

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

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

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

Removing the Program Shortcuts on Windows Operating Systems

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

To remove the program shortcuts on Windows:

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

Removing the Domain and Application Data

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

To remove the domain and application data:

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

On Linux operating systems, if your Domain home directory is `/home/Oracle/config/domains/odi_domain`, enter the following command:

```
cd /home/Oracle/config/domains
```

```
rm -rf odi_domain
```

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

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

On Linux operating systems, if your Application home directory is /home/Oracle/config/applications/odi_domain, enter the following commands:

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

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

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

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

Save and exit the file when you are finished.

Removing Oracle Data Integrator Wallet Information

The ODI wallet holds encrypted data credentials that must be removed manually.

The uninstaller does not remove the ODI wallet, and it must be manually removed. Wallet information is stored in a directory inside the user's home directory as follows:

- (UNIX) /home/exampleuser/.odi
- (Windows) C:\Users\exampleuser\AppData\Roaming\odi

Reinstalling the Software

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

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

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

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

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

You can either:

- Select a different installation type. In this case, only the feature sets that do not exist in the Oracle home directory are installed.
- Select a different Oracle home directory.
- Installing in an existing, non-empty Oracle home.

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

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

A

Troubleshooting Oracle Data Integrator

Learn techniques for troubleshooting issues with Oracle Data Integrator (ODI).

General Troubleshooting Tips

Troubleshoot most common issues with Oracle Data Integrator installation.

If you encounter an error during installation:

- Read the *Oracle Fusion Middleware Release Notes* for the latest updates. The most current version of the release notes is available on the Oracle Technology Network (OTN) at:

<http://www.oracle.com/technetwork/documentation/index.html#middleware>

- Verify your system and configuration is certified. Refer to the *System Requirements and Supported Platforms for Oracle Fusion Middleware* document on the Oracle Fusion Middleware Supported System Configurations page:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>

- Verify your system meets the minimum system requirements. Refer to the *System Requirements and Specifications* document:

<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html>

- If you entered incorrect information on one of the installation screens, return to that screen by clicking **Back** until you see the screen.
- If an error occurred while the installer is copying or linking files:
 1. Note the error and review the installation log files.
 2. Remove the failed installation by following the steps in: .
 3. Correct the issue that caused the error.
 4. Restart the installation.

Troubleshooting Using Installation and Configuration Log Files

Use the installation and configuration log files to troubleshoot Oracle Data Integrator issues.

The installation and configuration log files contain useful information to help troubleshoot any issues you may encounter. For more information about these log files and their contents, see *Configuring Installation and Configuration Log Files* in *Installing Software with the Oracle Universal Installer*.

Need More Help?

Access additional help through My Oracle Support.

If this appendix does not solve the problem you encountered, try looking for a solution on My Oracle Support (formerly *OracleMetaLink*):

<https://support.oracle.com/>

If you are unable to find a solution for your problem, open a service request.

B

Secondary Topologies for Oracle Data Integrator

This appendix describes secondary (alternate) topologies for Oracle Data Integrator.

Oracle Data Integrator Topology with the Oracle Data Integrator Console

Oracle Data Integrator (ODI) Console is a web-based console that administrators and developers can use to configure the ODI topology and browse development artifacts.

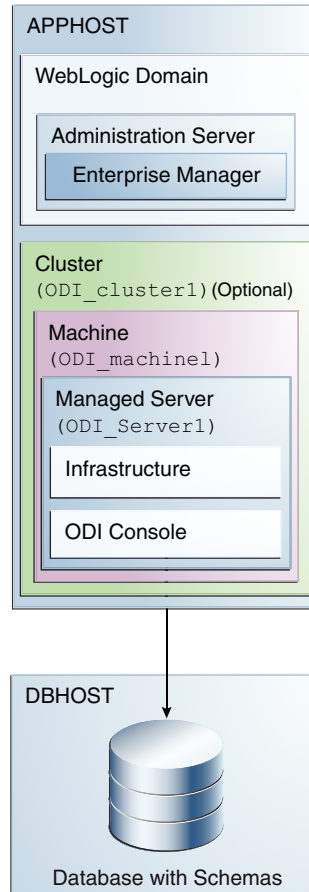
ODI Console is a web-based console available to:

- Administrators use Oracle Data Integrator Console to create and import repositories and to configure the topology.
- Business users and developers to browse development artifacts in this interface, using, for example, the Data Lineage and Flow Map features.

This ODI Console integrates with Oracle Fusion Middleware Control Console and allows Fusion Middleware administrators to drill down into the details of Oracle Data Integrator components and sessions.

[Figure B-1](#) illustrates an Oracle Data Integrator topology including the ODI Console.

Figure B-1 Secondary Installation Topology including the ODI Console



Using the Oracle Data Integrator Console

The Oracle Data Integrator (ODI) Console is available for selection in the Configuration Wizard.

To access the ODI Console, select **Oracle Data Integrator - Console - 14c (14.1.2.0.0)[odi]** in the Configuration Wizard **Templates** screen.

To use the ODI Console, follow the instructions in Using Oracle Data Integrator Console in *Administering Oracle Data Integrator*.

C

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

Consider that you have an unsupported JDK version installed on your machine. When you install and configure an Oracle Fusion Middleware product, the utilities, such as Configuration Wizard (`config.sh|exe`), OPatch, or RCU point to a default JDK. The supported JDK version for this release is `jdk17.0.12` and it carries security enhancements and bug fixes. You can upgrade the existing JDK to a newer version, and can have the complete product stack point to the newer version of the JDK.

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

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

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

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/jdk17.0.12`.

- 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.

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

(Linux) `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 `jdk17.0.12`.

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:

(Linux) `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:

(Linux) `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:

(Linux) `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 `jdk17.0.12`.

Updating the JDK Location in an Existing Domain Home

You must search the references to the current JDK manually, and replace those instances with the location of the new JDK.

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

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

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

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

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

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

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

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