

Oracle® Fusion Middleware

Upgrading Oracle Data Integrator

12c (12.2.1)

E55860-06

October 2016

Documentation for installers and system administrators that describes how to upgrade Oracle Data Integrator to 12c (12.2.1)

Oracle Fusion Middleware Upgrading Oracle Data Integrator, 12c (12.2.1)

E55860-06

Copyright © 2015, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Ingrid Snedecor

Contributing Authors: Priyanka Chheda, Lisa Jamen

Contributors: Ayush Ganeriwal, Srinivasa Ganti, Sachin Thatte

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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	vii
Audience	vii
Related Documents.....	vii
Conventions.....	vii

Part I Getting Started

1 Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1)	
1.1 About the Starting Points for an Oracle Data Integrator Upgrade.....	1-1
1.2 About the Oracle Data Integrator Standard Topologies	1-2
1.2.1 Oracle Data Integrator Standard Upgrade Topology for Standalone Agents.....	1-2
1.2.2 Oracle Data Integrator Standard Upgrade Topology for Standalone Collocated Agents	1-4
1.2.3 Oracle Data Integrator Standard Upgrade Topology for Java EE Agents	1-6
1.3 About the New Features for Oracle Data Integrator 12c	1-9
1.3.1 Standalone Agents are Managed by the WebLogic Management Framework.....	1-10
1.3.2 Standalone Agents are Installed in Their Own Directories.....	1-10
1.3.3 Repository Upgrade Validates Name Uniqueness for Objects.....	1-10
2 Pre-Upgrade Requirements	
2.1 Oracle Fusion Middleware Pre-Upgrade Checklist.....	2-2
2.2 Creating a Complete Backup	2-3
2.2.1 Backing Up the Schema Version Registry Table.....	2-4
2.2.2 Maintaining Custom Domain Environment Settings	2-4
2.3 Cloning Your Production Environment for Testing.....	2-5
2.4 Verifying Certification and System Requirements	2-7
2.4.1 Verify Your Environment Meets Certification Requirements	2-7
2.4.2 Verify System Requirements and Specifications	2-8
2.4.3 Verify That the Database Hosting Oracle Fusion Middleware Is Supported.....	2-11
2.4.4 Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware	2-11
2.5 Updating Policy Files when Using Enhanced Encryption (AES 256)	2-11
2.6 Purging Unused Data.....	2-12

2.7	Creating an Edition on the Server for Edition-Based Redefinition	2-12
2.8	Creating a Non-SYSDBA User to Run Upgrade Assistant	2-13
2.9	Identifying Existing Schemas Available for Upgrade	2-14
2.10	Configuring External Authentication for ODI	2-16

Part II Upgrading Oracle Data Integrator from 11g

3 Upgrading an Oracle Data Integrator Standalone Agent Environment from 11g

3.1	About the Oracle Data Integrator Standalone Agent Upgrade Process	3-2
3.2	Installing the Oracle Data Integrator Standalone Agent Environment	3-3
3.3	Creating the Required 12c Schemas with the RCU.....	3-5
3.4	Stopping Servers and Processes	3-9
3.5	Using the Upgrade Assistant to Upgrade Product Schemas.....	3-10
3.5.1	Starting the Upgrade Assistant	3-11
3.5.2	Upgrading Product Schemas Using the Upgrade Assistant	3-13
3.5.3	Verifying the Schema Upgrade	3-17
3.6	Upgrading Standalone System Component Configurations.....	3-18
3.6.1	Starting the Upgrade Assistant	3-18
3.6.2	Upgrading Standalone System Component Configurations Using the Upgrade Assistant	3-20
3.7	Starting Servers and Processes	3-23
3.8	Verifying the Upgrade	3-24

4 Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from 11g

4.1	About the Oracle Data Integrator Standalone Collocated Agent Upgrade Process	4-2
4.2	Installing the Oracle Data Integrator Standalone Collocated Agent Environment	4-5
4.3	Creating the Required 12c Schemas with the RCU.....	4-7
4.4	Running a Pre-Upgrade Readiness Check	4-11
4.4.1	About Running a Pre-Upgrade Readiness Check	4-11
4.4.2	Starting the Upgrade Assistant in Readiness Mode	4-12
4.4.3	Navigating the Upgrade Assistant Screens (Readiness Mode)	4-14
4.4.4	Understanding the Readiness Report.....	4-17
4.5	Stopping Servers and Processes	4-21
4.6	Using the Upgrade Assistant to Upgrade Product Schemas.....	4-22
4.6.1	Starting the Upgrade Assistant	4-22
4.6.2	Upgrading Product Schemas Using the Upgrade Assistant	4-25
4.6.3	Verifying the Schema Upgrade	4-29
4.7	Reconfiguring the Domain Using the Reconfiguration Wizard	4-29
4.7.1	Backing Up the Domain	4-30
4.7.2	Starting the Reconfiguration Wizard.....	4-31
4.7.3	Reconfiguring the Domain with the Reconfiguration Wizard	4-32
4.8	Upgrading Domain Component Configurations.....	4-34

4.8.1	Starting the Upgrade Assistant	4-34
4.8.2	Upgrading Domain Component Configurations Using the Upgrade Assistant	4-36
4.9	Starting Servers and Processes	4-38
4.10	Verifying the Upgrade	4-39

5 Upgrading an Oracle Data Integrator Java EE Agent Environment from 11g

5.1	About the Oracle Data Integrator Java EE Agent Upgrade Process	5-2
5.2	Installing the Oracle Data Integrator Java EE Environment	5-4
5.3	Creating the Required 12c Schemas with the RCU.....	5-6
5.4	Running a Pre-Upgrade Readiness Check	5-10
5.4.1	About Running a Pre-Upgrade Readiness Check	5-11
5.4.2	Starting the Upgrade Assistant in Readiness Mode	5-11
5.4.3	Navigating the Upgrade Assistant Screens (Readiness Mode)	5-14
5.4.4	Understanding the Readiness Report.....	5-16
5.5	Stopping Servers and Processes	5-20
5.6	Using the Upgrade Assistant to Upgrade Product Schemas.....	5-21
5.6.1	Starting the Upgrade Assistant	5-22
5.6.2	Upgrading Product Schemas Using the Upgrade Assistant	5-24
5.6.3	Verifying the Schema Upgrade	5-28
5.7	Reconfiguring the Domain Using the Reconfiguration Wizard	5-29
5.7.1	Backing Up the Domain	5-30
5.7.2	Starting the Reconfiguration Wizard.....	5-30
5.7.3	Reconfiguring the Domain with the Reconfiguration Wizard	5-31
5.8	Upgrading Domain Component Configurations.....	5-33
5.8.1	Starting the Upgrade Assistant	5-33
5.8.2	Upgrading Domain Component Configurations Using the Upgrade Assistant	5-35
5.9	Starting Servers and Processes	5-37
5.10	Verifying the Upgrade	5-39
5.11	Running a Pre-Upgrade Readiness Check	5-39
5.11.1	About Running a Pre-Upgrade Readiness Check	5-39
5.11.2	Starting the Upgrade Assistant in Readiness Mode	5-40
5.11.3	Navigating the Upgrade Assistant Screens (Readiness Mode)	5-42
5.11.4	Understanding the Readiness Report.....	5-45

Part III Upgrading Oracle Data Integrator from a Previous 12c Release

6 Upgrading an Oracle Data Integrator Standalone Agent Environment from a Previous 12c Release

7 Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from a Previous 12c Release

8 Upgrading an Oracle Data Integrator Java EE Agent Environment from a Previous 12c Release

Part IV Troubleshooting and Advanced Configurations

9 Troubleshooting the Upgrade

10 Advanced Upgrade Options

A Substitution API

Preface

This document describes how to upgrade an existing Oracle Data Integrator environment to 12c (12.2.1).

[Audience](#)

[Related Documents](#)

[Conventions](#)

Audience

This document is intended for system administrators who are responsible for installing, maintaining, and upgrading Oracle Data Integrator. It is assumed that readers have knowledge of the following:

- Oracle Fusion Middleware system administration and configuration
- Configuration parameters and expected behavior of the system being upgraded

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware documentation set:

- *Understanding Oracle Fusion Middleware*
- *Planning an Upgrade of Oracle Fusion Middleware*
- *Planning an Installation of Oracle Fusion Middleware*
- *Installing and Configuring Oracle Data Integrator*
- *Understanding Oracle Data Integrator*

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.

Convention	Meaning
<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.

Part I

Getting Started

This part of *Upgrading Oracle Data Integrator* introduces Oracle Data Integrator and provides steps you must perform to prepare for upgrading to 12c (12.2.1).

[Introduction to Upgrading Oracle Data Integrator to 12c \(12.2.1\)](#)

Before you begin, review all introductory information to understand the standard upgrade topologies and upgrade paths for Oracle Data Integrator 12c (12.2.1).

[Pre-Upgrade Requirements](#)

Before you begin the upgrade to Oracle Data Integrator 12c (12.2.1), you must perform pre-upgrade tasks such as backing up, cloning your current environment, and verifying that your system meets certified requirements.

Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1)

Before you begin, review all introductory information to understand the standard upgrade topologies and upgrade paths for Oracle Data Integrator 12c (12.2.1).

Following the pre-upgrade information and preparation steps, the procedures in this guide are divided into two sections:

- Upgrades from 11g, described in [Upgrading Oracle Data Integrator from 11g](#).
- Upgrades from a previous 12c release, described in [Upgrading Oracle Data Integrator from a Previous 12c Release](#).

Note: For general information about Fusion Middleware upgrade planning and other upgrade concepts and resources, see the following sections in *Planning an Upgrade of Oracle Fusion Middleware*:

- [Planning an Upgrade to Oracle Fusion Middleware 12c \(12.2.1\)](#)
 - [Understanding In-Place versus Out-of-Place Upgrades](#)
 - [Understanding the Basic 12c Upgrade Tasks](#)
-

The following topics describe the concepts related to upgrading Oracle Data Integrator:

[About the Starting Points for an Oracle Data Integrator Upgrade](#)

You can upgrade to Oracle Data Integrator 12c (12.2.1) from supported 11g and 12c releases.

[About the Oracle Data Integrator Standard Topologies](#)

The steps to upgrade Oracle Data Integrator to 12c (12.2.1) depend on the existing production topology (either 11g or a previous 12c release).

[About the New Features for Oracle Data Integrator 12c](#)

There are several changes between Oracle Data Integrator 11g and 12c.

1.1 About the Starting Points for an Oracle Data Integrator Upgrade

You can upgrade to Oracle Data Integrator 12c (12.2.1) from supported 11g and 12c releases.

Supported release starting points are:

- Oracle Data Integrator 11g Release 1 (11.1.1.6, 11.1.1.7, and 11.1.1.9)

- Oracle Data Integrator 12c (12.1.2 and 12.1.3)

The upgrade procedures in this guide explain how to upgrade an existing Oracle Data Integrator 11g or 12c environment to Oracle Fusion Middleware 12c (12.2.1). If your environment contains other components that also need to be upgraded, links to supporting documentation are provided.

If your existing version of Oracle Data Integrator is earlier than 11g Release 1 (11.1.1.6), you must first upgrade your software to one of the following supported versions before you can upgrade to 12c (12.2.1):

- To upgrade to 11g Release 1 (11.1.1.6), see [Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator](#) in the 11g Release 1 (11.1.1.6) documentation library.
- To upgrade to 11g Release 1 (11.1.1.7), see [Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator](#) in the 11g Release 1 (11.1.1.7) documentation library.

1.2 About the Oracle Data Integrator Standard Topologies

The steps to upgrade Oracle Data Integrator to 12c (12.2.1) depend on the existing production topology (either 11g or a previous 12c release).

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

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

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

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

[Oracle Data Integrator Standard Upgrade Topology for Standalone Agents](#)

[Oracle Data Integrator Standard Upgrade Topology for Standalone Collocated Agents](#)

[Oracle Data Integrator Standard Upgrade Topology for Java EE Agents](#)

1.2.1 Oracle Data Integrator Standard Upgrade Topology for Standalone Agents

When upgrading from 11g to 12c (12.2.1), the Oracle Data Integrator topology for standalone agents that are not configured in a WebLogic domain is updated as described here. If you are upgrading from a previous 12c release, the existing topology remains unchanged.

A standalone agent runs in a separate Java Virtual Machine (JVM) process. It connects to the work repository and to the source and target data servers via JDBC. Standalone agents can be installed on any server with a Java Virtual Machine installed. This type of agent is appropriate when you need to use a resource that is local to one of your data servers (for example, the file system or a loader utility installed with the database instance), and you do not want to install a Java EE application server on this machine.

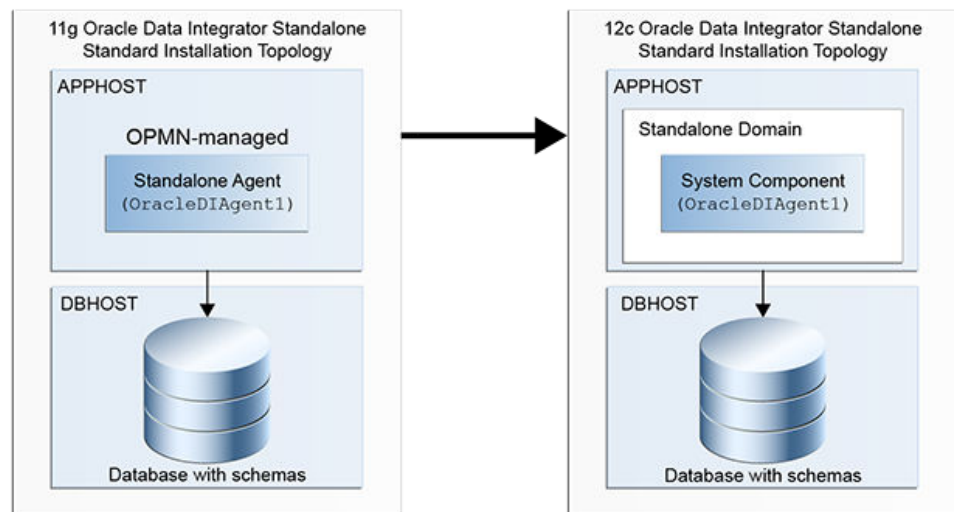
In 11g, the following standalone agent configurations were possible:

- Standalone agent as a standalone Oracle instance, not managed by OPMN. In this configuration, the Upgrade Assistant *does not* support the upgrade of standalone system component configurations to 12c (12.2.1).
- Standalone agent managed by OPMN. In this configuration, the Upgrade Assistant supports the upgrade of standalone system component configurations to 12c (12.2.1).

Figure 1-1 shows the 11g Oracle Data Integrator standard installation topology for a standalone agent managed by OPMN and the resulting 12c topology as it appears after you complete the upgrade procedures in this guide.

The upgrade roadmap and procedures for this topology are provided in [Upgrading an Oracle Data Integrator Standalone Agent Environment from 11g](#).

Figure 1-1 Oracle Data Integrator Standard Upgrade Topology for a Standalone Agent



All elements in this topology illustration are described in the following table.

Table 1-1 Description of the Elements in the Standalone Agent Standard Upgrade Topology for Standalone Agents

Element	Description and Links to Additional Documentation
11g Oracle Data Integrator Standalone Standard Installation Topology	<p>This is the label for the left side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 11g Oracle Data Integrator installer.</p> <p>It consists of a single standalone agent (OracleDIAgent1) on a single machine. The standalone agent is managed by OPMN.</p> <p>A relational database for the Master and Work Repository is also required, as shown in the figure.</p>

Table 1-1 (Cont.) Description of the Elements in the Standalone Agent Standard Upgrade Topology for Standalone Agents

Element	Description and Links to Additional Documentation
12c Oracle Data Integrator Standalone Agent Standard Installation Topology	This is the label for the right side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 12c Oracle Data Integrator distribution. It consists of a single standalone agent (OracleDIAgent1) configured in a standalone domain, along with a relational database for the Master and Work Repository.
APPHOST	Standard term used in Oracle documentation referring to the machine that is hosting the application tier.
DBHOST	Standard term used in Oracle documentation referring to the machine that is hosting the database.
Standalone Agent	The standalone Oracle Data Integrator agent that runs in a separate Java Virtual Machine (JVM) process. In 11g, the standalone agent is created directly as part of the installation.
Standalone Domain	For more information, see Standalone Domain in <i>Administrator's Guide for Oracle HTTP Server</i> .
System Component	In 12c, a standalone domain must be created before a standalone agent can be created. A system component corresponds to a standalone agent.

1.2.2 Oracle Data Integrator Standard Upgrade Topology for Standalone Collocated Agents

When upgrading from 11g to 12c (12.2.1), the Oracle Data Integrator topology for standalone agents that are configured in a WebLogic domain is updated as described here. If you are upgrading from a previous 12c release, the existing topology remains unchanged.

A standalone collocated agent runs in a separate Java Virtual Machine (JVM) process but is part of a WebLogic Server domain and controlled by a WebLogic Administration Server. Standalone collocated agents can be installed on any server with a Java Virtual Machine installed, but require a connection to the WebLogic Administration Server. This type of agent is appropriate when you need to use a resource that is local to one of your data servers but you want to centralize management of all applications in an enterprise application server.

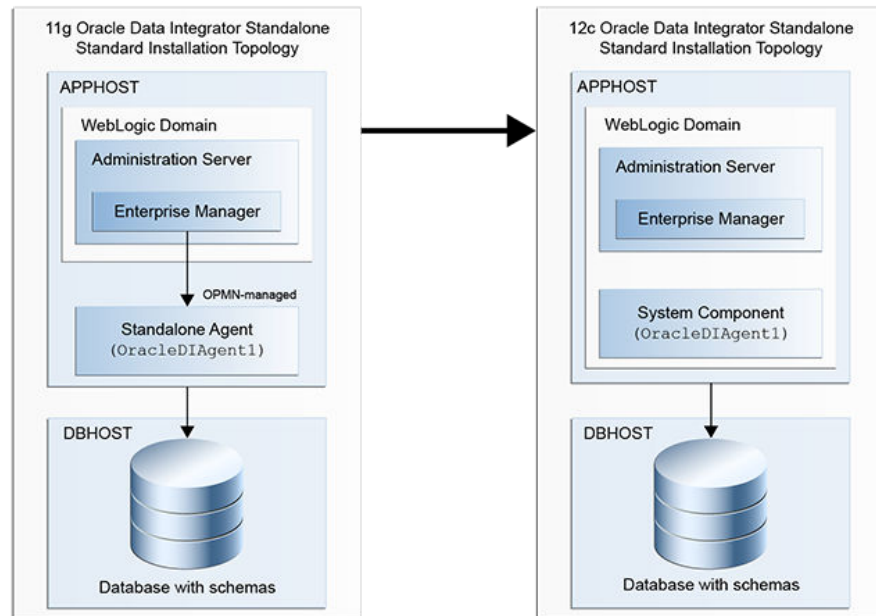
In 11g, the following standalone collocated agent configurations were possible:

- Standalone collocated agent not managed by OPMN. In this configuration, the Upgrade Assistant *does not* support the upgrade of WebLogic domain component configurations to 12c (12.2.1).
- Standalone collocated agent managed by OPMN. In this configuration, the Upgrade Assistant supports the upgrade of WebLogic domain component configurations to 12c (12.2.1).

Figure 1-2 shows the 11g Oracle Data Integrator standard installation topology for a standalone collocated agent managed by OPMN and the resulting 12c topology as it appears after you complete the upgrade procedures in this guide.

The upgrade roadmap and procedures for this topology are provided in [Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from 11g](#).

Figure 1-2 Oracle Data Integrator Standard Upgrade Topology for a Standalone Collocated Agent



All elements in this topology illustration are described in the following table.

Table 1-2 Description of the Elements in the Standard Upgrade Topology for Standalone Collocated Agents

Element	Description and Links to Additional Documentation
11g Oracle Data Integrator Standalone Standard Installation Topology	<p>This is the label for the left side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 11g Oracle Data Integrator installer.</p> <p>It consists of a single standalone agent (<code>OracleDIAgent1</code>) on a single machine. The standalone agent is managed by OPMN and is part of a WebLogic domain.</p> <p>A relational database for the Master and Work Repository is also required, as shown in the figure.</p>

Table 1-2 (Cont.) Description of the Elements in the Standard Upgrade Topology for Standalone Collocated Agents

Element	Description and Links to Additional Documentation
12c Oracle Data Integrator Standalone Standard Installation Topology	<p>This is the label for the right side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 12c Oracle Data Integrator distribution.</p> <p>It consists of a single standalone agent (OracleDIAgent1) configured in a WebLogic domain, along with a relational database for the Master and Work Repository.</p>
APPHOST	Standard term used in Oracle documentation referring to the machine that is hosting the application tier.
DBHOST	Standard term used in Oracle documentation referring to the machine that is hosting the database.
WebLogic Domain	<p>A logically related group of Java components (in this case, the Administration Server, Managed Servers, standalone agent, and other related software components).</p> <p>For more information, see <i>What Is an Oracle WebLogic Server Domain?</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Administration Server	<p>The central control entity of a domain that maintains the domain's configuration objects and distributes configuration changes to Managed Servers.</p> <p>For more information, see <i>What Is the Administration Server?</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Enterprise Manager	<p>Oracle Enterprise Manager Fusion Middleware Control.</p> <p>For more information, see <i>Oracle Enterprise Manager Fusion Middleware Control</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Standalone Agent	<p>The standalone Oracle Data Integrator agent that runs in a separate Java Virtual Machine (JVM) process.</p> <p>In 11g, the standalone agent is created directly as part of the installation.</p>
System Component	A system component corresponds to a standalone agent managed with the WebLogic Management Framework.

1.2.3 Oracle Data Integrator Standard Upgrade Topology for Java EE Agents

When upgrading from 11g to 12c (12.2.1), the Oracle Data Integrator topology for Java EE agents is updated as described here. If you are upgrading from a previous 12c release, the existing topology remains unchanged.

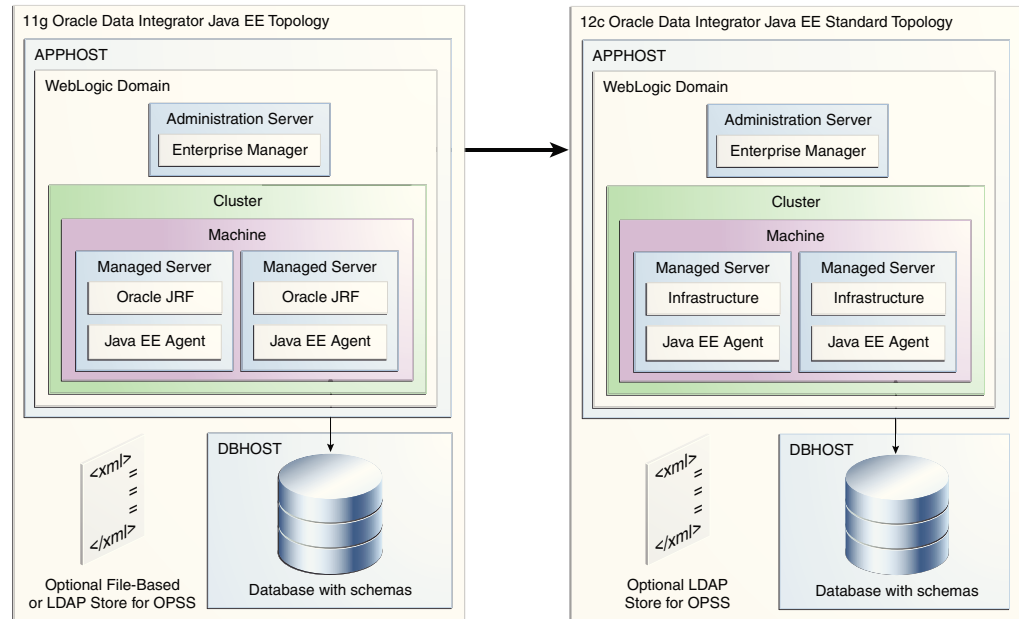
A Java EE agent is deployed as a web application in a Java EE application server, such as Oracle WebLogic Server. The Java EE agent can benefit from all the features of the application server (for example, JDBC data sources or clustering for Oracle WebLogic

Server). This type of agent is appropriate when there is a need for centralizing the deployment and management of all applications in an enterprise application server, or when you have requirements for high availability.

Figure 1-3 shows the 11g Oracle Data Integrator Java EE standard installation topology and the resulting 12c topology as it appears after you complete the upgrade procedures in this guide.

The upgrade roadmap and procedures for this topology are provided in [Upgrading an Oracle Data Integrator Java EE Agent Environment from 11g](#).

Figure 1-3 Oracle Data Integrator Standard Upgrade Topology for a Java EE Agent



All elements in this topology illustration are described in the following table.

Table 1-3 Description of the Elements in the Standard Upgrade Topology for Java EE Agents

Element	Description and Links to Additional Documentation
11g Oracle Data Integrator Java EE Topology	<p>This is the label for the left side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 11g Oracle Data Integrator installer.</p> <p>It consists of a single domain that contains a cluster of two managed servers, a Java EE agent, and the Administration Server. The domain also requires a relational database for the Master and Work Repository schema, and either an LDAP-based or file store for Oracle Platform Security Services (OPSS).</p>

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

Element	Description and Links to Additional Documentation
12c Oracle Data Integrator Java EE Standard Installation Topology	<p>This is the label for the right side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 12c Oracle Data Integrator distribution.</p> <p>Like the 11g topology, it also consists of a single domain that contains a cluster of two managed servers, a Java EE agent, the Administration Server, and a database for the Master and Work Repository schema.</p> <p>Unlike the 11g topology, only an LDAP based store can be used for OPSS; file-based stores are not allowed in 12c.</p>
APPHOST	Standard term used in Oracle documentation referring to the machine that is hosting the application tier.
DBHOST	Standard term used in Oracle documentation referring to the machine that is hosting the database.
WebLogic Domain	<p>A logically related group of Java components (in this case, the Administration Server, Managed Servers, Java EE agent, and other related software components).</p> <p>For more information, see <i>What Is an Oracle WebLogic Server Domain?</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Administration Server	<p>The central control entity of a domain that maintains the domain's configuration objects and distributes configuration changes to Managed Servers.</p> <p>For more information, see <i>What Is the Administration Server?</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Enterprise Manager	<p>Oracle Enterprise Manager Fusion Middleware Control.</p> <p>For more information, see <i>Oracle Enterprise Manager Fusion Middleware Control</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Cluster	<p>A collection of multiple WebLogic Server instances running simultaneously and working together.</p> <p>For more information, see <i>Understanding Managed Servers and Managed Server Clusters</i> in <i>Understanding Oracle Fusion Middleware</i>.</p>
Machine	<p>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.</p>

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

Element	Description and Links to Additional Documentation
Managed Server	<p>Host for your applications, application components, Web services, and their associated resources.</p> <p>For more information, see Understanding Managed Servers and Managed Server Clusters in <i>Understanding Oracle Fusion Middleware</i>.</p>
Java EE Agent	<p>A Java EE agent is a JEE application that is deployed and runs on a Managed Server configured in a WebLogic domain.</p> <p>For more information about these agents and how they fit into the overall Oracle Data Integrator topology, see Introduction to the Oracle Data Integrator Topology in <i>Developing Integration Projects with Oracle Data Integrator</i>.</p>
Oracle JRF	<p>Oracle JRF (Java Required Files) consists of those components not included in the Oracle WebLogic Server installation and that provide common functionality for Oracle business applications and application frameworks.</p> <p>JRF consists of several independently developed libraries and applications that are deployed into a common location. The components that are considered part of Java Required Files include Oracle Application Development Framework shared libraries and ODL logging handlers.</p>
Infrastructure	<p>Oracle Fusion Middleware 12c term (similar to Oracle JRF) that refers to the collection of services that include the following:</p> <ul style="list-style-type: none"> • Metadata repository (MDS) <p>This contains metadata for Oracle Fusion Middleware components, such as the Oracle Application Developer Framework.</p> <p>For more information, see What Is the Metadata Repository? in <i>Understanding Oracle Fusion Middleware</i>.</p> • Oracle Application Developer Framework (ADF) • Oracle Web Services Manager (OWSM)
Database with Schemas	<p>Represents a supported database, where the Oracle Fusion Middleware schemas have been created using the Repository Creation Utility (RCU).</p>

1.3 About the New Features for Oracle Data Integrator 12c

There are several changes between Oracle Data Integrator 11g and 12c.

To understand what's new in general in 12c, see New and Changed Features in *Understanding Oracle Fusion Middleware*.

If your environment includes Oracle WebLogic Server with Oracle ADF, see Key Differences Between Application Developer 11g and Infrastructure 12c.

The following key differences exist between Oracle Data Integrator 11g and 12c:

[Standalone Agents are Managed by the WebLogic Management Framework](#)

[Standalone Agents are Installed in Their Own Directories](#)

[Repository Upgrade Validates Name Uniqueness for Objects](#)

1.3.1 Standalone Agents are Managed by the WebLogic Management Framework

In 12c, system components, like the Oracle Data Integrator standalone agent, are managed by the WebLogic Management Framework.

For more information, see *What Is the WebLogic Management Framework?* in *Understanding Oracle Fusion Middleware*.

1.3.2 Standalone Agents are Installed in Their Own Directories

In 12c, standalone agents that are configured within a WebLogic Server domain are installed in a directory separate from the product installation directory.

For more information, see *Understanding the Standard Installation Topology for the Standalone Agent* in *Installing and Configuring Oracle Data Integrator*.

1.3.3 Repository Upgrade Validates Name Uniqueness for Objects

During the repository upgrade to 12c, the upgrade checks for duplicated names (according to the namespace rules) for a number of objects. The check is done before the actual upgrade starts and if duplicates are detected, then you must print a report with the list of duplicates in the upgrade log and stop the upgrade. You should then connect to the 11g repository using ODI Studio to manually fix the duplicates (either by rename or delete), and then restart the upgrade.

In 12c, the following objects must have unique names:

- Interface
- Folder
- Procedure/ KM

Note:

The Knowledge Modules must have unique names within the parent scope. This applies to either their own project or for global Knowledge Modules.

- Package
- Profile

Pre-Upgrade Requirements

Before you begin the upgrade to Oracle Data Integrator 12c (12.2.1), you must perform pre-upgrade tasks such as backing up, cloning your current environment, and verifying that your system meets certified requirements.

Oracle Fusion Middleware Pre-Upgrade Checklist

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

Creating a Complete Backup

Back up all system-critical files including all of the databases that host your Oracle Fusion Middleware schemas before you start the upgrade.

Cloning Your Production Environment for Testing

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

Verifying Certification and System Requirements

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

Updating Policy Files when Using Enhanced Encryption (AES 256)

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

Purging Unused Data

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

Creating an Edition on the Server for Edition-Based Redefinition

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

Creating a Non-SYSDBA User to Run Upgrade Assistant

Oracle recommends that you create a non-SYSDBA user to run the Upgrade Assistant. The user created using this procedure has the privileges required to complete the upgrade.

Identifying Existing Schemas Available for Upgrade

This optional task enables you to review the list of available schemas before you begin the upgrade by querying the schema version registry. The registry contains schema information such as version number,

component name and ID, date of creation and modification, and custom prefix.

[Configuring External Authentication for ODI](#)

Switch external authentication mode to internal authentication before you start the Upgrade Assistant.

2.1 Oracle Fusion Middleware Pre-Upgrade Checklist

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

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

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

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

Task	Description
<p>Required</p> <p>Create a complete backup of your existing environment.</p> <p>Creating a Complete Backup</p>	<p>Back up all system-critical files and database(s) that contain any schemas that are to be upgraded.</p> <p>Make sure that your backup includes the schema version registry table. See Backing Up the Schema Version Registry Table.</p> <p>If the upgrade fails, you must restore your pre-upgrade environment and begin the upgrade again.</p> <p>If you modified any of the startup scripts in your existing domain, you will need to copy them to temporary directory location (outside of the existing domain) during the upgrade and redeploy them after the upgrade. See Maintaining Custom Domain Environment Settings.</p>
<p>Optional</p> <p>Clone your production environment to use as an upgrade testing platform.</p> <p>Cloning Your Production Environment for Testing</p>	<p>In addition to creating a complete backup of your system files, Oracle strongly recommends that you clone your production environment. This environment can be used to test the upgrade.</p>
<p>Required</p> <p>Verify that you are installing and upgrading your product on a supported hardware and software configuration.</p> <p>Verifying Certification and System Requirements</p> <p>CAUTION: Do not attempt an upgrade if you are unable to use the latest supported operating system. As with all supported configurations, failure to comply with these requirements may cause your upgrade to fail.</p>	<p>Verify that your hardware and software configurations (including operating systems) are supported by the latest certifications and requirements documents.</p> <p>Oracle recommends that you verify this information right before you start the upgrade as the certification requirements are frequently updated.</p> <p>Make sure that you have applied the latest patches to your components before you upgrade.</p> <p>You must also make sure to use a supported JDK version before you install the 12c product distributions.</p>

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

Task	Description
<p>Required for 32-bit Operating Systems Only Migrate to a 64-bit operating system before you can upgrade. Migrating from a 32-Bit to a 64-Bit Operating System (Required only if you have a 32-Bit OS)</p>	<p>This is only required if you are currently running an unsupported 32-bit operating system.</p>
<p>Optional Update security policy files if you are using enhanced encryption (AES 256). Updating Policy Files when Using Enhanced Encryption (AES 256)</p>	<p>Some of the security algorithms used in Fusion Middleware 12c require additional policy files for the JDK. If you plan to use enhanced encryption, such as AES 256, Oracle recommends that you apply the latest required policy files to the JDK before you upgrade.</p>
<p>Optional Purge any outdated or unused data before you upgrade. Purging Unused Data</p>	<p>To optimize performance, Oracle strongly recommends that you purge data and objects that will not be used in the upgraded environment.</p>
<p>Required for Oracle Database Users Only Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c (12.2.1). Creating an Edition on the Server for Edition-Based Redefinition</p>	<p>If you are using an Edition-Based Redefinition (EBR) database, you must create the edition before starting the upgrade.</p>
<p>Optional Create a Non-SYSDBA user to run the Upgrade Assistant. Creating a Non-SYSDBA User to Run Upgrade Assistant</p>	<p>Oracle recommends that you create the FMW user to run Upgrade Assistant. User FMW can run the Upgrade Assistant without system administration privileges.</p>
<p>Optional Identify which schemas are currently in your domain before you begin. Identifying Existing Schemas Available for Upgrade</p>	<p>It is important that you know which schemas are in your pre-upgrade domain before you start the upgrade. You should know the schema owner names and passwords, as well as the versions of each schema.</p>

2.2 Creating a Complete Backup

Back up all system-critical files including all of the databases that host your Oracle Fusion Middleware schemas before you start the upgrade.

Performing a complete database backup prior to performing a schema upgrade is a prerequisite for running the Upgrade Assistant. The backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table so that you can restore the contents back to its pre-upgrade state if the upgrade fails.

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

For more information about creating a backup, see:

- Backing Up Your Oracle Fusion Middleware Environment in *Administering Oracle Fusion Middleware*
- Upgrading and Preparing Your Oracle Databases for 12c in *Planning an Upgrade of Oracle Fusion Middleware*

In addition to creating a complete backup of your system, you must also create a backup of your schema version registry and any custom settings you want to use in your upgraded environment. See:

Backing Up the Schema Version Registry Table

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

Maintaining Custom Domain Environment Settings

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

2.2.1 Backing Up the Schema Version Registry Table

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

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

Note: Performing these backups prior to performing a schema upgrade is a prerequisite for running the Upgrade Assistant. During the upgrade you will be required to acknowledge that backups have been performed.

2.2.2 Maintaining Custom Domain Environment Settings

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

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

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

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


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

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

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

Note:

If you are unable to create the `setUserOverrides` script before an upgrade, you need to reapply your settings as described in *Reapply Customizations to Startup Scripts in [Upgrading Oracle WebLogic Server](#)*.

2.3 Cloning Your Production Environment for Testing

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

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

Upgrades cannot be reversed. In most cases, if an error occurs, you must stop the upgrade and restore the entire environment from backup and begin the upgrade process from the beginning. Identifying potential upgrade issues in a development environment can eliminate unnecessary downtime.

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

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

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

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

Cloning an Oracle Data Integrator Production Environment for Testing

Cloning an Oracle Data Integrator production environment is a two-part process.

- Task 1: Clone (or copy) the existing production environment — referred to as **A** in the example below.
- Task 2: Run the full upgrade on the cloned environment — referred to as **B** in the example below.

Task 1: Clone existing production environment (A) for upgrade validation.

1. Install the ODI 11g or 12c version that matches the production ODI instance version on the test machine.
 - a. If your pre-upgrade environment is ODI 11g Release 1, see the [Installation Guide for Oracle Data Integrator](#) for the specific 11g release.
 - b. If your pre-upgrade environment is a previous ODI 12c release, see the [Installing and Configuring Oracle Data Integrator](#) guide for the specific 12c release.
2. Run the Repository Creation Utility (RCU) from the production version you just installed, and create new ODI repository schemas (B). You will want to perform testing on non-production schemas.

Note: If you are cloning an ODI 11g production environment:

When you create a new repository using the RCU, you are required to enter a repository ID for both Master and Work. The defaults are 0=Master and 1=Work. Make sure that the new IDs are different than those used in the existing production repository to avoid ID conflicts during repository import in Step 6.

This is not required if you are cloning an ODI 12c production environment.

3. If there are multiple work repositories, then you must create the other work repositories to match the production environment. For more information, see [Creating a Work Repository in the ODI installation guide](#) for your instance release.
4. Purge the execution logs to avoid exporting and importing excessive data as part of work repository export/import in the next step. See [Purging the Logs](#).
5. Export the ODI master and work repository from the production system using ODI export functionality. See [Exporting and Importing the Master Repository in Developing Integration Projects with Oracle Data Integrator](#).
6. Import the ODI master and work repository exports (created in Step 3) into the newly created ODI schemas (created in Step 2).

Your test environment should now be a clone of your production ODI environment.

Task 2: Upgrade the cloned environment (B) to test the upgrade

1. Review [Oracle Fusion Middleware Pre-Upgrade Checklist](#) to ensure have met all of the pre-upgrade requirements.
2. Install ODI 12c (12.2.1) and any other product distributions into a new Oracle home on the test machine. See Installing Oracle Data Integrator in *Installing and Configuring Oracle Data Integrator*.
3. Run a pre-upgrade Readiness Check with the Upgrade Assistant.
4. Follow the standard upgrade procedures for your environment. Make sure that you select the appropriate upgrade procedures for your environment. See [About the Oracle Data Integrator Standard Topologies](#).
5. Complete all post-upgrade configuration tasks and verify that the upgraded components work as expected.

2.4 Verifying Certification and System Requirements

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

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

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

[Verify Your Environment Meets Certification Requirements](#)

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

[Verify System Requirements and Specifications](#)

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

[Verify That the Database Hosting Oracle Fusion Middleware Is Supported](#)

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

[Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware](#)

Before you can install any Oracle Fusion Middleware product distributions, you must download and install a supported JDK on your system.

2.4.1 Verify Your Environment Meets Certification Requirements

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

Whenever new certifications occur, they are added to the appropriate certification document right away. New certifications can occur at any time, and for this reason the certification documents are kept outside of the documentation libraries and are available on Oracle Technology Network. For more information, see Certification Matrix for 12c (12.2.1).

2.4.2 Verify System Requirements and Specifications

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

The Oracle Fusion Middleware System Requirements and Specifications document should be used to verify that the requirements of the certification are met. For example, if the Certification Matrix for 12c (12.2.1) indicates that your product is certified for installation on 64-Bit Oracle Linux 7, the System Requirements and Specifications document should be used to verify that your Oracle Linux 7 system has met the required minimum specifications such as disk space, available memory, specific platform packages and patches, and other operating system-specific items. This document is updated as needed and resides outside of the documentation libraries. The latest version is available on Oracle Technology Network.

For a complete description of the system requirements for installing and upgrading to Oracle Fusion Middleware 12c, see Review System Requirements and Specifications.

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

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

[Migrating from a 32-Bit to a 64-Bit Operating System](#)

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

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

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

Make sure to validate the migration to ensure all your Oracle Fusion Middleware 11g software is working properly on the 64-bit machine, and only then perform the upgrade to Oracle Fusion Middleware 12c.

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

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

Upgrading an operating system typically involves the following:

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

[Procure the Hardware That Supports the Upgrade's 64-bit Software Requirement](#)

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

[Stop All Processes](#)

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

[Back Up All Files from the 32-bit Host Machine](#)

Make sure that you have created a complete backup of your entire 11g deployment before you begin the upgrade process. These files can be used if there is an issue during the migration and you have to restart the process.

[Set Up the Target 64-bit Machine with the 11g Host Name and IP Address](#)

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

[Restore the 11g Backup from 32-bit Host to 64-bit Host](#)

Restore the files you backed from the 32-bit host using the same directory structure that was used in 11g. The directory structure on the target machine must be identical to the structure of the host machine.

[Install the 12c Product Distributions on the Target Machine](#)

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

[Upgrade the Target 64-bit Environment Using the Standard Upgrade Procedure](#)

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

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

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

2.4.2.1.2 Stop All Processes

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

For example, to stop the Administration Server, enter the following command on UNIX:

```
DOMAIN_HOME/bin/stopWebLogic.sh username password [admin_url]
```

2.4.2.1.3 Back Up All Files from the 32-bit Host Machine

Make sure that you have created a complete backup of your entire 11g deployment before you begin the upgrade process. These files can be used if there is an issue during the migration and you have to restart the process.

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

For more information on backing up your 11g files, see [Backing Up Your Environment](#) in *Oracle Fusion Middleware Administrator's Guide*.

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

- 11g Domain Home
- 11g /nodemanager directory located in `$ORACLE_HOME/wlserver/common/`

Some of the backup and recovery procedures described in [Backing Up Your Environment](#) in *Oracle Fusion Middleware Administrator's Guide* are product-specific. Do not proceed with the upgrade until you have a complete backup.

2.4.2.1.4 Set Up the Target 64-bit Machine with the 11g Host Name and IP Address

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

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

2.4.2.1.5 Restore the 11g Backup from 32-bit Host to 64-bit Host

Restore the files you backed from the 32-bit host using the same directory structure that was used in 11g. The directory structure on the target machine must be identical to the structure of the host machine.

For detailed information about restoring your 11g files to the 64-bit target machine, see [Recovering Your Environment](#) in *Oracle Fusion Middleware Administrator's Guide*.

2.4.2.1.6 Install the 12c Product Distributions on the Target Machine

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

For detailed instructions on how to obtain 12c distributions, see [Understanding and Obtaining Product Distributions](#) in *Planning an Installation of Oracle Fusion Middleware*. To identify an installation user see [Selecting an Installation User](#). To understand the directory structure for installation and configuration, see [Understanding Directories for Installation and Configuration](#). Refer to the component-specific installation guides for the component(s) you are installing.

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

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

For a complete upgrade procedure, see the component-specific upgrade guide for the component(s) you are upgrading.

Note: The Node Manager upgrade procedure requires access to the original Node Manager files. Use the 11g Node Manger files that you backed up from the 32-bit source machine as part of [Back Up All Files from the 32-bit Host Machine](#).

2.4.3 Verify That the Database Hosting Oracle Fusion Middleware Is Supported

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

Review the Fusion Middleware database requirements before starting the upgrade to ensure that the database hosting Oracle Fusion Middleware is supported and has sufficient space to perform an upgrade. For more information, see Certification Matrix for 12c (12.2.1).

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

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

Before you can install any Oracle Fusion Middleware product distributions, you must download and install a supported JDK on your system.

Refer to the Oracle Fusion Middleware System Requirements and Specifications document to verify that the JDK you are using is supported.

At the time this document was published, the certified JDK was 1.8.0_60.

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

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

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

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

2.5 Updating Policy Files when Using Enhanced Encryption (AES 256)

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

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

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

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

2.6 Purging Unused Data

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

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

Purge the execution logs to avoid exporting and importing excessive data as part of work repository export/import in the next step. See Purging the Logs.

2.7 Creating an Edition on the Server for Edition-Based Redefinition

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

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

Note: This task must be completed by an Oracle Database user with DBA privileges.

Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c. The new edition for 12c must be a child of your existing 11g or 12c edition.

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

```
create edition Oracle_FMW_12_2_1 as child of
Oracle_FMW_11_1_1_7_0;
```

Where, `Oracle_FMW_11_1_1_7_0` is an example of the edition name you specified in RCU 11.1.1.7 when the 11.1.1.7 schemas were created. Be sure to provide the actual name used when creating the edition.

If the edition is created successfully, you get the following message:

```
Edition created.
```

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

```
ALTER DATABASE DEFAULT EDITION = Oracle_FMW_12_2_1;
```

2.8 Creating a Non-SYSDBA User to Run Upgrade Assistant

Oracle recommends that you create a non-SYSDBA user to run the Upgrade Assistant. The user created using this procedure has the privileges required to complete the upgrade.

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

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

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

Note: If you are upgrading an ORASDPM schema that was created using RCU 11g (11.1.1.1.4 or earlier), and you subsequently upgraded ORASDPM to 11g (11.1.1.6 or later), the FMW user will need to grant the `CREATE TABLE` privilege to user `prefix_ORASDPM` before upgrading to 12c (12.2.1).

```
grant CREATE TABLE to prefix_ORASDPM;
```

Where `prefix` is the name given to the schema when it was created.

In the example below, `welcome1` is the password. Make sure that you specify your actual password when granting privileges.

```
create user FMW identified by welcome1;
grant dba to FMW;
grant execute on DBMS_LOB to FMW with grant option;
grant execute on DBMS_OUTPUT to FMW with grant option;
grant execute on DBMS_STATS to FMW with grant option;
grant execute on sys.dbms_aqadm to FMW with grant option;
grant execute on sys.dbms_aqin to FMW with grant option;
grant execute on sys.dbms_aqjms to FMW with grant option;
grant execute on sys.dbms_aq to FMW with grant option;
grant execute on utl_file to FMW with grant option;
grant execute on dbms_lock to FMW with grant option;
grant select on sys.V_$INSTANCE to FMW with grant option;
grant select on sys.GV_$INSTANCE to FMW with grant option;
grant select on sys.V_$SESSION to FMW with grant option;
grant select on sys.GV_$SESSION to FMW with grant option;
grant select on dba_scheduler_jobs to FMW with grant option;
```

```
grant select on dba_scheduler_job_run_details to FMW with grant option;
grant select on dba_scheduler_running_jobs to FMW with grant option;
grant select on dba_aq_agents to FMW with grant option;
grant execute on sys.DBMS_SHARED_POOL to FMW with grant option;
grant select on dba_2pc_pending to FMW with grant option;
grant select on dba_pending_transactions to FMW with grant option;
grant execute on DBMS_FLASHBACK to FMW with grant option;
grant execute on dbms_crypto to FMW with grant option;
grant execute on DBMS_REPUTIL to FMW with grant option;
grant execute on dbms_job to FMW with grant option;
grant select on pending_trans$ to FMW with grant option;
grant select on dba_scheduler_job_classes to fmw with grant option;
grant select on SYS.DBA_DATA_FILES to FMW with grant option;
grant select on SYS.V_$ASM_DISKGROUP to FMW with grant option;
grant select on v$xsatrans$ to FMW with grant option;
grant execute on sys.dbms_system to FMW with grant option;
grant execute on DBMS_SCHEDULER to FMW with grant option;
grant select on dba_data_files to FMW with grant option;
grant execute on UTL_RAW to FMW with grant option;
grant execute on DBMS_XMLDOM to FMW with grant option;
grant execute on DBMS_APPLICATION_INFO to FMW with grant option;
grant execute on DBMS_UTILITY to FMW with grant option;
grant execute on DBMS_SESSION to FMW with grant option;
grant execute on DBMS_METADATA to FMW with grant option;
grant execute on DBMS_XMLGEN to FMW with grant option;
grant execute on DBMS_DATAPUMP to FMW with grant option;
grant execute on DBMS_MVIEW to FMW with grant option;
grant select on ALL_ENCRYPTED_COLUMNS to FMW with grant option;
grant select on dba_queue_subscribers to FMW with grant option;
grant execute on SYS.DBMS_ASSERT to FMW with grant option;
```

Note:

Oracle Database 11.2.0.3 Database Users ONLY: You must apply **Oracle Patch 13036331** before you begin the upgrade. Go to My Oracle Support to download the patch.

If you do not apply this patch, then you will have to grant additional privileges for some schemas.

2.9 Identifying Existing Schemas Available for Upgrade

This optional task enables you to review the list of available schemas before you begin the upgrade by querying the schema version registry. The registry contains schema information such as version number, component name and ID, date of creation and modification, and custom prefix.

Previewing a list of all the schemas that are available for an upgrade before you run the Upgrade Assistant helps you to decide if you want to select individual schemas to upgrade or allow the Upgrade Assistant to automatically select and upgrade all of the schemas in the domain.

To identify existing schemas in your domain that are available for upgrade:

1. If you are using an Oracle database, connect to the database as a user having Oracle DBA privileges, and run the following from SQL*Plus to get the current version numbers:

```

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

```

2. Examine the report that is generated. If the value in the `VERSION` column is at `11.1.1.6.0` or higher, and the `STATUS` column value is `VALID`, then the schema is supported for upgrade.

If an upgrade is not needed for a schema, the `schema_version_registry` table retains the schema at its pre-upgrade version.

3. Note the schema prefix name that was used for your existing schemas. You will use the same prefix when creating new 12c schemas.

Notes:

- If your existing schemas are not at a supported version, then you must upgrade them to a supported version before using the 12c (12.2.1) upgrade procedures. Refer to your pre-upgrade version documentation for more information.
- Some components, such as Oracle Enterprise Data Quality, Oracle GoldenGate Monitor and Oracle GoldenGate Veridata, support an upgrade from versions other than the standard Oracle Fusion Middleware supported versions.
- If you used a file-based policy store in 11g, then you must reassociate the file-based policy store with a database-based security store before running the Upgrade Assistant.

For more information see [Reassociating a File-Based Policy Store Before Upgrade](#).

- If you used an OID-based policy store in 11g, make sure that you create a new OPSS schema before you perform the upgrade. After the upgrade, the OPSS schema will remain an LDAP-based store.
- If you are upgrading an ORASDPM schema that was created using RCU 11g (11.1.1.4 or earlier), and you subsequently upgraded ORASDPM to 11g (11.1.1.6 or later), the FMW user will need to grant the `CREATE TABLE` privilege to user `prefix_ORASDPM` before upgrading to 12c (12.2.1).

```
grant CREATE TABLE to prefix_ORASDPM;
```

where `prefix` is the name given to the schema when it was created.

- You can only upgrade schemas for products that are available for upgrade in the Oracle Fusion Middleware release 12c (12.2.1). **Do not attempt to upgrade a domain that includes components that are not yet available for upgrade to 12c (12.2.1).**
-
-

2.10 Configuring External Authentication for ODI

Switch external authentication mode to internal authentication before you start the Upgrade Assistant.

If ODI is configured with external authentication mode, then you must switch the authentication mechanism to internal authentication before the upgrade so that the Upgrade Assistant can authenticate the given ODI credentials. This external authentication should be switched back again in the upgraded environment once the upgrade process is complete.

Note: This only applies if you are using external authentication. If you are not using external authentication, skip this step.

Refer to the following topics in *Administering Oracle Data Integrator*:

- Configuring External Authentication
- Switching an Existing Master Repository to External Authentication Mode

Part II

Upgrading Oracle Data Integrator from 11g

This part of *Upgrading Oracle Data Integrator* provides information about upgrading Oracle Data Integrator from an 11g release to 12c (12.2.1).

[Upgrading an Oracle Data Integrator Standalone Agent Environment from 11g](#)

You can upgrade an Oracle Data Integrator standalone agent environment that is not configured in a WebLogic domain from Oracle Fusion Middleware 11g to 12c (12.2.1) .

[Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from 11g](#)

You can upgrade an Oracle Data Integrator standalone agent environment that is configured in a WebLogic domain from Oracle Fusion Middleware 11g to 12c (12.2.1).

[Upgrading an Oracle Data Integrator Java EE Agent Environment from 11g](#)

You can upgrade an Oracle Data Integrator Java EE agent environment from Oracle Fusion Middleware 11g to 12c (12.2.1).

Upgrading an Oracle Data Integrator Standalone Agent Environment from 11g

You can upgrade an Oracle Data Integrator standalone agent environment that is not configured in a WebLogic domain from Oracle Fusion Middleware 11g to 12c (12.2.1) .

Follow the steps in the following topics to perform this upgrade:

[About the Oracle Data Integrator Standalone Agent Upgrade Process](#)

Review the flowchart and roadmap for an overview of the upgrade process for an Oracle Data Integrator Standalone agent that is not configured in a WebLogic domain.

[Installing the Oracle Data Integrator Standalone Agent Environment](#)

Before beginning your upgrade, download the Oracle Data Integrator 12c (12.2.1) distribution on the target system and install it using Oracle Universal Installer.

[Creating the Required 12c Schemas with the RCU](#)

When upgrading from 11g, you must use the Repository Creation Utility (RCU) to create the required 12c schemas before you begin the upgrade.

[Stopping Servers and Processes](#)

Before running the Upgrade Assistant to upgrade your schemas and config, shut down all processes and servers, including the Administration server and any managed servers.

[Using the Upgrade Assistant to Upgrade Product Schemas](#)

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

[Upgrading Standalone System Component Configurations](#)

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

[Starting Servers and Processes](#)

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

[Verifying the Upgrade](#)

After completing all the upgrade steps, verify that the upgrade was successful.

3.1 About the Oracle Data Integrator Standalone Agent Upgrade Process

Review the flowchart and roadmap for an overview of the upgrade process for an Oracle Data Integrator Standalone agent that is not configured in a WebLogic domain.

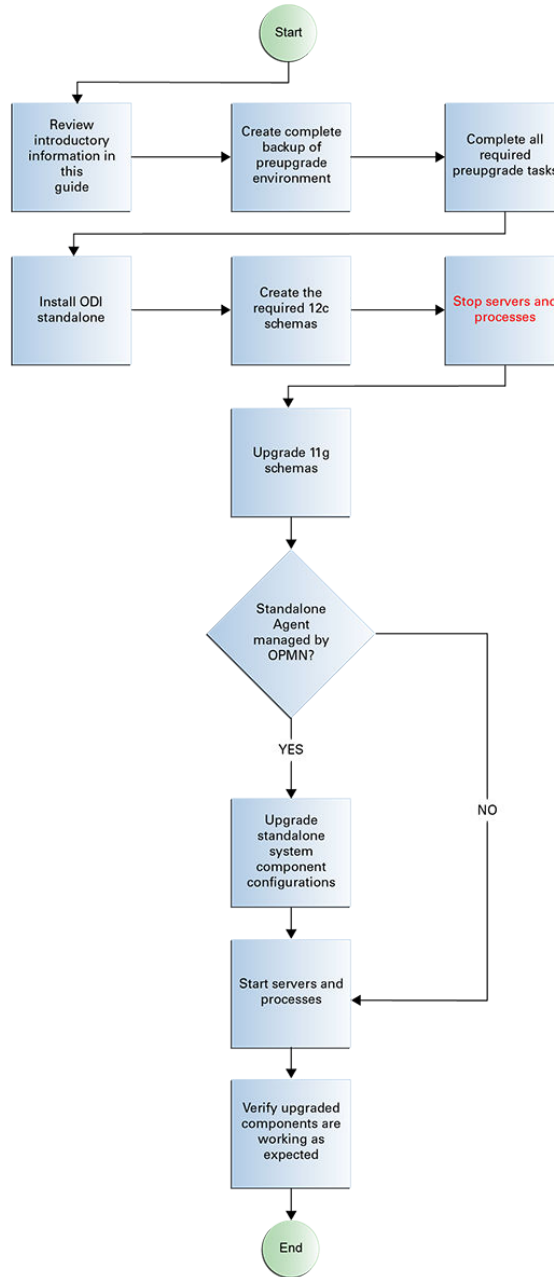


Table 3-1 Tasks for Upgrading an Oracle Data Integrator Standalone Agent from 11g

Task	Description
<p>Required</p> <p>If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.</p>	<p>See:</p> <ul style="list-style-type: none"> • Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1) • Pre-Upgrade Requirements
<p>Required</p> <p>Install Oracle Data Integrator Standalone 12c (12.2.1) in a new Oracle home.</p>	<p>Install the product software in a <i>new</i> Oracle home on the same host as the 11g production deployment before you begin the upgrade. In 12c, Oracle home is used to describe the 11g Middleware home.</p> <p>See Installing the Oracle Data Integrator Standalone Agent Environment.</p>
<p>Required</p> <p>Start the Repository Creation Utility (RCU) to create the required 12c database schemas.</p>	<p>Create the STB schema.</p> <p>See Creating the Required 12c Schemas with the RCU.</p>
<p>Required</p> <p>Shut down the 11g environment (stop all Administration and Managed Servers).</p>	<p>WARNING: Failure to shut down your servers during an upgrade may lead to data corruption.</p> <p>See Stopping Servers and Processes.</p>
<p>Required</p> <p>Start the Upgrade Assistant to upgrade the 11g database schemas and to migrate all active (in flight) instance data.</p>	<p>See Using the Upgrade Assistant to Upgrade Product Schemas.</p> <p>NOTE: The upgrade of active instance data is started automatically when running the Upgrade Assistant. Once the data is successfully upgraded to the new 12c (12.2.1) environment, you can close the Upgrade Assistant. The closed instances will continue to upgrade through a background process.</p>
<p>Required if managed by OPMN</p> <p>Start the Upgrade Assistant (again) to upgrade standalone system component configurations.</p>	<p>Run the Upgrade Assistant to upgrade the standalone agent's component configurations when the agent is managed by OPMN. The standalone component configuration upgrade does not support upgrade of a standalone agent that is not managed by OPMN.</p> <p>See Upgrading Standalone System Component Configurations.</p>
<p>Required</p> <p>Restart the servers and the 12c (12.2.1) instance.</p>	<p>When the upgrade process is complete, restart the 12c (12.2.1) instance.</p> <p>See Starting Servers and Processes.</p>
<p>Required</p> <p>Verify the upgrade.</p>	<p>Ensure all of the upgraded components are working as expected before deleting your backups.</p> <p>See Verifying the Upgrade.</p>

3.2 Installing the Oracle Data Integrator Standalone Agent Environment

Before beginning your upgrade, download the Oracle Data Integrator 12c (12.2.1) distribution on the target system and install it using Oracle Universal Installer.

To install the 12c (12.2.1) distribution:

1. Sign in to the target system where you want to install the 12c (12.2.1) product distribution.
2. Download the following 12c (12.2.1) product distribution from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle Data Integrator (fmw_12.2.1.0_odi_Disk1_1of2.zip and fmw_12.2.1.0_odi_Disk1_2of2.zip)
3. Change to the directory where you downloaded the 12c (12.2.1) product distribution.
4. Unzip the installer fmw_12.2.1.0_odi_Disk1_1of2.zip and fmw_12.2.1.0_odi_Disk1_2of2.zip files.

5. Enter the following command to start the installer for your product distribution and repeat the steps above to navigate through the installer screens:

(UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.0_odi.jar`

(Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.0_odi.jar`

6. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an Oracle product on this host.

Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location and click **Next**.

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

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

Click **Next**.

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

For more information about Oracle Fusion Middleware directory structure, see *Selecting Directories for Installation and Configuration* in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

10. On the Installation Type screen, select **Standalone Installation**.

Click **Next**.

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

To view the list of tasks that 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).

12. On the Installation Summary screen, verify the installation options you selected.

If you want to save these options to a response file, click **Save Response File** and enter the response file location and name. The response file collects and stores all the information that you have entered, and enables you to perform a silent installation (from the command line) at a later time.

Click **Install** to begin the installation.

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

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

3.3 Creating the Required 12c Schemas with the RCU

When upgrading from 11g, you must use the Repository Creation Utility (RCU) to create the required 12c schemas before you begin the upgrade.

For the ODI environments described in this guide, the required schemas are:

- ODI standalone agent (no WebLogic domain): STB
- ODI standalone collocated agent (with WebLogic domain): STB, OPSS, IAU, IAU_VIEWER, and IAU_APPEND
- ODI Java EE agent: STB, OPSS, IAU, IAU_VIEWER, and IAU_APPEND

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

If you are upgrading from 11g, refer to the Pre-Upgrade Checklist to identify the existing schemas in your domain. The following schemas must exist before you upgrade to 12c:

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

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

- **Oracle Platform Security Services (OPSS)** schema (*prefix_OPSS*). This schema is required if you are using an OID-based security store in 11g. This schema is automatically created when you run the Repository Creation Utility (RCU). The only supported LDAP-based OPSS security store is Oracle Internet Directory (OID). An LDAP-based policy store is typically used in production environments. You do not need to reassociate an OID-based security store before upgrade. While the Upgrade Assistant is running, you can select the OPSS schema. The Upgrade Assistant upgrades the OID-based security store automatically. **Note:** The 12c OPSS database schema is required so that you can reference the 12c schema during the reconfiguration of the domain. Your domain continues to use the OID-based security store after the upgrade is complete.
- **Audit** schemas. When upgrading audit services (*_IAU*), make sure that you select *_IAU_VIEWER* and *_IAU_APPEND* in addition to *_IAU*. The Upgrade Assistant will handle their creation for you automatically when they are selected.

To create the 12c schemas with the RCU:

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

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

2. If you have not already done so, set the *JAVA_HOME* environment variable and add *\$JAVA_HOME/bin* to *\$PATH*. The supported JDK version for 12c (12.2.1) is 1.8.0_60.
3. Go to the *oracle_common/bin* directory:
 - (UNIX) *ORACLE_HOME/oracle_common/bin*
 - (Windows) *ORACLE_HOME\oracle_common\bin*
4. Start the RCU:
 - (UNIX) *./rcu*
 - (Windows) *rcu.bat*
5. On the Welcome screen, click **Next**.
6. On the Create Repository screen, select **Create Repository** and then select **System Load and Product Load**.

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

Click **Next**.

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

Table 3-2 Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition

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

Table 3-3 Connection Credentials for MySQL Databases

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

Table 3-4 Connection Credentials for Microsoft SQL Server Databases

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

Table 3-5 Connection Credentials for IBM DB2 Databases

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

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

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

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

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

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

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

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

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

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

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

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

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

3.4 Stopping Servers and Processes

Before running the Upgrade Assistant to upgrade your schemas and config, shut down all processes and servers, including the Administration server and any managed servers.

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

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

To stop your Fusion Middleware environment, follow the steps below.

Step 1: Stop System Components

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

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

You can stop system components in any order.

Step 2: Stop the Managed Servers

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

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

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

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

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

Step 4: Stop the Administration Server

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

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

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

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

Note: If external password storage is set up for the repository, then the server hosting the credential store should be up and running so that the work repository password can be retrieved during upgrade. For more information, see *Setting Up External Password Storage in Administering Oracle Data Integrator*.

Step 5: Stop Node Manager

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

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

3.5 Using the Upgrade Assistant to Upgrade Product Schemas

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

The Upgrade Assistant allows you to upgrade individually selected schemas or all schemas associated with a domain. The option you select determines which Upgrade Assistant screens you will use.

Starting the Upgrade Assistant

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

(12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

Upgrading Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

3.5.1 Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

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

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

Upgrade Assistant Command Line Parameters

3.5.1.1 Upgrade Assistant Command Line Parameters

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

Table 3-6 Upgrade Assistant Command Line Parameters

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

Table 3-6 (Cont.) Upgrade Assistant Command Line Parameters

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

Table 3-6 (Cont.) Upgrade Assistant Command Line Parameters

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

3.5.2 Upgrading Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

Notes:

- If you are using external authentication, make sure that external authentication is changed to internal authentication.
 - **Edition-based redefinition (EBR) Users Only:** Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c. The new edition for 12c must be a child of your existing 11g or 12c edition. For more information on creating an edition on the server for redefinition, see *Creating an Edition on the Server for Editions-Based Redefinition in Planning an Upgrade of Oracle Fusion Middleware*.
-
-

To upgrade product schemas with the Upgrade Assistant:

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

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

2. On the Selected Schemas screen, select **Individually Selected Schemas**.

Caution: Upgrade only those schemas that are used to support your 12c (12.2.1) components. Do not upgrade schemas that are currently being used to support 11g or 12c components that are not included in Oracle Fusion Middleware 12c (12.2.1).

Click **Next**.

3. On the Available Components screen, select **Oracle Data Integrator** to upgrade the Master and Work Repository schema.
4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

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

5. On the ODI Schema screen, specify the database connection details for each schema you are upgrading:
 - Select the database type from the **Database Type** drop-down menu.
 - Enter the database connection details, and click **Connect**.
 - Select the schema you want to upgrade from the **Schema User Name** drop-down menu, and then enter the password for the schema. Be sure to use the correct schema prefix for the schemas you are upgrading.
6. On the ODI Options screen, select all of the options.

Table 3-7 ODI Options

Option	Description
Replace Knowledge Modules with mandatory updates	This selection replaces standard Knowledge Modules with the newest version. Customizations to Oracle installed Knowledge Modules will be overwritten. But if you copy an installed Knowledge Module and customize the Knowledge Module, the customizations are not lost.
Upgrade topology and security metadata	This selection replaces topology and security artifacts such as Technologies, Datatypes, Security Profiles and others with the newest version. Customizations of installed objects will be overwritten. If the object is copied and customized, then the customizations are not lost. For more information on how to upgrade manually, see <i>Developing Integration Projects with Oracle Data Integrator</i> in <i>Developing Integration Projects with Oracle Data Integrator</i> .

For information on advanced upgrade options, see [Advanced Upgrade Options](#).

7. On the ODI Supervisor screen, enter the Supervisor account credentials for the ODI repository to be upgraded.

The installed Supervisor account is `SUPERVISOR`. Check with your ODI administrator for proper Supervisor account name and password, supplied when prompted by the Repository Creation Utility (RCU) when creating the Master and Work repositories for ODI.

Note: When **All Schemas Used by a Domain** is selected, the Supervisor credentials for ODI are not pre-populated in the first instance as the domain does not contain them. If there are multiple ODI schemas, the Upgrade Assistant populates the user entry using the first set of credentials.

8. **For 11g to 12c upgrades only.** On the ODI Upgrade Key screen, use the auto-generated upgrade key to convert 11g IDs for repository objects into unique GUIDs, or specify your own key in the **Upgrade Key** field.

Recommendations:

- Edit the auto-generated key to provide a meaningful key that is easier to remember.
 - Note down the upgrade key so that the same upgrade key can be provided when the ODI objects are imported from the XML file.
-

ODI objects exist in ODI repositories and also in XML files exported from such repositories, which can be used, for example, in metadata exchanges between repositories. As such, there may be multiple copies of the same object, in different repositories and XML files.

In 12c, ODI uses GUIDs instead of internal numeric IDs for object identification. In order to make sure the object identity is preserved after upgrade, a deterministic algorithm is applied to calculate GUIDs from the internal IDs for existing objects (note that for new objects, ODI will generate random GUIDs).

Because the internal numeric IDs were not really universally unique, and were dependent on the repository ID to achieve a "pseudo-uniqueness," ODI allows the user to specify the upgrade key in order to reduce the likelihood of generating duplicated GUIDs. The upgrade key is fed into the GUID generation algorithm together with the internal numeric ID, to calculate the GUID.

Thus, choosing different upgrade keys protects from getting duplicated GUIDs for objects that accidentally have the same internal numeric IDs. However, when multiple copies of the same object exists (in a repository or exported in XML files), the same GUID should be produced for all copies of the object. For this reason, the same upgrade key must be used for all upgrade operations involving the copies of that particular object.

For example, suppose you have a project with 1001 as the ID in the 11g repository, and you also have a file exported from the same repository, which contains the same project (ID = 1001). In this case, the upgrade key used to upgrade the repository should be the same as the upgrade key used to import the XML file into the upgraded 12c repository. This ensures that the project object in the import file will be properly matched with the project object in the repository (when using one of SYNONYM import modes). However, if there is an 11g XML export file provided from a source containing objects created in another repository of which you have no information, there is a chance that it may contain a project that accidentally has the same internal ID (1001). In this case, to protect from erroneous object matching, which may corrupt the metadata, a different, custom upgrade key should be used when importing that file into the repository.

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

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

Note:

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

10. On the Upgrade Summary screen, review the summary of the options you have selected for schema upgrade.

Verify that the correct Source and Target Versions are listed for each schema you intend to upgrade.

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

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

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

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

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

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

Click **Next**.

12. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard.

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

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

3.5.3 Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

If you are using an Oracle database, connect to the database as a user having Oracle DBA privileges, and run the following from SQL*Plus to get the current version numbers:

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

In the query result:

- Check that the number in the `VERSION` column matches the latest version number for that schema. For example, verify that the schema version number is `12.2.1.0`. Note, however, that not all schema versions will be updated. Some schemas do not require an upgrade to this release and will retain their pre-upgrade version number.
- The `STATUS` field will be either `UPGRADING` or `UPGRADED` during the schema patching operation, and will become `VALID` when the operation is completed.
- If the status appears as `INVALID`, the schema update failed. You should examine the logs files to determine the reason for the failure.
- Synonym objects owned by `IAU_APPEND` and `IAU_VIEWER` will appear as `INVALID`, but that does not indicate a failure.

They become invalid because the target object changes after the creation of the synonym. The synonyms objects will become valid when they are accessed. You can safely ignore these `INVALID` objects.

3.6 Upgrading Standalone System Component Configurations

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

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

Upgrading Standalone System Component Configurations Using the Upgrade Assistant

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

3.6.1 Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

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

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

[Upgrade Assistant Command Line Parameters](#)

3.6.1.1 Upgrade Assistant Command Line Parameters

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

Table 3-8 Upgrade Assistant Command Line Parameters

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

Table 3-8 (Cont.) Upgrade Assistant Command Line Parameters

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

3.6.2 Upgrading Standalone System Component Configurations Using the Upgrade Assistant

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

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

To upgrade standalone system component configurations with the Upgrade Assistant:

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

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

2. On the next screen:
 - Select **Standalone System Component Configurations**, then select **Create a New Domain**.

Note: Beginning with 12c, standalone system components will have their own standalone domain. When you are upgrading your 11g standalone system components (which had no previous domain associations), you must first create a new standalone domain for your system components.

- In the **Domain Directory** field, specify the full path of the domain you are creating. Oracle recommends that you locate your domain home in accordance with the directory structure summarized in Understanding the Recommended Directory Structure in *Planning an Installation of Oracle Fusion Middleware*, where the domain home is located outside the Oracle home directory. This directory structure will help you avoid issues when you need to upgrade or reinstall your software.

Tip: The **Update an Existing Domain** option might be used in a situation where an upgrade was already performed and the domain created, either from another system component upgrade, or from a partial previous Oracle Data Integrator upgrade. These are examples of a situation where you would not need to create a new domain.

Click **Next**.

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

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

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

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

5. On the Instance Directories screen, specify the location of one or more Oracle instance directories to be upgraded.
6. On the Node Manager screen, specify the credentials of the Node Manager that will be used to create a domain during the upgrade of standalone system components.
7. On the Examine screen, review the status of the Upgrade Assistant as it examines each standalone component, verifying that the standalone component configuration is ready for upgrade. If the status is **Examine finished**, click **Next**.

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

Note:

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

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

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

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

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

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

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

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

Click **Next**.

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

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

3.7 Starting Servers and Processes

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

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

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

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

Step 1: Start the Administration Server

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

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

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

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

Step 2: Start Node Manager

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

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

Step 3: Start Oracle Identity Management Components

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

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

Step 4: Start the Managed Servers

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

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

When prompted, enter your user name and password.

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

Step 5: Start System Components

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

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

You can start system components in any order.

3.8 Verifying the Upgrade

After completing all the upgrade steps, verify that the upgrade was successful.

To verify the upgrade:

- Start the Node Manager and standalone agent.

For more information, see Starting a Standalone Agent with Node Manager in *Installing and Configuring Oracle Data Integrator*.

Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from 11g

You can upgrade an Oracle Data Integrator standalone agent environment that is configured in a WebLogic domain from Oracle Fusion Middleware 11g to 12c (12.2.1).

Follow the steps in the following topics to perform this upgrade:

[About the Oracle Data Integrator Standalone Collocated Agent Upgrade Process](#)

Review the process flowchart for an overview of the upgrade process for an Oracle Data Integrator standalone agent that is configured in a WebLogic domain.

[Installing the Oracle Data Integrator Standalone Collocated Agent Environment](#)

Before beginning your upgrade, download the Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12c (12.2.1) distributions on the target system and install them using Oracle Universal Installer.

[Creating the Required 12c Schemas with the RCU](#)

When upgrading from 11g, you must use the Repository Creation Utility (RCU) to create the required 12c schemas before you begin the upgrade.

[Running a Pre-Upgrade Readiness Check](#)

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

[Stopping Servers and Processes](#)

Before running the Upgrade Assistant to upgrade your schemas and config, shut down all processes and servers, including the Administration server and any managed servers.

[Using the Upgrade Assistant to Upgrade Product Schemas](#)

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

[Reconfiguring the Domain Using the Reconfiguration Wizard](#)

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1).

Upgrading Domain Component Configurations

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

Starting Servers and Processes

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

Verifying the Upgrade

After completing all the upgrade steps, verify that the upgrade was successful.

4.1 About the Oracle Data Integrator Standalone Collocated Agent Upgrade Process

Review the process flowchart for an overview of the upgrade process for an Oracle Data Integrator standalone agent that is configured in a WebLogic domain.

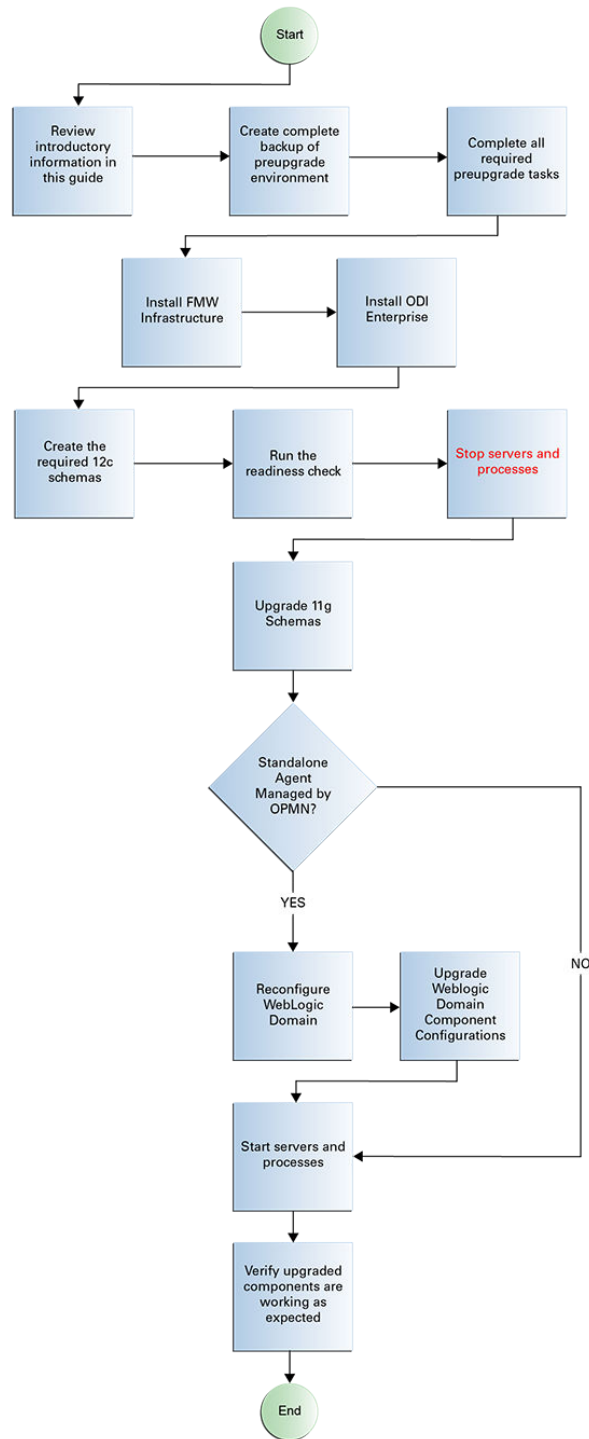


Table 4-1 Tasks for Upgrading Oracle Data Integrator Standalone Collocated Agent from 11g

Task	Description
<p>Required</p> <p>If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.</p>	<p>See:</p> <ul style="list-style-type: none"> • Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1) • Pre-Upgrade Requirements
<p>Required</p> <p>Install Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12c (12.2.1) in a new Oracle home.</p>	<p>See Installing the Oracle Data Integrator Standalone Collocated Agent Environment.</p>
<p>Required</p> <p>Start the Repository Creation Utility (RCU) to create the required 12c database schemas.</p>	<p>Create the STB, OPSS, IAU, IAU_VIEWER, and IAU_APPEND schemas.</p> <p>See Creating the Required 12c Schemas with the RCU.</p>
<p>Optional</p> <p>Run a pre-upgrade readiness check.</p>	<p>See Running a Pre-Upgrade Readiness Check.</p>
<p>Required</p> <p>Shut down the 11g environment (stop all Administration and Managed Servers).</p>	<p>WARNING: Failure to shut down your servers during an upgrade may lead to data corruption.</p> <p>See Stopping Servers and Processes.</p>
<p>Required</p> <p>Start the Upgrade Assistant to upgrade the 11g database schemas and to migrate all active (in flight) instance data.</p>	<p>See Using the Upgrade Assistant to Upgrade Product Schemas.</p> <p>NOTE: The upgrade of active instance data is started automatically when running the Upgrade Assistant. Once the data is successfully upgraded to the new 12c (12.2.1) environment, you can close the Upgrade Assistant. The closed instances will continue to upgrade through a background process.</p>
<p>Required</p> <p>Start the Reconfiguration Wizard to reconfigure the domain and node manager.</p>	<p>Run the Configuration Wizard to update the existing domain to use the newly installed software.</p> <p>See Reconfiguring the Domain Using the Reconfiguration Wizard.</p>
<p>Required</p> <p>Start the Upgrade Assistant (again) to upgrade domain configurations.</p>	<p>Run the Upgrade Assistant to update the reconfigured domain's component configurations. The domain component configuration upgrade does not support upgrade of a standalone agent that is not managed by OPMN.</p> <p>See Upgrading Domain Component Configurations.</p>
<p>Required</p> <p>Restart the servers and the 12c (12.2.1) instance.</p>	<p>When the upgrade process is complete, restart the 12c (12.2.1) instance.</p> <p>See Starting Servers and Processes.</p>
<p>Required</p> <p>Verify the upgrade.</p>	<p>Ensure all of the upgraded components are working as expected before deleting your backups.</p> <p>See Verifying the Upgrade.</p>

4.2 Installing the Oracle Data Integrator Standalone Collocated Agent Environment

Before beginning your upgrade, download the Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12c (12.2.1) distributions on the target system and install them using Oracle Universal Installer.

Note: You must install the Oracle Fusion Middleware Infrastructure distribution first before installing other Fusion Middleware products, when Infrastructure is required for the upgrade.

To install the 12c (12.2.1) distributions:

1. Sign in to the target system where you want to install the 12c (12.2.1) product distribution.
2. Download the following 12c (12.2.1) product distributions from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle Fusion Middleware Infrastructure (fmw_12.2.1.0_infrastructure_generic.jar)
 - Oracle Data Integrator (fmw_12.2.1.0_odi_Disk1_1of2.zip and fmw_12.2.1.0_odi_Disk1_2of2.zip)
3. Change to the directory where you downloaded the 12c (12.2.1) product distribution.
4. Unzip the installer fmw_12.2.1.0_odi_Disk1_1of2.zip and fmw_12.2.1.0_odi_Disk1_2of2.zip files.
5. Start the installation program for Oracle Fusion Middleware Infrastructure:
 - (UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.0_infrastructure_generic.jar`
 - (Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.0_infrastructure_generic.jar`
6. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an Oracle product on this host.

Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location and click **Next**.

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

7. On the Welcome screen, review the information to make sure that you have met all the prerequisites. Click **Next**.
8. On the Auto Updates screen, select **Skip Auto Updates**. Options are:

- **Skip Auto Updates:** If you do not want your system to check for software updates at this time.
- **Select patches from directory:** To navigate to a local directory if you downloaded patch files.
- **Search My Oracle Support for Updates:** To automatically download software updates if you have a My Oracle Support account. You must enter Oracle Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

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

For more information about Oracle Fusion Middleware directory structure, see *Selecting Directories for Installation and Configuration in Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

10. On the Installation Type screen, select the following:

- For Infrastructure, select **Fusion Middleware Infrastructure**
- For Oracle Data Integrator, select **Enterprise Installation**

Click **Next**.

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

To view the list of tasks that 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).

12. On the Installation Summary screen, verify the installation options you selected.

If you want to save these options to a response file, click **Save Response File** and enter the response file location and name. The response file collects and stores all the information that you have entered, and enables you to perform a silent installation (from the command line) at a later time.

Click **Install** to begin the installation.

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

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

15. After you have installed Oracle Fusion Middleware Infrastructure, enter the following command to start the installer for your product distribution and repeat the steps above to navigate through the installer screens:

```
(UNIX) JDK_HOME/bin/java -jar fmw_12.2.1.0_odi.jar
```

```
(Windows) JDK_HOME\bin\java -jar fmw_12.2.1.0_odi.jar
```

4.3 Creating the Required 12c Schemas with the RCU

When upgrading from 11g, you must use the Repository Creation Utility (RCU) to create the required 12c schemas before you begin the upgrade.

For the ODI environments described in this guide, the required schemas are:

- ODI standalone agent (no WebLogic domain): STB
- ODI standalone collocated agent (with WebLogic domain): STB, OPSS, IAU, IAU_VIEWER, and IAU_APPEND
- ODI Java EE agent: STB, OPSS, IAU, IAU_VIEWER, and IAU_APPEND

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

If you are upgrading from 11g, refer to the Pre-Upgrade Checklist to identify the existing schemas in your domain. The following schemas must exist before you upgrade to 12c:

- **Service Table** schema (*prefix_STB*). This schema is new in 12c and is required for domain-based upgrades. It stores basic schema configuration information (for example, schema prefixes and passwords) that can be accessed and used by other Oracle Fusion Middleware components during the domain creation. This schema is automatically created when you run the Repository Creation Utility (RCU), where you specify the existing schema owner prefix that you used for your other 11g schemas. **Note:** If the Service Table schema does not exist, you may encounter the error message UPGAST-00328 : The schema version registry table does not exist on this database. If that happens it is necessary to create the service table schema in order to run Upgrade Assistant.
- **Oracle Platform Security Services (OPSS)** schema (*prefix_OPSS*). This schema is required if you are using an OID-based security store in 11g. This schema is automatically created when you run the Repository Creation Utility (RCU). The only supported LDAP-based OPSS security store is Oracle Internet Directory (OID). An LDAP-based policy store is typically used in production environments. You do not need to reassociate an OID-based security store before upgrade. While the Upgrade Assistant is running, you can select the OPSS schema. The Upgrade Assistant upgrades the OID-based security store automatically. **Note:** The 12c OPSS database schema is required so that you can reference the 12c schema during the reconfiguration of the domain. Your domain continues to use the OID-based security store after the upgrade is complete.
- **Audit** schemas. When upgrading audit services (*_IAU*), make sure that you select *_IAU_VIEWER* and *_IAU_APPEND* in addition to *_IAU*. The Upgrade Assistant will handle their creation for you automatically when they are selected.

To create the 12c schemas with the RCU:

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

```

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

```

2. If you have not already done so, set the `JAVA_HOME` environment variable and add `$JAVA_HOME/bin` to `$PATH`. The supported JDK version for 12c (12.2.1) is 1.8.0_60.
3. Go to the `oracle_common/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/bin`
 - (Windows) `ORACLE_HOME\oracle_common\bin`
4. Start the RCU:
 - (UNIX) `./rcu`
 - (Windows) `rcu.bat`
5. On the Welcome screen, click **Next**.
6. On the Create Repository screen, select **Create Repository** and then select **System Load and Product Load**.

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

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

Table 4-2 Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition

Option	Description and Example
Host Name	Specify the name of the server where your database is running in the following format: <code>examplehost.exampledomain.com</code> For Oracle RAC databases, specify the VIP name or one of the node names in this field.
Port	Specify the port number for your database. The default port number for Oracle databases is 1521.

Table 4-2 (Cont.) Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition

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

Table 4-3 Connection Credentials for MySQL Databases

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

Table 4-4 Connection Credentials for Microsoft SQL Server Databases

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

Table 4-5 Connection Credentials for IBM DB2 Databases

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

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

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

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

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

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

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

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

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

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

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

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

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

4.4 Running a Pre-Upgrade Readiness Check

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

About Running a Pre-Upgrade Readiness Check

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

Starting the Upgrade Assistant in Readiness Mode

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

Navigating the Upgrade Assistant Screens (Readiness Mode)

You must navigate all the Upgrade Assistant screens that appear, based on your primary selection, to complete the Readiness Check. In addition, if you want to save the response file for performing readiness check on other domains/system, you can do that on the Readiness Summary screen.

Understanding the Readiness Report

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

4.4.1 About Running a Pre-Upgrade Readiness Check

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

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

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

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

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

Note:

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

4.4.2 Starting the Upgrade Assistant in Readiness Mode

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

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

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

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

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

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

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

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

[Upgrade Assistant Command Line Parameters](#)

4.4.2.1 Upgrade Assistant Command Line Parameters

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

Table 4-6 Upgrade Assistant Command Line Parameters

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

Table 4-6 (Cont.) Upgrade Assistant Command Line Parameters

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

Table 4-6 (Cont.) Upgrade Assistant Command Line Parameters

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

4.4.3 Navigating the Upgrade Assistant Screens (Readiness Mode)

You must navigate all the Upgrade Assistant screens that appear, based on your primary selection, to complete the Readiness Check. In addition, if you want to save the response file for performing readiness check on other domains/system, you can do that on the Readiness Summary screen.

To complete the Readiness Check:

1. The Welcome screen provides an overview of the readiness check. Review the information on this screen and click **Next**.
2. You can perform readiness check on schemas or domain configurations that are at a supported starting point.

On the Readiness Check Type screen, select one of the following options depending upon the type of readiness check you want to perform:

- **Individually Selected Schemas**
 - **Domain Based**
3. To review and perform readiness check on specific schemas, select **Individually Selected Schemas**.

The **Individually Selected Schemas** option allows you to select the schemas you want to review before the upgrade. When you perform the readiness check on the

schemas, the readiness check report tells you whether a schema is supported for an upgrade, or where an upgrade is needed. Click **Next**.

Note: When you select the **Individually Selected Schemas** option, the screen name changes from Readiness Check Type to Selected Schema.

4. To perform a readiness check per domain, select **Domain Based**.

The **Domain Based** option allows you to check all of the upgrade-eligible schemas and/or component configurations used by the domain. The Upgrade Assistant detects all of the schemas for you. You can check schemas and component configurations at the same time. Or, if you prefer, you can select to review one thing at a time by selecting one of the following options:

- **Include checks for all schemas**

Select this option to enable the Upgrade Assistant to discover and review all components that have a schema available to upgrade.

- **Include checks for all configurations**

Select this option to review component configurations for a managed WebLogic Server domain.

Note: When you select the **Domain Based** option, the screen name changes from Readiness Check Type to Schemas and Configuration.

Specify the **Domain Directory** by entering the path to your domain or by clicking **Browse** to use the navigation tree.

If you want to perform additional readiness check when your domain is online, select the **Perform online and offline readiness checks** option. In that case, you must specify the domain's host, port, user name, and password in the respective fields. For more information about these fields, click **Help**.

If you do not select this option your domain can be offline. Click **Next** to continue.

5. The Available Components screen is displayed if you have selected **Individually Selected Schemas** option.

This screen lists the available components for which the schemas will be selected. If you select something here, readiness check will be performed on that component's schema.

Select one or more components from the list to perform readiness check on them and click **Next**.

Note:

- Based on what you select on the Available Components screen, you will see additional screens. For example STB schema, Domain Directory, WLS Schema, MDS Schema (UCSUMS), etc.
 - If you select some component that has dependent components, then those components are automatically selected. For example, if you select Oracle Platform Security Services, then Oracle Audit Services is automatically selected.
 - In next screens, depending upon your selection on the Available Components screen, you must specify the domain directory (if you select Oracle Audit Services), or specify schema credentials to connect to the selected schema (if you select any other services, for example).
 - Select the Database Type, specify the Database Connect String, DBA User Name, and DBA Password. Then click **Connect**. For more information about entering values in these fields, click **Help**.
 - If the connection is successful, you can see the success message in the message bar at the bottom of the screen.
 - Select the **Schema User Name** from the drop-down list and specify the schema password in the **Schema Password** field. Click **Next**.
-
-

6. The Component List screen is displayed if you have selected **Domain Based** option.

The Component List screen is read-only and provides a list of components that are included in the domain you have selected and that are to be checked.

Review the list to make sure that all the components within your domain are listed and click **Next**.

7. The Readiness Summary screen provides a high-level overview of the readiness checks performed based on your selections.

To save your selections in a response file, click **Save Response File**.

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

Click **Next**.

8. The Readiness Check screen displays the current status of the readiness check. Depending on what you have selected to check, the process can take several minutes.

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

For a detailed text report, click **View Readiness Report**. This button is enabled only after all the readiness checks are complete.

Remember: If you are running the readiness check on your online production environment, Oracle recommends that you perform the check during off-peak hours to prevent performance degradation.

When done, click **Continue**.

9. The End of Readiness either displays **Readiness Success or **Readiness Failure**.**

If the readiness check is successful, you can now review the complete report by clicking **View Readiness Report**. If the readiness check encounters an issue or error, review the log file to identify the issues, correct the issues, and then restart the readiness check.

Oracle recommends that even with a successful completion of the readiness check, you should review the Readiness Report before you perform the actual upgrade.

A formatted Readiness Report is prepared for you after running the check. Make sure that you review the report and correct any issues before you start the actual upgrade. Use the **Find** option to search for a particular word within the report (such as a schema name or command, for example). The report also indicates where the completed Readiness Check Report file is located.

The **View Log** button is available on every screen. You can use it to see the latest logged information. The log file is managed by the command line options you set. For more information about the command line parameters, see [Upgrade Assistant Command Line Parameters](#).

4.4.4 Understanding the Readiness Report

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

The format of the readiness report file is:

`readiness_timestamp.txt`

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

A readiness report contains the following information:

Table 4-7 Readiness Report Elements

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

Table 4-7 (Cont.) Readiness Report Elements

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

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

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

```
This readiness check report was created on Tue May 30 11:15:52 EDT 2016
Log file is located at: ORACLE_HOME/oracle_common/upgrade/logs/
ua2016-05-30-11-14-06AM.log
Readiness Check Report File: ORACLE_HOME/oracle_common/upgrade/logs/
readiness2016-05-30-11-15-52AM.txt
```

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

```
Oracle Metadata Services
```

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

```
Schema User Name: DEV11_MDS
```

```
Database Type: Oracle Database
```

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


VERSION Schema DEV11_MDS is currently at version 12.1.1.1.0. Readiness checks will now be performed.

Starting schema test: TEST_REQUIRED_TABLES Test that the schema contains all the required tables

Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema contains all the required tables +++ PASS

Starting schema test: TEST_REQUIRED_PROCEDURES Test that the schema contains all the required stored procedures

EXCEPTION Schema is missing a required procedure: GETREPOSITORYFEATURES

Completed schema test: TEST_REQUIRED_PROCEDURES --> Test that the schema contains all the required stored procedures +++ FAIL

Starting schema test: TEST_REQUIRED_VIEWS Test that the schema contains all the required database views

Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema contains all the required database views +++ PASS

Starting index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes

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

Starting index test for table MDS_COMPONENTS: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes

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

Starting schema test: TEST_REQUIRED_TRIGGERS Test that the schema has all the required triggers

Completed schema test: TEST_REQUIRED_TRIGGERS --> Test that the schema has all the required triggers +++ PASS

Starting schema test: TEST_MISSING_COLUMNS Test that tables and views are not missing any required columns

Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and views are not missing any required columns +++ PASS

Starting schema test: TEST_UNEXPECTED_TABLES Test that the schema does not contain any unexpected tables

Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema does not contain any unexpected tables +++ PASS

Starting schema test: TEST_UNEXPECTED_PROCEDURES Test that the schema does not contain any unexpected stored procedures

Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the schema does not contain any unexpected stored procedures +++ PASS

Starting schema test: TEST_UNEXPECTED_VIEWS Test that the schema does not contain any unexpected views

Completed schema test: TEST_UNEXPECTED_VIEWS --> Test that the schema does not contain any unexpected views +++ PASS

Starting index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes

Completed index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes +++ PASS

Completed index test for table MDS_LABELS: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes +++ PASS

Starting index test for table MDS_LARGE_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes

Starting schema test: TEST_UNEXPECTED_TRIGGERS Test that the schema does not contain any unexpected triggers

Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema does not contain any unexpected triggers +++ PASS

Starting schema test: TEST_UNEXPECTED_COLUMNS Test that tables and views do not contain any unexpected columns

Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables and views do not contain any unexpected columns +++ PASS

Starting datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the proper datatypes

```
Completed datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
Starting datatype test for table MDS_COMPONENTS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
Starting permissions test: TEST_DBA_TABLE_GRANTS Test that DBA user has
privilege to view all user tables
Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA user has
privilege to view all user tables +++ PASS
Starting schema test: TEST_ENOUGH_TABLESPACE Test that the schema tablespaces
automatically extend if full
Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the schema
tablespaces automatically extend if full +++ PASS
Starting schema test: TEST_USER_TABLESPACE_QUOTA Test that tablespace quota for
this user is sufficient to perform the upgrade
Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that tablespace quota
for this user is sufficient to perform the upgrade +++ PASS
Starting schema test: TEST_ONLINE_TABLESPACE Test that schema tablespaces are
online
Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema tablespaces
are online +++ PASS
Starting schema test: TEST_DATABASE_VERSION Test that the database server
version number is supported for upgrade
INFO Database product version: Oracle Database 11g Enterprise Edition Release
11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Completed schema test: TEST_DATABASE_VERSION --> Test that the database server
version number is supported for upgrade +++ PASS
Finished readiness check of Oracle Metadata Services with status: FAILURE.
```

If you are running the 12.1.3.0 version of Oracle Fusion Middleware IAU Schemas that were upgraded from 11.1.1.7 and later or 12.1.2.0 releases, your readiness check may fail with the following error:

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

Note: You can ignore the “missing index” error in the Readiness Report. It is a known issue. The corresponding missing index is added during the schema upgrade operation. This error does not occur if the schema to be upgraded was created in 12c using the RCU.

4.5 Stopping Servers and Processes

Before running the Upgrade Assistant to upgrade your schemas and config, shut down all processes and servers, including the Administration server and any managed servers.

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

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

To stop your Fusion Middleware environment, follow the steps below.

Step 1: Stop System Components

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

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

You can stop system components in any order.

Step 2: Stop the Managed Servers

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

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

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

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

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

Step 4: Stop the Administration Server

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

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

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

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

Note: If external password storage is set up for the repository, then the server hosting the credential store should be up and running so that the work repository password can be retrieved during upgrade. For more information, see *Setting Up External Password Storage in Administering Oracle Data Integrator*.

Step 5: Stop Node Manager

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

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

4.6 Using the Upgrade Assistant to Upgrade Product Schemas

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

The Upgrade Assistant allows you to upgrade individually selected schemas or all schemas associated with a domain. The option you select determines which Upgrade Assistant screens you will use.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

Upgrading Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

4.6.1 Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends

that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

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

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

[Upgrade Assistant Command Line Parameters](#)

4.6.1.1 Upgrade Assistant Command Line Parameters

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

Table 4-8 Upgrade Assistant Command Line Parameters

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

Table 4-8 (Cont.) Upgrade Assistant Command Line Parameters

Parameter	Required or Optional	Description
<code>-examine</code>	Optional	<p>Performs the examine phase but does not perform an actual upgrade.</p> <p>Do not specify this parameter if you have specified the <code>-readiness</code> parameter.</p>
<code>-logLevel attribute</code>	Optional	<p>Sets the logging level, specifying one of the following attributes:</p> <ul style="list-style-type: none"> • TRACE • NOTIFICATION • WARNING • ERROR • INCIDENT_ERROR <p>The default logging level is NOTIFICATION.</p> <p>Consider setting the <code>-logLevel</code> attribute to TRACE so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if <code>-logLevel TRACE</code> is used.</p>
<code>-logDir location</code>	Optional	<p>Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant will create log files and temporary files.</p> <p>The default locations are:</p> <p>(UNIX) <code>ORACLE_HOME/oracle_common/upgrade/logs</code> <code>ORACLE_HOME/oracle_common/upgrade/temp</code></p> <p>(Windows) <code>ORACLE_HOME\oracle_common\upgrade\logs</code> <code>ORACLE_HOME\oracle_common\upgrade\temp</code></p>
<code>-help</code>	Optional	Displays all of the command line options.

4.6.2 Upgrading Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

Notes:

- If you are using external authentication, make sure that external authentication is changed to internal authentication.
 - **Edition-based redefinition (EBR) Users Only:** Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c. The new edition for 12c must be a child of your existing 11g or 12c edition. For more information on creating an edition on the server for redefinition, see *Creating an Edition on the Server for Editions-Based Redefinition in Planning an Upgrade of Oracle Fusion Middleware*.
-
-

To upgrade product schemas with the Upgrade Assistant:

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

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

2. On the Selected Schemas screen, select **Individually Selected Schemas**.

Caution: Upgrade only those schemas that are used to support your 12c (12.2.1) components. Do not upgrade schemas that are currently being used to support 11g or 12c components that are not included in Oracle Fusion Middleware 12c (12.2.1).

Click **Next**.

3. On the Available Components screen, select **Oracle Data Integrator** to upgrade the Master and Work Repository schema.
4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

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

5. On the ODI Schema screen, specify the database connection details for each schema you are upgrading:
 - Select the database type from the **Database Type** drop-down menu.
 - Enter the database connection details, and click **Connect**.

- Select the schema you want to upgrade from the **Schema User Name** drop-down menu, and then enter the password for the schema. Be sure to use the correct schema prefix for the schemas you are upgrading.
6. On the ODI Options screen, select all of the options.

Table 4-9 ODI Options

Option	Description
Replace Knowledge Modules with mandatory updates	This selection replaces standard Knowledge Modules with the newest version. Customizations to Oracle installed Knowledge Modules will be overwritten. But if you copy an installed Knowledge Module and customize the Knowledge Module, the customizations are not lost.
Upgrade topology and security metadata	This selection replaces topology and security artifacts such as Technologies, Datatypes, Security Profiles and others with the newest version. Customizations of installed objects will be overwritten. If the object is copied and customized, then the customizations are not lost. For more information on how to upgrade manually, see <i>Developing Integration Projects with Oracle Data Integrator</i> in <i>Developing Integration Projects with Oracle Data Integrator</i> .

For information on advanced upgrade options, see [Advanced Upgrade Options](#).

7. On the ODI Supervisor screen, enter the Supervisor account credentials for the ODI repository to be upgraded.

The installed Supervisor account is SUPERVISOR. Check with your ODI administrator for proper Supervisor account name and password, supplied when prompted by the Repository Creation Utility (RCU) when creating the Master and Work repositories for ODI.

The screenshot shows a window titled "Enter the Supervisor account credentials for the ODI repository to be upgraded." It contains two input fields: "ODI Supervisor User Name:" with the text "SUPERVISOR" entered, and "ODI Supervisor Password:" with a masked password represented by seven blue dots.

Note: When **All Schemas Used by a Domain** is selected, the Supervisor credentials for ODI are not pre-populated in the first instance as the domain does not contain them. If there are multiple ODI schemas, the Upgrade Assistant populates the user entry using the first set of credentials.

8. **For 11g to 12c upgrades only.** On the ODI Upgrade Key screen, use the auto-generated upgrade key to convert 11g IDs for repository objects into unique GUIDs, or specify your own key in the **Upgrade Key** field.

Recommendations:

- Edit the auto-generated key to provide a meaningful key that is easier to remember.
 - Note down the upgrade key so that the same upgrade key can be provided when the ODI objects are imported from the XML file.
-

ODI objects exist in ODI repositories and also in XML files exported from such repositories, which can be used, for example, in metadata exchanges between repositories. As such, there may be multiple copies of the same object, in different repositories and XML files.

In 12c, ODI uses GUIDs instead of internal numeric IDs for object identification. In order to make sure the object identity is preserved after upgrade, a deterministic algorithm is applied to calculate GUIDs from the internal IDs for existing objects (note that for new objects, ODI will generate random GUIDs).

Because the internal numeric IDs were not really universally unique, and were dependent on the repository ID to achieve a "pseudo-uniqueness," ODI allows the user to specify the upgrade key in order to reduce the likelihood of generating duplicated GUIDs. The upgrade key is fed into the GUID generation algorithm together with the internal numeric ID, to calculate the GUID.

Thus, choosing different upgrade keys protects from getting duplicated GUIDs for objects that accidentally have the same internal numeric IDs. However, when multiple copies of the same object exists (in a repository or exported in XML files), the same GUID should be produced for all copies of the object. For this reason, the same upgrade key must be used for all upgrade operations involving the copies of that particular object.

For example, suppose you have a project with 1001 as the ID in the 11g repository, and you also have a file exported from the same repository, which contains the same project (ID = 1001). In this case, the upgrade key used to upgrade the repository should be the same as the upgrade key used to import the XML file into the upgraded 12c repository. This ensures that the project object in the import file will be properly matched with the project object in the repository (when using one of `SYNONYM` import modes). However, if there is an 11g XML export file provided from a source containing objects created in another repository of which you have no information, there is a chance that it may contain a project that accidentally has the same internal ID (1001). In this case, to protect from erroneous object matching, which may corrupt the metadata, a different, custom upgrade key should be used when importing that file into the repository.

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

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

Note:

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

10. On the Upgrade Summary screen, review the summary of the options you have selected for schema upgrade.

Verify that the correct Source and Target Versions are listed for each schema you intend to upgrade.

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

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

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

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

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

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

Click **Next**.

12. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard.

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

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

4.6.3 Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

If you are using an Oracle database, connect to the database as a user having Oracle DBA privileges, and run the following from SQL*Plus to get the current version numbers:

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

In the query result:

- Check that the number in the `VERSION` column matches the latest version number for that schema. For example, verify that the schema version number is `12.2.1.0`. Note, however, that not all schema versions will be updated. Some schemas do not require an upgrade to this release and will retain their pre-upgrade version number.
- The `STATUS` field will be either `UPGRADING` or `UPGRADED` during the schema patching operation, and will become `VALID` when the operation is completed.
- If the status appears as `INVALID`, the schema update failed. You should examine the logs files to determine the reason for the failure.
- Synonym objects owned by `IAU_APPEND` and `IAU_VIEWER` will appear as `INVALID`, but that does not indicate a failure.

They become invalid because the target object changes after the creation of the synonym. The synonyms objects will become valid when they are accessed. You can safely ignore these `INVALID` objects.

4.7 Reconfiguring the Domain Using the Reconfiguration Wizard

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1).

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

- WebLogic Server core infrastructure
- Domain version

Note:

Before you begin the domain reconfiguration, note the following limitations:

- The Reconfiguration Wizard does not update any of your own applications that are included in the domain.
- Transforming a non-dynamic cluster domain to a dynamic cluster domain during the upgrade process is not supported.

The dynamic cluster feature is available when running the Reconfiguration Wizard, but Oracle only supports upgrading a non-dynamic cluster upgrade and then adding dynamic clusters. You cannot add dynamic cluster during the upgrade process.

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

- The domain version number in the `config.xml` file for the domain is updated to the Administration Server's installed WebLogic Server version.
- Reconfiguration templates for all installed Oracle products are automatically selected and applied to the domain. These templates define any reconfiguration tasks that are required to make the WebLogic domain compatible with the current WebLogic Server version.
- Start scripts are updated.

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

Note: Once the domain reconfiguration process starts, it is irreversible. Before running the Reconfiguration Wizard, ensure that you have backed up the domain as covered in the pre-upgrade checklist. If an error or other interruption occurs while running the Reconfiguration Wizard, you must restore the domain by copying the files and directories from the backup location to the original domain directory. This is the only way to ensure that the domain has been returned to its original state before reconfiguration.

Follow these instructions to reconfigure the existing domain using the Reconfiguration Wizard. For general information about how the domain is reconfigured, see *Reconfiguring WebLogic Domains in Upgrading Oracle WebLogic Server*.

[Backing Up the Domain](#)

[Starting the Reconfiguration Wizard](#)

[Reconfiguring the Domain with the Reconfiguration Wizard](#)

4.7.1 Backing Up the Domain

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

To create a backup of the domain directory:

1. Copy the source domain to a separate location to preserve the contents.

For example, copy `C:\domains\mydomain` to `C:\domains\mydomain_backup`.

2. Before updating the domain on each remote Managed Server, create a backup copy of the domain directory on each remote machine.
3. Verify that the backed up versions of the domain are complete.

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

4.7.2 Starting the Reconfiguration Wizard

To start the Reconfiguration Wizard in graphical mode:

1. Sign in to the system on which the domain resides.
2. Open the command shell (on UNIX operating systems) or open a command prompt window (on Windows operating systems).
3. **Edition Based Database Users Only:** If your schemas are configured with EBR database, a default edition name must be manually supplied before you run the Reconfiguration Wizard.

Run the following SQL command to set the default edition:

```
ALTER DATABASE DEFAULT EDITION = edition_name;
```

where *edition_name* is the child edition name.

4. Go to the `oracle_common/common/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/common/bin`
 - (Windows) `ORACLE_HOME\oracle_common\commom\bin`
5. Start the Reconfiguration Wizard with the following logging options:
 - (UNIX) `./reconfig.sh -log=log_file -log_priority=ALL`
 - (Windows) `reconfig.cmd -log=log_file -log_priority=ALL`

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

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

Note:

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

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

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

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

4.7.3 Reconfiguring the Domain with the Reconfiguration Wizard

Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain before running the Upgrade Assistant

Note: If the source is a clustered environment, run the Reconfiguration Wizard on the primary node only. Use the `pack` and `unpack` utilities to apply the changes to other cluster members in the domain.

To reconfigure the domain with the Reconfiguration Wizard:

1. On the Select Domain screen, specify the location of the domain you want to upgrade or click **Browse** to navigate and select the domain directory. Click **Next**.
2. On the Reconfiguration Setup Progress screen, view the progress of the setup process. When complete, click **Next**.

During this process:

- The reconfiguration templates for your installed products, including Fusion Middleware products, are automatically applied. This updates various domain configuration files such as `config.xml`, `config-groups.xml`, and `security.xml` (among others).
 - Schemas, scripts, and other such files that support your Fusion Middleware products are updated.
 - The domain upgrade is validated.
3. On the Domain Mode and JDK screen, select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use. The supported JDK version for 12c (12.2.1) is 1.8.0_60 and later. Click **Next**.

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

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

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

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

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

NOTE: For any existing 11g datasource, the reconfiguration will preserve the existing values. For new datasources where the schema was created for 12c by the RCU, the default connection data will be retrieved from the `_STB` schema. If no connection data for a given schema is found in the `_STB` schema, then the default connection data is used.

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

5. On the JDBC Component Schema screen, verify that the DBMS/Service and the Host name is correct for each component schema and click **Next**.
6. On the JDBC Component Schema Test screen, select all the component schemas and click **Test Selected Connections** to test the connection for each schema. The result of the test is indicated in the **Status** column.

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

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

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

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

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

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

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

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

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

During this process:

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

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

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

was reconfigured as well as the Administration Server URL (including the listen port). If the reconfiguration is successful, it displays **Oracle WebLogic Server Reconfiguration Succeeded**.

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

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

4.8 Upgrading Domain Component Configurations

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

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

Upgrading Domain Component Configurations Using the Upgrade Assistant

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

4.8.1 Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

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

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

[Upgrade Assistant Command Line Parameters](#)

4.8.1.1 Upgrade Assistant Command Line Parameters

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

Table 4-10 Upgrade Assistant Command Line Parameters

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

Table 4-10 (Cont.) Upgrade Assistant Command Line Parameters

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

4.8.2 Upgrading Domain Component Configurations Using the Upgrade Assistant

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

After running the Reconfiguration Wizard to reconfigure the WebLogic domain to 12c (12.2.1), you must run the Upgrade Assistant to upgrade the domain *component* configurations to match the updated domain configuration.

To upgrade domain component configurations with the Upgrade Assistant:

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

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

2. On the next screen:
 - Select **All Configurations Used By a Domain**. The screen name changes to WebLogic Components.
 - In the **Domain Directory** field, enter the WebLogic domain directory path.

Click **Next**.

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

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

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

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

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

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

Note:

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

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

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

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

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

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

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

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

Click **Next**.

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

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

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

4.9 Starting Servers and Processes

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

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

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

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

Step 1: Start the Administration Server

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

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

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

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

Step 2: Start Node Manager

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

- (UNIX) `DOMAIN_HOME/bin/startNodeManager.sh`

- (Windows) `DOMAIN_HOME\bin\startNodeManager.cmd`

Step 3: Start Oracle Identity Management Components

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

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

Step 4: Start the Managed Servers

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

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

When prompted, enter your user name and password.

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

Step 5: Start System Components

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

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

You can start system components in any order.

4.10 Verifying the Upgrade

After completing all the upgrade steps, verify that the upgrade was successful.

To verify the upgrade:

- Start the Node Manager and standalone collocated agent.

For more information, see Starting a Standalone Collocated Agent with Node Manager in *Installing and Configuring Oracle Data Integrator*.

Upgrading an Oracle Data Integrator Java EE Agent Environment from 11g

You can upgrade an Oracle Data Integrator Java EE agent environment from Oracle Fusion Middleware 11g to 12c (12.2.1).

Follow the steps in the following topics to perform this upgrade:

[About the Oracle Data Integrator Java EE Agent Upgrade Process](#)

Review the process flowchart for an overview of the upgrade process for an Oracle Data Integrator Java EE agent.

[Installing the Oracle Data Integrator Java EE Environment](#)

Before beginning your upgrade, download Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12c (12.2.1) distributions on the target system and install them using Oracle Universal Installer.

[Creating the Required 12c Schemas with the RCU](#)

When upgrading from 11g, you must use the Repository Creation Utility (RCU) to create the required 12c schemas before you begin the upgrade.

[Running a Pre-Upgrade Readiness Check](#)

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

[Stopping Servers and Processes](#)

Before running the Upgrade Assistant to upgrade your schemas and config, shut down all processes and servers, including the Administration server and any managed servers.

[Using the Upgrade Assistant to Upgrade Product Schemas](#)

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

[Reconfiguring the Domain Using the Reconfiguration Wizard](#)

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1).

[Upgrading Domain Component Configurations](#)

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

Starting Servers and Processes

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

Verifying the Upgrade

After completing all the upgrade steps, verify that the upgrade was successful.

Running a Pre-Upgrade Readiness Check

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

5.1 About the Oracle Data Integrator Java EE Agent Upgrade Process

Review the process flowchart for an overview of the upgrade process for an Oracle Data Integrator Java EE agent.

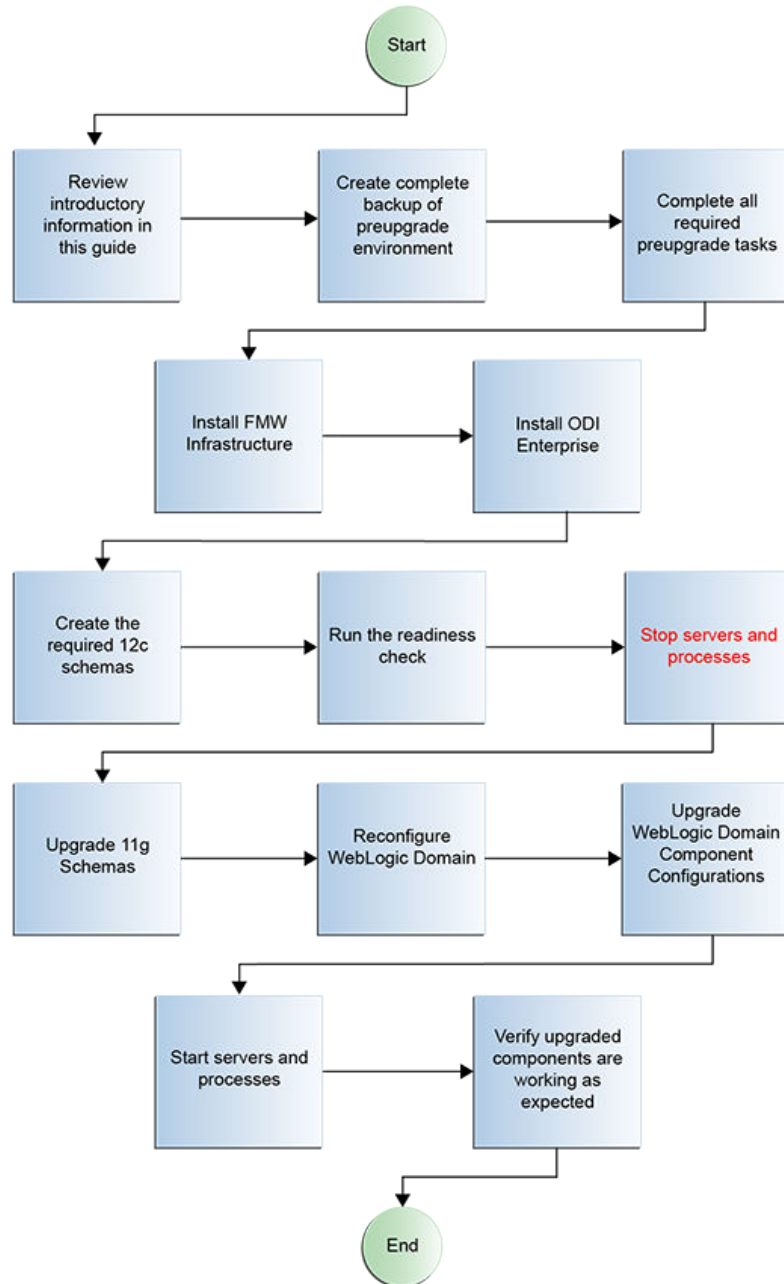


Table 5-1 Tasks for Upgrading Oracle Data Integrator

Task	Description
Required If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.	See: <ul style="list-style-type: none"> • Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1) • Pre-Upgrade Requirements

Table 5-1 (Cont.) Tasks for Upgrading Oracle Data Integrator

Task	Description
Required Install Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12c (12.2.1) in the new Oracle home.	See Installing the Oracle Data Integrator Java EE Environment .
Required Start the Repository Creation Utility (RCU) to create the required 12c database schemas.	Create the STB, OPSS, IAU, IAU_VIEWER, and IAU_APPEND schemas. See #unique_122 .
Optional Run a pre-upgrade readiness check.	See #unique_123 .
Required Shut down the 11g environment (stop all Administration and Managed Servers).	WARNING: Failure to shut down your servers during an upgrade may lead to data corruption. See #unique_124 .
Required Start the Upgrade Assistant to upgrade the 11g database schemas and to migrate all active (in flight) instance data.	See #unique_125 . NOTE: The upgrade of active instance data is started automatically when running the Upgrade Assistant. Once the data is successfully upgraded to the new 12c (12.2.1) environment, you can close the Upgrade Assistant. The closed instances will continue to upgrade through a background process.
Required Start the Reconfiguration Wizard to reconfigure the domain.	Run the Configuration Wizard to update the existing domain to use the newly installed software. See #unique_126 .
Required if JRF component upgrade needed Start the Upgrade Assistant (again) to upgrade domain component configurations.	Run the Upgrade Assistant to update the reconfigured domain's component configurations. See #unique_127 .
Required Restart the servers and the 12c (12.2.1) instance.	When the upgrade process is complete, restart the 12c (12.2.1) instance. See #unique_128 .
Required Verify the upgrade.	Ensure all of the upgraded components are working as expected before deleting your backups. See #unique_129 .

5.2 Installing the Oracle Data Integrator Java EE Environment

Before beginning your upgrade, download Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12c (12.2.1) distributions on the target system and install them using Oracle Universal Installer.

Note: You must install the Oracle Fusion Middleware Infrastructure distribution first before installing other Fusion Middleware products, when Infrastructure is required for the upgrade.

To install the 12c (12.2.1) distributions:

1. Sign in to the target system where you want to install the 12c (12.2.1) product distribution.
2. Download the following 12c (12.2.1) product distributions from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle Fusion Middleware Infrastructure (fmw_12.2.1.0_infrastructure_generic.jar)
 - Oracle Data Integrator (fmw_12.2.1.0_odi_Disk1_1of2.zip and fmw_12.2.1.0_odi_Disk1_2of2.zip)
3. Change to the directory where you downloaded the 12c (12.2.1) product distribution.
4. Unzip the installer fmw_12.2.1.0_odi_Disk1_1of2.zip and fmw_12.2.1.0_odi_Disk1_2of2.zip files.
5. Start the installation program for Oracle Fusion Middleware Infrastructure:
 - (UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.0_infrastructure_generic.jar`
 - (Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.0_infrastructure_generic.jar`
6. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an Oracle product on this host.

Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location and click **Next**.

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

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

Click **Next**.

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

For more information about Oracle Fusion Middleware directory structure, see *Selecting Directories for Installation and Configuration* in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

10. On the Installation Type screen, select the following:

- For Infrastructure, select **Fusion Middleware Infrastructure**
- For Oracle Data Integrator, select **Enterprise Installation**

Click **Next**.

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

To view the list of tasks that 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).

12. On the Installation Summary screen, verify the installation options you selected.

If you want to save these options to a response file, click **Save Response File** and enter the response file location and name. The response file collects and stores all the information that you have entered, and enables you to perform a silent installation (from the command line) at a later time.

Click **Install** to begin the installation.

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

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

15. After you have installed Oracle Fusion Middleware Infrastructure, enter the following command to start the installer for your product distribution and repeat the steps above to navigate through the installer screens:

(UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.0_odi.jar`

(Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.0_odi.jar`

5.3 Creating the Required 12c Schemas with the RCU

When upgrading from 11g, you must use the Repository Creation Utility (RCU) to create the required 12c schemas before you begin the upgrade.

For the ODI environments described in this guide, the required schemas are:

- ODI standalone agent (no WebLogic domain): `STB`
- ODI standalone collocated agent (with WebLogic domain): `STB`, `OPSS`, `IAU`, `IAU_VIEWER`, and `IAU_APPEND`
- ODI Java EE agent: `STB`, `OPSS`, `IAU`, `IAU_VIEWER`, and `IAU_APPEND`

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

If you are upgrading from 11g, refer to the Pre-Upgrade Checklist to identify the existing schemas in your domain. The following schemas must exist before you upgrade to 12c:

- **Service Table** schema (*prefix_STB*). This schema is new in 12c and is required for domain-based upgrades. It stores basic schema configuration information (for example, schema prefixes and passwords) that can be accessed and used by other Oracle Fusion Middleware components during the domain creation. This schema is automatically created when you run the Repository Creation Utility (RCU), where you specify the existing schema owner prefix that you used for your other 11g schemas. **Note:** If the Service Table schema does not exist, you may encounter the error message UPGAST-00328 : The schema version registry table does not exist on this database. If that happens it is necessary to create the service table schema in order to run Upgrade Assistant.
- **Oracle Platform Security Services (OPSS)** schema (*prefix_OPSS*). This schema is required if you are using an OID-based security store in 11g. This schema is automatically created when you run the Repository Creation Utility (RCU). The only supported LDAP-based OPSS security store is Oracle Internet Directory (OID). An LDAP-based policy store is typically used in production environments. You do not need to reassociate an OID-based security store before upgrade. While the Upgrade Assistant is running, you can select the OPSS schema. The Upgrade Assistant upgrades the OID-based security store automatically. **Note:** The 12c OPSS database schema is required so that you can reference the 12c schema during the reconfiguration of the domain. Your domain continues to use the OID-based security store after the upgrade is complete.
- **Audit** schemas. When upgrading audit services (*_IAU*), make sure that you select *_IAU_VIEWER* and *_IAU_APPEND* in addition to *_IAU*. The Upgrade Assistant will handle their creation for you automatically when they are selected.

To create the 12c schemas with the RCU:

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

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

2. If you have not already done so, set the *JAVA_HOME* environment variable and add *\$JAVA_HOME/bin* to *\$PATH*. The supported JDK version for 12c (12.2.1) is 1.8.0_60.

3. Go to the `oracle_common/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/bin`
 - (Windows) `ORACLE_HOME\oracle_common\bin`
4. Start the RCU:
 - (UNIX) `./rcu`
 - (Windows) `rcu.bat`
5. On the Welcome screen, click **Next**.
6. On the Create Repository screen, select **Create Repository** and then select **System Load and Product Load**.

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

Click **Next**.

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

Table 5-2 Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition

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

Table 5-3 Connection Credentials for MySQL Databases

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

Table 5-4 Connection Credentials for Microsoft SQL Server Databases

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

Table 5-5 Connection Credentials for IBM DB2 Databases

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

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

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

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

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

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

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

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

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

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

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

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

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

5.4 Running a Pre-Upgrade Readiness Check

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

About Running a Pre-Upgrade Readiness Check

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

Starting the Upgrade Assistant in Readiness Mode

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

Navigating the Upgrade Assistant Screens (Readiness Mode)

You must navigate all the Upgrade Assistant screens that appear, based on your primary selection, to complete the Readiness Check. In addition, if you want to save the response file for performing readiness check on other domains/system, you can do that on the Readiness Summary screen.

Understanding the Readiness Report

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

5.4.1 About Running a Pre-Upgrade Readiness Check

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

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

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

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

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

Note:

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

5.4.2 Starting the Upgrade Assistant in Readiness Mode

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

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

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

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

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

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

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

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

[Upgrade Assistant Command Line Parameters](#)

5.4.2.1 Upgrade Assistant Command Line Parameters

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

Table 5-6 Upgrade Assistant Command Line Parameters

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

Table 5-6 (Cont.) Upgrade Assistant Command Line Parameters

Parameter	Required or Optional	Description
-examine	Optional	Performs the examine phase but does not perform an actual upgrade. Do not specify this parameter if you have specified the -readiness parameter.
-logLevel <i>attribute</i>	Optional	Sets the logging level, specifying one of the following attributes: <ul style="list-style-type: none"> • TRACE • NOTIFICATION • WARNING • ERROR • INCIDENT_ERROR The default logging level is NOTIFICATION. Consider setting the -logLevel attribute to TRACE so that more information is logged. This is useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if -logLevel TRACE is used.
-logDir <i>location</i>	Optional	Sets the default location of upgrade log files and temporary files. You must specify an existing, writable directory where the Upgrade Assistant will create log files and temporary files. The default locations are: (UNIX) <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs <i>ORACLE_HOME</i> / oracle_common/upgrade/ temp (Windows) <i>ORACLE_HOME</i> \oracle_common\upgrade \logs <i>ORACLE_HOME</i> \oracle_common\upgrade \temp
-help	Optional	Displays all of the command line options.

5.4.3 Navigating the Upgrade Assistant Screens (Readiness Mode)

You must navigate all the Upgrade Assistant screens that appear, based on your primary selection, to complete the Readiness Check. In addition, if you want to save the response file for performing readiness check on other domains/system, you can do that on the Readiness Summary screen.

To complete the Readiness Check:

1. The Welcome screen provides an overview of the readiness check. Review the information on this screen and click **Next**.
2. You can perform readiness check on schemas or domain configurations that are at a supported starting point.

On the Readiness Check Type screen, select one of the following options depending upon the type of readiness check you want to perform:

- **Individually Selected Schemas**
 - **Domain Based**
3. To review and perform readiness check on specific schemas, select **Individually Selected Schemas**.

The **Individually Selected Schemas** option allows you to select the schemas you want to review before the upgrade. When you perform the readiness check on the schemas, the readiness check report tells you whether a schema is supported for an upgrade, or where an upgrade is needed. Click **Next**.

Note: When you select the **Individually Selected Schemas** option, the screen name changes from Readiness Check Type to Selected Schema.

4. To perform a readiness check per domain, select **Domain Based**.

The **Domain Based** option allows you to check all of the upgrade-eligible schemas and/or component configurations used by the domain. The Upgrade Assistant detects all of the schemas for you. You can check schemas and component configurations at the same time. Or, if you prefer, you can select to review one thing at a time by selecting one of the following options:

- **Include checks for all schemas**

Select this option to enable the Upgrade Assistant to discover and review all components that have a schema available to upgrade.

- **Include checks for all configurations**

Select this option to review component configurations for a managed WebLogic Server domain.

Note: When you select the **Domain Based** option, the screen name changes from Readiness Check Type to Schemas and Configuration.

Specify the **Domain Directory** by entering the path to your domain or by clicking **Browse** to use the navigation tree.

If you want to perform additional readiness check when your domain is online, select the **Perform online and offline readiness checks** option. In that case, you must specify the domain's host, port, user name, and password in the respective fields. For more information about these fields, click **Help**.

If you do not select this option your domain can be offline. Click **Next** to continue.

5. The Available Components screen is displayed if you have selected **Individually Selected Schemas** option.

This screen lists the available components for which the schemas will be selected. If you select something here, readiness check will be performed on that component's schema.

Select one or more components from the list to perform readiness check on them and click **Next**.

Note:

- Based on what you select on the Available Components screen, you will see additional screens. For example STB schema, Domain Directory, WLS Schema, MDS Schema (UCSUMS), etc.
- If you select some component that has dependent components, then those components are automatically selected. For example, if you select Oracle Platform Security Services, then Oracle Audit Services is automatically selected.
- In next screens, depending upon your selection on the Available Components screen, you must specify the domain directory (if you select Oracle Audit Services), or specify schema credentials to connect to the selected schema (if you select any other services, for example).
 - Select the Database Type, specify the Database Connect String, DBA User Name, and DBA Password. Then click **Connect**. For more information about entering values in these fields, click **Help**.
 - If the connection is successful, you can see the success message in the message bar at the bottom of the screen.
 - Select the **Schema User Name** from the drop-down list and specify the schema password in the **Schema Password** field. Click **Next**.

6. The Component List screen is displayed if you have selected **Domain Based** option.

The Component List screen is read-only and provides a list of components that are included in the domain you have selected and that are to be checked.

Review the list to make sure that all the components within your domain are listed and click **Next**.

7. The Readiness Summary screen provides a high-level overview of the readiness checks performed based on your selections.

To save your selections in a response file, click **Save Response File**.

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

Click **Next**.

8. The Readiness Check screen displays the current status of the readiness check. Depending on what you have selected to check, the process can take several minutes.

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

For a detailed text report, click **View Readiness Report**. This button is enabled only after all the readiness checks are complete.

Remember: If you are running the readiness check on your online production environment, Oracle recommends that you perform the check during off-peak hours to prevent performance degradation.

When done, click **Continue**.

9. The End of Readiness either displays **Readiness Success** or **Readiness Failure**.

If the readiness check is successful, you can now review the complete report by clicking **View Readiness Report**. If the readiness check encounters an issue or error, review the log file to identify the issues, correct the issues, and then restart the readiness check.

Oracle recommends that even with a successful completion of the readiness check, you should review the Readiness Report before you perform the actual upgrade.

A formatted Readiness Report is prepared for you after running the check. Make sure that you review the report and correct any issues before you start the actual upgrade. Use the **Find** option to search for a particular word within the report (such as a schema name or command, for example). The report also indicates where the completed Readiness Check Report file is located.

The **View Log** button is available on every screen. You can use it to see the latest logged information. The log file is managed by the command line options you set. For more information about the command line parameters, see [Upgrade Assistant Command Line Parameters](#).

5.4.4 Understanding the Readiness Report

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

The format of the readiness report file is:

```
readiness_timestamp.txt
```

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

A readiness report contains the following information:

Table 5-7 Readiness Report Elements

Report Information	Description	Required Action
Overall Readiness Status: SUCCESS or FAILURE	The top of the report indicates whether the Upgrade readiness check passed or completed with one or more errors.	If the report completed with one or more errors, search for FAIL and correct the failing issues before attempting to upgrade. You can re-run the readiness check as many times as necessary before an upgrade.
Timestamp	This is the date and time that the report was generated.	No action required.
Log file location <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs	This is the directory location of the generated log file.	No action required.
Readiness Report location <i>ORACLE_HOME</i> / oracle_common/upgrade/ logs	This is the directory location of the generated readiness report.	No action required.
Names of components that were checked	The names and versions of the components included in the check and status.	If your domain includes components that cannot be upgraded to this release, such as SOA Core Extension, then do not attempt an upgrade.
Names of schemas that were checked	The names and current versions of the schemas included in the check and status.	Review the version numbers of your schemas. If your domain includes schemas that cannot be upgraded to this release, then do not attempt an upgrade.
Individual Object Test Status: FAIL	The readiness check test detected an issue with a specific object.	Do not upgrade until all FAILED issues have been resolved.
Individual Object Test Status: PASS	The readiness check test detected no issues for the specific object.	If your readiness check report shows only the PASS status, then you can upgrade your environment. Note, however, that the Readiness Check cannot detect issues with externals such as hardware or connectivity during an upgrade. You should always monitor the progress of your upgrade.
Completed Readiness Check of <Object> Status: FAILURE	The readiness check detected one or more errors that must be resolved for a particular object such as a schema, an index or datatype.	Do not upgrade until all FAILED issues have been resolved.
Completed Readiness Check of <Object> Status: SUCCESS	The readiness check test detected no issues.	No action required.

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

Upgrade readiness check completed with one or more errors.

This readiness check report was created on Tue May 30 11:15:52 EDT 2016
Log file is located at: *ORACLE_HOME*/oracle_common/upgrade/logs/
ua2016-05-30-11-14-06AM.log
Readiness Check Report File: *ORACLE_HOME*/oracle_common/upgrade/logs/
readiness2016-05-30-11-15-52AM.txt

Starting readiness check of components.

Oracle Metadata Services

Starting readiness check of Oracle Metadata Services.

Schema User Name: DEV11_MDS

Database Type: Oracle Database

Database Connect String: machinename@yourcompany.com

VERSION Schema DEV11_MDS is currently at version 12.1.1.1.0. Readiness checks will now be performed.

Starting schema test: TEST_REQUIRED_TABLES Test that the schema contains all the required tables

Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema contains all the required tables +++ PASS

Starting schema test: TEST_REQUIRED_PROCEDURES Test that the schema contains all the required stored procedures

EXCEPTION Schema is missing a required procedure: GETREPOSITORYFEATURES
Completed schema test: TEST_REQUIRED_PROCEDURES --> Test that the schema contains all the required stored procedures +++ FAIL

Starting schema test: TEST_REQUIRED_VIEWS Test that the schema contains all the required database views

Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema contains all the required database views +++ PASS

Starting index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes

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

Starting index test for table MDS_COMPONENTS: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes

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

Starting schema test: TEST_REQUIRED_TRIGGERS Test that the schema has all the required triggers

Completed schema test: TEST_REQUIRED_TRIGGERS --> Test that the schema has all the required triggers +++ PASS

Starting schema test: TEST_MISSING_COLUMNS Test that tables and views are not missing any required columns

Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and views are not missing any required columns +++ PASS

Starting schema test: TEST_UNEXPECTED_TABLES Test that the schema does not contain any unexpected tables

Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema does not contain any unexpected tables +++ PASS

Starting schema test: TEST_UNEXPECTED_PROCEDURES Test that the schema does not contain any unexpected stored procedures

Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the schema does not contain any unexpected stored procedures +++ PASS

Starting schema test: TEST_UNEXPECTED_VIEWS Test that the schema does not contain any unexpected views

Completed schema test: TEST_UNEXPECTED_VIEWS --> Test that the schema does not contain any unexpected views +++ PASS

Starting index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes

Completed index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test


```

that the table does not contain any unexpected indexes +++ PASS
  Completed index test for table MDS_LABELS: TEST_UNEXPECTED_INDEXES --> Test that
the table does not contain any unexpected indexes +++ PASS
  Starting index test for table MDS_LARGE_ATTRIBUTES: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes
  Starting schema test: TEST_UNEXPECTED_TRIGGERS Test that the schema does not
contain any unexpected triggers
  Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema does not
contain any unexpected triggers +++ PASS
  Starting schema test: TEST_UNEXPECTED_COLUMNS Test that tables and views do not
contain any unexpected columns
  Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables and views do
not contain any unexpected columns +++ PASS
  Starting datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
  Completed datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
  Starting datatype test for table MDS_COMPONENTS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
  Starting permissions test: TEST_DBA_TABLE_GRANTS Test that DBA user has
privilege to view all user tables
  Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA user has
privilege to view all user tables +++ PASS
  Starting schema test: TEST_ENOUGH_TABLESPACE Test that the schema tablespaces
automatically extend if full
  Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the schema
tablespaces automatically extend if full +++ PASS
  Starting schema test: TEST_USER_TABLESPACE_QUOTA Test that tablespace quota for
this user is sufficient to perform the upgrade
  Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that tablespace quota
for this user is sufficient to perform the upgrade +++ PASS
  Starting schema test: TEST_ONLINE_TABLESPACE Test that schema tablespaces are
online
  Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema tablespaces
are online +++ PASS
  Starting schema test: TEST_DATABASE_VERSION Test that the database server
version number is supported for upgrade
  INFO Database product version: Oracle Database 11g Enterprise Edition Release
11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
  Completed schema test: TEST_DATABASE_VERSION --> Test that the database server
version number is supported for upgrade +++ PASS
  Finished readiness check of Oracle Metadata Services with status: FAILURE.

```

If you are running the 12.1.3.0 version of Oracle Fusion Middleware IAU Schemas that were upgraded from 11.1.1.7 and later or 12.1.2.0 releases, your readiness check may fail with the following error:

```

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

```

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

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

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

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

Note: You can ignore the “missing index” error in the Readiness Report. It is a known issue. The corresponding missing index is added during the schema upgrade operation. This error does not occur if the schema to be upgraded was created in 12c using the RCU.

5.5 Stopping Servers and Processes

Before running the Upgrade Assistant to upgrade your schemas and config, shut down all processes and servers, including the Administration server and any managed servers.

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

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

To stop your Fusion Middleware environment, follow the steps below.

Step 1: Stop System Components

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

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

You can stop system components in any order.

Step 2: Stop the Managed Servers

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

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

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

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

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

Step 4: Stop the Administration Server

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

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

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

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

Note: If external password storage is set up for the repository, then the server hosting the credential store should be up and running so that the work repository password can be retrieved during upgrade. For more information, see *Setting Up External Password Storage in Administering Oracle Data Integrator*.

Step 5: Stop Node Manager

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

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

5.6 Using the Upgrade Assistant to Upgrade Product Schemas

After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

The Upgrade Assistant allows you to upgrade individually selected schemas or all schemas associated with a domain. The option you select determines which Upgrade Assistant screens you will use.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

Upgrading Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

5.6.1 Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

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

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

Upgrade Assistant Command Line Parameters

5.6.1.1 Upgrade Assistant Command Line Parameters

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

Table 5-8 Upgrade Assistant Command Line Parameters

Parameter	Required or Optional	Description
<code>-readiness</code>	Required for readiness checks NOTE: Readiness checks cannot be performed on standalone installations (those not managed by the WebLogic Server)	Performs the upgrade readiness check without performing an actual upgrade. Schemas and configurations are checked. Do not use this parameter if you have specified the <code>-examine</code> parameter.
<code>-threads</code>	Optional	Identifies the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. The default is 4.

Table 5-8 (Cont.) Upgrade Assistant Command Line Parameters

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

Table 5-8 (Cont.) Upgrade Assistant Command Line Parameters

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

5.6.2 Upgrading Product Schemas Using the Upgrade Assistant

Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

Notes:

- If you are using external authentication, make sure that external authentication is changed to internal authentication.
 - **Edition-based redefinition (EBR) Users Only:** Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c. The new edition for 12c must be a child of your existing 11g or 12c edition. For more information on creating an edition on the server for redefinition, see *Creating an Edition on the Server for Editions-Based Redefinition* in *Planning an Upgrade of Oracle Fusion Middleware*.
-
-

To upgrade product schemas with the Upgrade Assistant:

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

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

2. On the Selected Schemas screen, select **Individually Selected Schemas**.

Caution: Upgrade only those schemas that are used to support your 12c (12.2.1) components. Do not upgrade schemas that are currently being used to support 11g or 12c components that are not included in Oracle Fusion Middleware 12c (12.2.1).

Click **Next**.

3. On the Available Components screen, select **Oracle Data Integrator** to upgrade the Master and Work Repository schema.
4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

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

5. On the ODI Schema screen, specify the database connection details for each schema you are upgrading:
 - Select the database type from the **Database Type** drop-down menu.
 - Enter the database connection details, and click **Connect**.
 - Select the schema you want to upgrade from the **Schema User Name** drop-down menu, and then enter the password for the schema. Be sure to use the correct schema prefix for the schemas you are upgrading.
6. On the ODI Options screen, select all of the options.

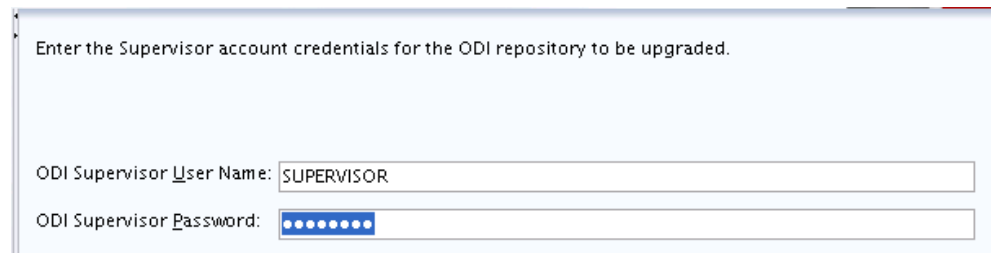
Table 5-9 ODI Options

Option	Description
Replace Knowledge Modules with mandatory updates	This selection replaces standard Knowledge Modules with the newest version. Customizations to Oracle installed Knowledge Modules will be overwritten. But if you copy an installed Knowledge Module and customize the Knowledge Module, the customizations are not lost.
Upgrade topology and security metadata	This selection replaces topology and security artifacts such as Technologies, Datatypes, Security Profiles and others with the newest version. Customizations of installed objects will be overwritten. If the object is copied and customized, then the customizations are not lost. For more information on how to upgrade manually, see <i>Developing Integration Projects with Oracle Data Integrator</i> in <i>Developing Integration Projects with Oracle Data Integrator</i> .

For information on advanced upgrade options, see [Advanced Upgrade Options](#).

7. On the ODI Supervisor screen, enter the Supervisor account credentials for the ODI repository to be upgraded.

The installed Supervisor account is `SUPERVISOR`. Check with your ODI administrator for proper Supervisor account name and password, supplied when prompted by the Repository Creation Utility (RCU) when creating the Master and Work repositories for ODI.



Enter the Supervisor account credentials for the ODI repository to be upgraded.

ODI Supervisor User Name: SUPERVISOR

ODI Supervisor Password: ●●●●●●●

Note: When **All Schemas Used by a Domain** is selected, the Supervisor credentials for ODI are not pre-populated in the first instance as the domain does not contain them. If there are multiple ODI schemas, the Upgrade Assistant populates the user entry using the first set of credentials.

- 8. For 11g to 12c upgrades only.** On the ODI Upgrade Key screen, use the auto-generated upgrade key to convert 11g IDs for repository objects into unique GUIDs, or specify your own key in the **Upgrade Key** field.

Recommendations:

- Edit the auto-generated key to provide a meaningful key that is easier to remember.
 - Note down the upgrade key so that the same upgrade key can be provided when the ODI objects are imported from the XML file.
-

ODI objects exist in ODI repositories and also in XML files exported from such repositories, which can be used, for example, in metadata exchanges between repositories. As such, there may be multiple copies of the same object, in different repositories and XML files.

In 12c, ODI uses GUIDs instead of internal numeric IDs for object identification. In order to make sure the object identity is preserved after upgrade, a deterministic algorithm is applied to calculate GUIDs from the internal IDs for existing objects (note that for new objects, ODI will generate random GUIDs).

Because the internal numeric IDs were not really universally unique, and were dependent on the repository ID to achieve a "pseudo-uniqueness," ODI allows the user to specify the upgrade key in order to reduce the likelihood of generating duplicated GUIDs. The upgrade key is fed into the GUID generation algorithm together with the internal numeric ID, to calculate the GUID.

Thus, choosing different upgrade keys protects from getting duplicated GUIDs for objects that accidentally have the same internal numeric IDs. However, when multiple copies of the same object exists (in a repository or exported in XML files), the same GUID should be produced for all copies of the object. For this reason, the same upgrade key must be used for all upgrade operations involving the copies of that particular object.

For example, suppose you have a project with 1001 as the ID in the 11g repository, and you also have a file exported from the same repository, which contains the same project (ID = 1001). In this case, the upgrade key used to upgrade the repository should be the same as the upgrade key used to import the XML file into the upgraded 12c repository. This ensures that the project object in the import file will be properly matched with the project object in the repository (when using one of SYNONYM import modes). However, if there is an 11g XML export file provided from a source containing objects created in another repository of which you have no information, there is a chance that it may contain a project that accidentally has the same internal ID (1001). In this case, to protect from erroneous object matching, which may corrupt the metadata, a different, custom upgrade key should be used when importing that file into the repository.

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

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

Note:

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

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

Verify that the correct Source and Target Versions are listed for each schema you intend to upgrade.

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

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

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

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

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

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

Click **Next**.

12. If the upgrade is successful: On the Upgrade Success screen, click **Close** to complete the upgrade and close the wizard.

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

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

5.6.3 Verifying the Schema Upgrade

After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

If you are using an Oracle database, connect to the database as a user having Oracle DBA privileges, and run the following from SQL*Plus to get the current version numbers:

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

In the query result:

- Check that the number in the `VERSION` column matches the latest version number for that schema. For example, verify that the schema version number is 12.2.1.0. Note, however, that not all schema versions will be updated. Some schemas do not require an upgrade to this release and will retain their pre-upgrade version number.
- The `STATUS` field will be either `UPGRADING` or `UPGRADED` during the schema patching operation, and will become `VALID` when the operation is completed.
- If the status appears as `INVALID`, the schema update failed. You should examine the logs files to determine the reason for the failure.
- Synonym objects owned by `IAU_APPEND` and `IAU_VIEWER` will appear as `INVALID`, but that does not indicate a failure.

They become invalid because the target object changes after the creation of the synonym. The synonyms objects will become valid when they are accessed. You can safely ignore these `INVALID` objects.

5.7 Reconfiguring the Domain Using the Reconfiguration Wizard

Run the Reconfiguration Wizard to reconfigure your domain component configurations to 12c (12.2.1).

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

- WebLogic Server core infrastructure
- Domain version

Note:

Before you begin the domain reconfiguration, note the following limitations:

- The Reconfiguration Wizard does not update any of your own applications that are included in the domain.
- Transforming a non-dynamic cluster domain to a dynamic cluster domain during the upgrade process is not supported.

The dynamic cluster feature is available when running the Reconfiguration Wizard, but Oracle only supports upgrading a non-dynamic cluster upgrade and then adding dynamic clusters. You cannot add dynamic cluster during the upgrade process.

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

- The domain version number in the `config.xml` file for the domain is updated to the Administration Server's installed WebLogic Server version.
- Reconfiguration templates for all installed Oracle products are automatically selected and applied to the domain. These templates define any reconfiguration tasks that are required to make the WebLogic domain compatible with the current WebLogic Server version.
- Start scripts are updated.

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

Note: Once the domain reconfiguration process starts, it is irreversible. Before running the Reconfiguration Wizard, ensure that you have backed up the domain as covered in the pre-upgrade checklist. If an error or other interruption occurs while running the Reconfiguration Wizard, you must restore the domain by copying the files and directories from the backup location to the original domain directory. This is the only way to ensure that the domain has been returned to its original state before reconfiguration.

Follow these instructions to reconfigure the existing domain using the Reconfiguration Wizard. For general information about how the domain is reconfigured, see Reconfiguring WebLogic Domains in *Upgrading Oracle WebLogic Server*.

[Backing Up the Domain](#)

[Starting the Reconfiguration Wizard](#)

[Reconfiguring the Domain with the Reconfiguration Wizard](#)

5.7.1 Backing Up the Domain

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

To create a backup of the domain directory:

1. Copy the source domain to a separate location to preserve the contents.

For example, copy `C:\domains\mydomain` to `C:\domains\mydomain_backup`.

2. Before updating the domain on each remote Managed Server, create a backup copy of the domain directory on each remote machine.
3. Verify that the backed up versions of the domain are complete.

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

5.7.2 Starting the Reconfiguration Wizard

To start the Reconfiguration Wizard in graphical mode:

1. Sign in to the system on which the domain resides.
2. Open the command shell (on UNIX operating systems) or open a command prompt window (on Windows operating systems).
3. **Edition Based Database Users Only:** If your schemas are configured with EBR database, a default edition name must be manually supplied before you run the Reconfiguration Wizard.

Run the following SQL command to set the default edition:

```
ALTER DATABASE DEFAULT EDITION = edition_name;
```

where *edition_name* is the child edition name.

4. Go to the `oracle_common/common/bin` directory:
 - (UNIX) `ORACLE_HOME/oracle_common/common/bin`
 - (Windows) `ORACLE_HOME\oracle_common\commom\bin`
5. Start the Reconfiguration Wizard with the following logging options:
 - (UNIX) `./reconfig.sh -log=log_file -log_priority=ALL`

- (Windows) `reconfig.cmd -log=log_file -log_priority=ALL`

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

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

Note:

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

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

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

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

5.7.3 Reconfiguring the Domain with the Reconfiguration Wizard

Navigate through the screens in the Reconfiguration Wizard to reconfigure your existing domain before running the Upgrade Assistant

Note: If the source is a clustered environment, run the Reconfiguration Wizard on the primary node only. Use the `pack` and `unpack` utilities to apply the changes to other cluster members in the domain.

To reconfigure the domain with the Reconfiguration Wizard:

1. On the Select Domain screen, specify the location of the domain you want to upgrade or click **Browse** to navigate and select the domain directory. Click **Next**.
2. On the Reconfiguration Setup Progress screen, view the progress of the setup process. When complete, click **Next**.

During this process:

- The reconfiguration templates for your installed products, including Fusion Middleware products, are automatically applied. This updates various domain configuration files such as `config.xml`, `config-groups.xml`, and `security.xml` (among others).
 - Schemas, scripts, and other such files that support your Fusion Middleware products are updated.
 - The domain upgrade is validated.
3. On the Domain Mode and JDK screen, select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use. The supported JDK version for 12c (12.2.1) is 1.8.0_60 and later. Click **Next**.

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

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

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

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

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

NOTE: For any existing 11g datasource, the reconfiguration will preserve the existing values. For new datasources where the schema was created for 12c by the RCU, the default connection data will be retrieved from the _STB schema. If no connection data for a given schema is found in the _STB schema, then the default connection data is used.

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

5. On the JDBC Component Schema screen, verify that the DBMS/Service and the Host name is correct for each component schema and click **Next**.
6. On the JDBC Component Schema Test screen, select all the component schemas and click **Test Selected Connections** to test the connection for each schema. The result of the test is indicated in the **Status** column.

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

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

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

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

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

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

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

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

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

During this process:

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

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

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

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

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

5.8 Upgrading Domain Component Configurations

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

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

Upgrading Domain Component Configurations Using the Upgrade Assistant

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

5.8.1 Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

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

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

Upgrade Assistant Command Line Parameters

5.8.1.1 Upgrade Assistant Command Line Parameters

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

Table 5-10 Upgrade Assistant Command Line Parameters

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

Table 5-10 (Cont.) Upgrade Assistant Command Line Parameters

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

5.8.2 Upgrading Domain Component Configurations Using the Upgrade Assistant

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

After running the Reconfiguration Wizard to reconfigure the WebLogic domain to 12c (12.2.1), you must run the Upgrade Assistant to upgrade the domain *component* configurations to match the updated domain configuration.

To upgrade domain component configurations with the Upgrade Assistant:

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

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

2. On the next screen:
 - Select **All Configurations Used By a Domain**. The screen name changes to WebLogic Components.
 - In the **Domain Directory** field, enter the WebLogic domain directory path.Click **Next**.

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

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

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

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

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

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

Note:

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

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

The response file collects and stores all the information that you have entered, and enables you to perform a silent upgrade at a later time. The silent upgrade

performs exactly the same function that the Upgrade Assistant performs, but you do not have to manually enter the data again. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file.

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

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

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

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

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

Click **Next**.

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

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

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

5.9 Starting Servers and Processes

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

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

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

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

Step 1: Start the Administration Server

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

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

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

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

Step 2: Start Node Manager

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

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

Step 3: Start Oracle Identity Management Components

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

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

Step 4: Start the Managed Servers

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

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

When prompted, enter your user name and password.

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

Step 5: Start System Components

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

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

You can start system components in any order.

5.10 Verifying the Upgrade

After completing all the upgrade steps, verify that the upgrade was successful.

To verify the upgrade:

- If you have not done so already, start the Node Manager, Administration Server, and ODI Managed Server for the Java EE agent.

For more information, see Starting a Java EE Agent in *Installing and Configuring Oracle Data Integrator*.

5.11 Running a Pre-Upgrade Readiness Check

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

About Running a Pre-Upgrade Readiness Check

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

Starting the Upgrade Assistant in Readiness Mode

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

Navigating the Upgrade Assistant Screens (Readiness Mode)

You must navigate all the Upgrade Assistant screens that appear, based on your primary selection, to complete the Readiness Check. In addition, if you want to save the response file for performing readiness check on other domains/system, you can do that on the Readiness Summary screen.

Understanding the Readiness Report

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

5.11.1 About Running a Pre-Upgrade Readiness Check

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

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

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

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

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

Note:

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

5.11.2 Starting the Upgrade Assistant in Readiness Mode

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

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

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

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

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

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

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

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

[Upgrade Assistant Command Line Parameters](#)

5.11.2.1 Upgrade Assistant Command Line Parameters

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

Table 5-11 Upgrade Assistant Command Line Parameters

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

Table 5-11 (Cont.) Upgrade Assistant Command Line Parameters

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

5.11.3 Navigating the Upgrade Assistant Screens (Readiness Mode)

You must navigate all the Upgrade Assistant screens that appear, based on your primary selection, to complete the Readiness Check. In addition, if you want to save the response file for performing readiness check on other domains/system, you can do that on the Readiness Summary screen.

To complete the Readiness Check:

1. The Welcome screen provides an overview of the readiness check. Review the information on this screen and click **Next**.
2. You can perform readiness check on schemas or domain configurations that are at a supported starting point.

On the Readiness Check Type screen, select one of the following options depending upon the type of readiness check you want to perform:

- **Individually Selected Schemas**
 - **Domain Based**
3. To review and perform readiness check on specific schemas, select **Individually Selected Schemas**.

The **Individually Selected Schemas** option allows you to select the schemas you want to review before the upgrade. When you perform the readiness check on the

schemas, the readiness check report tells you whether a schema is supported for an upgrade, or where an upgrade is needed. Click **Next**.

Note: When you select the **Individually Selected Schemas** option, the screen name changes from Readiness Check Type to Selected Schema.

4. To perform a readiness check per domain, select **Domain Based**.

The **Domain Based** option allows you to check all of the upgrade-eligible schemas and/or component configurations used by the domain. The Upgrade Assistant detects all of the schemas for you. You can check schemas and component configurations at the same time. Or, if you prefer, you can select to review one thing at a time by selecting one of the following options:

- **Include checks for all schemas**

Select this option to enable the Upgrade Assistant to discover and review all components that have a schema available to upgrade.

- **Include checks for all configurations**

Select this option to review component configurations for a managed WebLogic Server domain.

Note: When you select the **Domain Based** option, the screen name changes from Readiness Check Type to Schemas and Configuration.

Specify the **Domain Directory** by entering the path to your domain or by clicking **Browse** to use the navigation tree.

If you want to perform additional readiness check when your domain is online, select the **Perform online and offline readiness checks** option. In that case, you must specify the domain's host, port, user name, and password in the respective fields. For more information about these fields, click **Help**.

If you do not select this option your domain can be offline. Click **Next** to continue.

5. The Available Components screen is displayed if you have selected **Individually Selected Schemas** option.

This screen lists the available components for which the schemas will be selected. If you select something here, readiness check will be performed on that component's schema.

Select one or more components from the list to perform readiness check on them and click **Next**.

Note:

- Based on what you select on the Available Components screen, you will see additional screens. For example STB schema, Domain Directory, WLS Schema, MDS Schema (UCSUMS), etc.
 - If you select some component that has dependent components, then those components are automatically selected. For example, if you select Oracle Platform Security Services, then Oracle Audit Services is automatically selected.
 - In next screens, depending upon your selection on the Available Components screen, you must specify the domain directory (if you select Oracle Audit Services), or specify schema credentials to connect to the selected schema (if you select any other services, for example).
 - Select the Database Type, specify the Database Connect String, DBA User Name, and DBA Password. Then click **Connect**. For more information about entering values in these fields, click **Help**.
 - If the connection is successful, you can see the success message in the message bar at the bottom of the screen.
 - Select the **Schema User Name** from the drop-down list and specify the schema password in the **Schema Password** field. Click **Next**.
-

6. The Component List screen is displayed if you have selected **Domain Based** option.

The Component List screen is read-only and provides a list of components that are included in the domain you have selected and that are to be checked.

Review the list to make sure that all the components within your domain are listed and click **Next**.

7. The Readiness Summary screen provides a high-level overview of the readiness checks performed based on your selections.

To save your selections in a response file, click **Save Response File**.

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

Click **Next**.

8. The Readiness Check screen displays the current status of the readiness check. Depending on what you have selected to check, the process can take several minutes.

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

For a detailed text report, click **View Readiness Report**. This button is enabled only after all the readiness checks are complete.

Remember: If you are running the readiness check on your online production environment, Oracle recommends that you perform the check during off-peak hours to prevent performance degradation.

When done, click **Continue**.

9. The End of Readiness either displays **Readiness Success or **Readiness Failure**.**

If the readiness check is successful, you can now review the complete report by clicking **View Readiness Report**. If the readiness check encounters an issue or error, review the log file to identify the issues, correct the issues, and then restart the readiness check.

Oracle recommends that even with a successful completion of the readiness check, you should review the Readiness Report before you perform the actual upgrade.

A formatted Readiness Report is prepared for you after running the check. Make sure that you review the report and correct any issues before you start the actual upgrade. Use the **Find** option to search for a particular word within the report (such as a schema name or command, for example). The report also indicates where the completed Readiness Check Report file is located.

The **View Log** button is available on every screen. You can use it to see the latest logged information. The log file is managed by the command line options you set. For more information about the command line parameters, see [Upgrade Assistant Command Line Parameters](#).

5.11.4 Understanding the Readiness Report

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

The format of the readiness report file is:

`readiness_timestamp.txt`

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

A readiness report contains the following information:

Table 5-12 Readiness Report Elements

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

Table 5-12 (Cont.) Readiness Report Elements

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

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

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

```
This readiness check report was created on Tue May 30 11:15:52 EDT 2016
Log file is located at: ORACLE_HOME/oracle_common/upgrade/logs/
ua2016-05-30-11-14-06AM.log
Readiness Check Report File: ORACLE_HOME/oracle_common/upgrade/logs/
readiness2016-05-30-11-15-52AM.txt
```

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

```
Oracle Metadata Services
```

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

```
Schema User Name: DEV11_MDS
```

```
Database Type: Oracle Database
```

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

VERSION Schema DEV11_MDS is currently at version 12.1.1.1.0. Readiness checks will now be performed.

Starting schema test: TEST_REQUIRED_TABLES Test that the schema contains all the required tables

Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema contains all the required tables +++ PASS

Starting schema test: TEST_REQUIRED_PROCEDURES Test that the schema contains all the required stored procedures

EXCEPTION Schema is missing a required procedure: GETREPOSITORYFEATURES

Completed schema test: TEST_REQUIRED_PROCEDURES --> Test that the schema contains all the required stored procedures +++ FAIL

Starting schema test: TEST_REQUIRED_VIEWS Test that the schema contains all the required database views

Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema contains all the required database views +++ PASS

Starting index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes

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

Starting index test for table MDS_COMPONENTS: TEST_REQUIRED_INDEXES --> Test that the table contains all the required indexes

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

Starting schema test: TEST_REQUIRED_TRIGGERS Test that the schema has all the required triggers

Completed schema test: TEST_REQUIRED_TRIGGERS --> Test that the schema has all the required triggers +++ PASS

Starting schema test: TEST_MISSING_COLUMNS Test that tables and views are not missing any required columns

Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and views are not missing any required columns +++ PASS

Starting schema test: TEST_UNEXPECTED_TABLES Test that the schema does not contain any unexpected tables

Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema does not contain any unexpected tables +++ PASS

Starting schema test: TEST_UNEXPECTED_PROCEDURES Test that the schema does not contain any unexpected stored procedures

Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the schema does not contain any unexpected stored procedures +++ PASS

Starting schema test: TEST_UNEXPECTED_VIEWS Test that the schema does not contain any unexpected views

Completed schema test: TEST_UNEXPECTED_VIEWS --> Test that the schema does not contain any unexpected views +++ PASS

Starting index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes

Completed index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes +++ PASS

Completed index test for table MDS_LABELS: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes +++ PASS

Starting index test for table MDS_LARGE_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test that the table does not contain any unexpected indexes

Starting schema test: TEST_UNEXPECTED_TRIGGERS Test that the schema does not contain any unexpected triggers

Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema does not contain any unexpected triggers +++ PASS

Starting schema test: TEST_UNEXPECTED_COLUMNS Test that tables and views do not contain any unexpected columns

Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables and views do not contain any unexpected columns +++ PASS

Starting datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 --> Test that all table columns have the proper datatypes

```

Completed datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
Starting datatype test for table MDS_COMPONENTS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
Starting permissions test: TEST_DBA_TABLE_GRANTS Test that DBA user has
privilege to view all user tables
Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA user has
privilege to view all user tables +++ PASS
Starting schema test: TEST_ENOUGH_TABLESPACE Test that the schema tablespaces
automatically extend if full
Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the schema
tablespaces automatically extend if full +++ PASS
Starting schema test: TEST_USER_TABLESPACE_QUOTA Test that tablespace quota for
this user is sufficient to perform the upgrade
Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that tablespace quota
for this user is sufficient to perform the upgrade +++ PASS
Starting schema test: TEST_ONLINE_TABLESPACE Test that schema tablespaces are
online
Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema tablespaces
are online +++ PASS
Starting schema test: TEST_DATABASE_VERSION Test that the database server
version number is supported for upgrade
INFO Database product version: Oracle Database 11g Enterprise Edition Release
11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Completed schema test: TEST_DATABASE_VERSION --> Test that the database server
version number is supported for upgrade +++ PASS
Finished readiness check of Oracle Metadata Services with status: FAILURE.

```

If you are running the 12.1.3.0 version of Oracle Fusion Middleware IAU Schemas that were upgraded from 11.1.1.7 and later or 12.1.2.0 releases, your readiness check may fail with the following error:

```

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

```

Note: You can ignore the “missing index” error in the Readiness Report. It is a known issue. The corresponding missing index is added during the schema upgrade operation. This error does not occur if the schema to be upgraded was created in 12c using the RCU.

Part III

Upgrading Oracle Data Integrator from a Previous 12c Release

This part of *Upgrading Oracle Data Integrator* provides information about upgrading Oracle Data Integrator from a previous 12c release, such as 12.1.2 and 12.1.3.

[Upgrading an Oracle Data Integrator Standalone Agent Environment from a Previous 12c Release](#)

Follow these steps to upgrade an Oracle Data Integrator standalone agent environment when it is not part of a WebLogic domain from a previous Oracle Fusion Middleware 12c release to 12c (12.2.1).

[Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from a Previous 12c Release](#)

Follow these steps to upgrade an Oracle Data Integrator standalone agent environment when it is part of a WebLogic domain from a previous Oracle Fusion Middleware 12c release to 12c (12.2.1).

[Upgrading an Oracle Data Integrator Java EE Agent Environment from a Previous 12c Release](#)

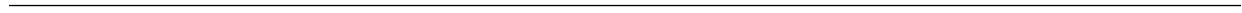
Follow these steps to upgrade an Oracle Data Integrator Java EE agent environment from a previous Oracle Fusion Middleware 12c release to 12c (12.2.1) using Upgrade Assistant.

Upgrading an Oracle Data Integrator Standalone Agent Environment from a Previous 12c Release

Follow these steps to upgrade an Oracle Data Integrator standalone agent environment when it is not part of a WebLogic domain from a previous Oracle Fusion Middleware 12c release to 12c (12.2.1).

Table 6-1 Tasks for Upgrading Oracle Data Integrator

Task	Description
If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.	See: <ul style="list-style-type: none"> • Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1) • Pre-Upgrade Requirements
Upgrade the Oracle Data Integrator standalone agent environment: <ul style="list-style-type: none"> • Install Oracle Data Integrator 12c (12.2.1). • Start the Upgrade Assistant to upgrade the database schemas if needed. • Start the Upgrade Assistant (again) to upgrade standalone system component configurations. • Complete and verify the upgrade. 	The steps to upgrade the Oracle Data Integrator Java EE agent environment from a previous 12c release (12.1.2 and 12.1.3) to 12c (12.2.1) are similar to upgrading from 11g. You do not need to run the Repository Creation Utility (RCU) to create the required 12c schemas if they already exist. See Upgrading an Oracle Data Integrator Standalone Agent Environment from 11g .



Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from a Previous 12c Release

Follow these steps to upgrade an Oracle Data Integrator standalone agent environment when it is part of a WebLogic domain from a previous Oracle Fusion Middleware 12c release to 12c (12.2.1).

Table 7-1 Tasks for Upgrading Oracle Data Integrator

Task	Description
If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.	See: <ul style="list-style-type: none"> Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1) Pre-Upgrade Requirements
Upgrade the Oracle Data Integrator standalone collocated agent environment: <ul style="list-style-type: none"> Install Oracle Data Integrator 12c (12.2.1). Start the Upgrade Assistant to upgrade the database schemas. Start the Reconfiguration Wizard to reconfigure the domain. Start the Upgrade Assistant (again) to upgrade domain component configurations. Complete and verify the upgrade. 	The steps to upgrade the Oracle Data Integrator Java EE agent environment from a previous 12c release (12.1.2 and 12.1.3) to 12c (12.2.1) are similar to upgrading from 11g. You do not need to run the Repository Creation Utility (RCU) to create the required 12c schemas if they already exist. See Upgrading an Oracle Data Integrator Standalone Collocated Agent Environment from 11g .



Upgrading an Oracle Data Integrator Java EE Agent Environment from a Previous 12c Release

Follow these steps to upgrade an Oracle Data Integrator Java EE agent environment from a previous Oracle Fusion Middleware 12c release to 12c (12.2.1) using Upgrade Assistant.

Table 8-1 Tasks for Upgrading Oracle Data Integrator

Task	Description
If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.	See: <ul style="list-style-type: none"> Introduction to Upgrading Oracle Data Integrator to 12c (12.2.1) Pre-Upgrade Requirements
Upgrade the Oracle Data Integrator Java EE agent environment: <ul style="list-style-type: none"> Install Oracle Data Integrator 12c (12.2.1). Start the Upgrade Assistant to upgrade the database schemas. Start the Reconfiguration Wizard to reconfigure the domain. Start the Upgrade Assistant (again) to upgrade domain component configurations. Complete and verify the upgrade. 	The steps to upgrade the Oracle Data Integrator Java EE agent environment from a previous 12c release (12.1.2 and 12.1.3) to 12c (12.2.1) are similar to upgrading from 11g. You do not need to run the Repository Creation Utility (RCU) to create the required 12c schemas if they already exist. See Upgrading an Oracle Data Integrator Java EE Agent Environment from 11g .



Part IV

Troubleshooting and Advanced Configurations

This part of *Upgrading Oracle Data Integrator* provides information about troubleshooting upgrade issues, and options for advanced upgrade configurations.

[Troubleshooting the Upgrade](#)

If the upgrade process fails, you must close the Upgrade Assistant, correct the issue, and then restart the Upgrade Assistant.

[Advanced Upgrade Options](#)

You can set extra options for upgrading Oracle Data Integrator.

Troubleshooting the Upgrade

If the upgrade process fails, you must close the Upgrade Assistant, correct the issue, and then restart the Upgrade Assistant.

If the upgrade process fails *after* the upgrade process has started, you must drop the cloned repository and start from a freshly cloned repository in addition to correcting the underlying issue. There is no way to restart the failed upgrade process.

For more troubleshooting information, see *Troubleshooting Your Upgrade in Upgrading with the Upgrade Assistant*.

Troubleshooting Upgrade Log Errors

If the Upgrade Assistant log file contains <#>, 11g interfaces are converted with errors, check for the interface name and ID in the Upgrade Assistant log and correct the issue using ODI Studio in the 11g repository. An error during interface conversion can be caused by a validation failure for the 11g interface, such as an unmatched parenthesis in an expression. The conversion error can cause the resulting mapping to be incomplete, but has no effect on other conversions.

How to proceed when you encounter such interface upgrade errors:

- If the interfaces highlighted by Upgrade Assistant log file are not used and can be ignored, you can continue to use the upgraded 12c repository.
- If the interface is converted to a valid 12c mapping, each erroneous interface needs to be fixed in the 11g environment and reconverted using one of these two methods:
 - Export only those interfaces, and import them into the upgraded 12c repository. The import process internally upgrades the 11g interfaces into 12c mappings. For more information about importing an interface into a 12c repository, see *Importing Objects in Developing Integration Projects with Oracle Data Integrator*.
 - Repeat the entire upgrade with fixed interfaces. This is recommended if the number of erroneous interfaces is large.

Troubleshooting Upgrade Performance Errors

The number of sessions present in a repository affects upgrade performance. Oracle recommends that you archive and purge your session logs to improve upgrade performance.

Troubleshooting DB2 Database Transaction Log Errors

When using a DB2 database, the database transaction log may become full during ODI upgrade.

You can increase the database configuration parameter to allow for a larger log file. A larger log file requires more space, but it reduces the need for applications to retry the operation. You should set the log file size to at least 10000 and the number of primary log files to at least 50. Use the following commands:

```
db2 'update database config for database_alias using LOGFILSIZ 10000'  
db2 'update database config for database_alias using LOGPRIMARY 50'
```

Advanced Upgrade Options

You can set extra options for upgrading Oracle Data Integrator.

These are extra options that were visible in the 12c (12.1.3) Upgrade Assistant, but are hidden in the 12c (12.2.1) Upgrade Assistant. If you need these options, set them in the response file used in silent mode.

Table 10-1 Advanced Upgrade Options

Option name in 12c (12.1.3)	Option name in 12c (12.2.1) response file	Possible values (default = yes)	Description
Upgrade repository to use GUIDs	ODI_GUID.option	yes/no	<p>This selection sets the repository to 12c full mode. All objects are referenced using the 12c GUID rather than the internal ID.</p> <p>You should leave this option checked in order to take advantage of the truly universally unique identification scheme in Oracle data Integrator 12c.</p> <p>Note: This option only applies when upgrading from 11g repository. For the 12c repository, the option is ignored, and the repository stays in the same mode as before the upgrade.</p> <p>If you have custom Knowledge Modules and procedures that use <code>odiRef</code> substitution APIs, which take internal identifiers as parameters such as <code>getFlexFieldValue()</code>, you may choose to not select this option, leaving your repository in "11g compatibility mode." Scenarios generated from objects using such Knowledge Modules and procedures continue to work in "11g compatibility mode" but will not work in 12c full mode. "11g compatibility mode" can be used to smoothly transition the custom Knowledge Modules and procedures to use the new <code>odiRef</code> substitution APIs (the ones that take GUIDs as parameters). After all custom Knowledge Modules and procedures have been modified to use the new <code>odiRef</code> substitution APIs, the repository can be switched to 12c full mode.</p>

Table 10-1 (Cont.) Advanced Upgrade Options

Option name in 12c (12.1.3)	Option name in 12c (12.2.1) response file	Possible values (default = yes)	Description
			<p>To change the repository to full GUID mode after upgrade (in case you did not choose the option in Upgrade Assistant during upgrade), go to Studio, in the ODI drop down menu, select the option Switch Repository Compatibility Mode. This brings up the option to switch to full GUID mode. If the repository is already in full GUID mode, then that option is disabled.</p> <p>For a list of all substitution methods impacted by the GUID changes, see Table A-1.</p>

Table 10-1 (Cont.) Advanced Upgrade Options

Option name in 12c (12.1.3)	Option name in 12c (12.2.1) response file	Possible values (default = yes)	Description
Upgrade interfaces to use 12c mappings - losing 11g SDK compatibility	ODI_SDK.option	yes/no	<p>This selection converts all 11g interfaces to 12c mappings. Once converted to 12c mappings, all of the existing scenarios must be regenerated before use. There is no ability to use existing 11g SDK applications; they must be upgraded to use the 12c SDK.</p> <p>Some conversion is performed, but the resulting mappings are left in 11g compatible mode, which allows them to be modified using 11g Java SDK. But they can only be modified using 11g SDK; in Studio UI they are read-only.</p> <p>If this option is not selected, some conversion to 12c mappings are performed but the resulting mappings are left in "11g compatibility mode." After these interfaces are modified using the 11g SDK, they can then be converted to 12c mappings using the ODI Studio graphical interface or the 12c SDK.</p> <p>Oracle recommends leaving this option checked, unless you have significant amount of Java code that uses the 11g SDK to read or update existing interfaces. This option only applies when upgrading from 11g repository. For the 12c repository the option is ignored, and the repository stays in the same mode as before upgrade.</p> <p>NOTE: In order for this migration to work properly, all interfaces in 11g repository must be valid (for example, they</p>

Table 10-1 (Cont.) Advanced Upgrade Options

Option name in 12c (12.1.3)	Option name in 12c (12.2.1) response file	Possible values (default = yes)	Description
			<p>should not return any errors when validating from 11g Studio, for example). If an 11g interface is not valid, the Upgrade Assistant will try to migrate it into a 12c mapping, but there are no guarantees about the result: the migration of that interface may fail, or exceptions may printed out the in log file. In any case the resulting mapping will be invalid. The best way to ensure a smooth upgrade is to make sure all interfaces in 11g repository are valid to start with.</p> <p>The upgrade process does not stop even if some 11g interfaces fail during the migration; the upgrade will continue until all interfaces are processed.</p>
Use AES-128 encryption algorithm	ODI_AES.option	yes/no	<p>AES with 128-bit keys provides adequate protection for sensitive information. AES with 256-bit keys is required to protect classified information of higher importance. If the option is unchecked, then AES-256 encryption will be used.</p>

Table 10-2 contains descriptions of the combinations of options that may or may not be selected on the ODI Options screen.

Table 10-2 Possible ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	Description
Yes	Yes	Yes	This is the most common combination and is the configuration with which the new 12c repositories are created. With this combination, all objects use the new GUID identification, and all interfaces are converted into full 12c mappings, which can be modified in ODI Studio editors.
Yes	No	Yes	With this combination, the repository stays in ID compatibility mode, which means that <code>odiRef</code> APIs that use legacy numeric identifiers continue to work. Use this combination if you have significant number of custom Knowledge Modules or procedures that use <code>odiRef</code> APIs with numeric IDs as arguments.
No	No	Yes	With this selection, the Knowledge Modules existing in the repository are preserved and are not overwritten with the new 12c updates. Also, the repository stays in ID compatibility mode. Use this combination if you have modified default Knowledge Modules but would like to use full 12c mappings.

Table 10-2 (Cont.) Possible ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	Description
Yes	Yes	No	With combination, the 11g interfaces are converted to 11g compatible mappings, which can be accessed and modified through 11g interface SDK. These mappings are read-only in ODI Studio editors. Use this combination if you have significant investment in programs or scripts that use the 11g interface SDK.
Yes	No	No	With this selection, the repository stays in ID compatibility mode, which means that odiRef APIs that use legacy numeric identifiers continue to work. Also, the 11g interfaces are converted to 11g compatible mappings, which can be accessed and modified through the 11g SDK. However, they are read-only in ODI Studio editors. Use this combination if you have significant number of custom Knowledge Modules or procedures that use odiRef APIs with numeric IDs as arguments, and you have significant investment in programs or scripts that use 11g interface SDK.

Table 10-2 (Cont.) Possible ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	Description
No	No	No	If you do not select any options, the Knowledge Modules existing in the repository are going to be preserved and not overwritten with the new 12c updates. Use this combination if you have existing applications that use the ODI 11g SDK to read or modify these mappings, but you do not have modified Oracle supplied Knowledge Modules for your own purpose.

Table 10-3 describes the Invalid ODI Options screen combinations.

Table 10-3 Invalid ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	Description
No	Yes	No	With this selection, most Knowledge Modules will not function correctly since they will be using deprecated <code>odiRef</code> APIs, which use legacy numeric IDs.
No	Yes	Yes	With this combination, most Knowledge Modules will not function correctly since they will be using deprecated <code>odiRef</code> APIs, which use legacy numeric IDs.

Note:

The **Upgrade topology and security metadata** option can be selected or not selected independently of all the other options and has no effect on the other options.

Also **Use AES-128 encryption algorithm** option can be selected or not selected independently of all the other options and has no effect on the other options.

Substitution API

Upgrading to GUIDs impacts some Substitution API methods that use internal IDs as parameters.

[Table A-1](#) lists the new Substitution API methods.

Table A-1 New Substitution API Methods

Impacted ODIREf Methods	Previous ODIREf Method	New ODIREf Method
getAK() method	getAK("ID")	getAK("GUID") /* Return the Alternate Key GUID */
getCK() method	getCK("ID")	getCK("GUID") /* Return the Check GUID */
getContext() method	getContext("ID")	getContext("GLOBAL_ID") /* Return the Context GUID*/
getFK() method	getFK("ID")	getFK("GUID") /* Return the Foreign Key GUID */
getFK() method	getFK("ID_TABLE_PK")	getFK("GUID_TABLE_PK") /* Return the Primary Table GUID */
getFK() method	getFK("PK_I_MOD")	getFK ("PK_MOD_GUID") /* Return the Model GUID of Primary Key table */
getInfo() method	getInfo("I_SRC_SET")	getInfo("GUID_SRC_SET") /* Return the SourceSet GUID */
getInfo() method	getInfo("CT_ERR_ID")	getInfo("CT_ERR_GUID") /* Return the Checked Source GUID (a datastore for Static Check or a Mapping for a Flow Check) */
getInfo() method	getInfo("DEST_I_CONNECT")	getInfo("DEST_CONNECT_GUID") /* Return the Destination connection GUID */
getInfo() method	getInfo("SRC_I_CONNECT")	getInfo("SRC_CONNECT_GUID") /* Return the Source connection GUID */
getInfo() method	getInfo("DEST_I_PSCHEMA")	getInfo("DEST_PSCHEMA_GUID") /* Return the Destination Physical Schema GUID */
getInfo() method	getInfo("SRC_I_PSCHEMA")	getInfo("SRC_PSCHEMA_GUID") /* Return the Source Physical Schema GUID */

Table A-1 (Cont.) New Substitution API Methods

Impacted ODIRef Methods	Previous ODIRef Method	New ODIRef Method
getInfo() method	getInfo("DEST_I_LSCHEMA")	getInfo("DEST_LSCHEMA_GUID") /* Return the Destination Logical Schema GUID */
getInfo() method	getInfo("SRC_I_LSCHEMA")	getInfo("SRC_LSCHEMA_GUID") /* Return the Source Logical Schema GUID */
getInfo() method	getInfo("DEST_I_TECHNO")	getInfo("DEST_TECHNO_GUID") /* Return the Destination Technology GUID */
getInfo() method	getInfo("SRC_I_TECHNO")	getInfo("SRC_TECHNO_GUID") /* Return the Source Technology GUID */
getLoadPlanInstance() method	getLoadPlanInstance("BATCH_ID")	getLoadPlanInstance("BATCH_GUID") /* Return the batch GUID */
getNotNullCol() method	getNotNullCol("ID")	getNotNullCol("GLOBAL_ID") /* Return the Attribute GUID */
getModel() method	getModel("ID")	getModel("GLOBAL_ID") /* Return the Model GUID */
getPackage() method	getPackage("I_PACKAGE")	getPackage("PACKAGE_GUID") /* Return the Package GUID */
getPK() method	getPK("ID")	getPK("GLOBAL_ID") /* Return the Primary Key GUID */
getPrevStepLog() method	getPrevStepLog("SESS_NO")	getPrevStepLog("SESS_GUID") /* Return the Session GUID */
getSession() method	getSession("SESS_NO")	getSession("SESS_GUID") /* Return the Session GUID */
getSrcTablesList() method	getSrcTablesList("I_TABLE")	getSrcTablesList("TABLE_GUID") /* Return the Source Table GUID */
getStep() method	getStep("SESS_NO")	getStep("SESS_GUID") /* Return the Session GUID */
getTable() method	getTable("ID")	getTable("GLOBAL_ID") /* Return the target Table GUID */
getTargetTable() method	getTargetTable("I_TABLE")	getTargetTable("TABLE_GUID") /* Return the target table GUID */
getUser() method	getUser("I_USER")	getUser("USER_GUID") /* Return the User GUID */