Oracle® Fusion Middleware

Upgrading Oracle Data Integrator

12*c* (12.1.3)

E49725-01

June 2014

Documentation for installers and system administrators that describes how to upgrade Oracle Data Integrator from an 11*g* and 12.1.2 release to 12*c*.

ORACLE®

Oracle Fusion Middleware Upgrading Oracle Data Integrator 12*c* (12.1.3)

E49725-01

Contents

A Appendix

Preface

This document describes how to upgrade an existing 11*g* Oracle Data Integrator environment to 12*c*. To understand the valid starting points for the Oracle Data Integrator upgrade, see Section 1.1.

Audience

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

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For important information about Oracle Fusion Middleware products, see the following manuals:

- *Understanding Oracle Fusion Middleware*

  This book contains important concepts and describes new features in 12*c* that may be important to understand prior to beginning your upgrade.

- *Planning an Upgrade of Oracle Fusion Middleware*

  This book contains important information to help you plan your upgrade.

- *Planning an Installation of Oracle Fusion Middleware*

This book contains important information about preparing your system for the latest release of Oracle Fusion Middleware software.

Conventions

The following text conventions are used in this document:

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

1

Preparing to Upgrade Oracle Data Integrator

This chapter provides a summary of items you should read and understand before upgrading Oracle Data Integrator.

The following topics are covered:

- Section 1.1, "Understanding the Valid Starting Points for the Oracle Data Integrator Upgrade"

- Section 1.2, "Key Differences Between Oracle Data Integrator 11g and Oracle Data Integrator 12c"

- Section 1.3, "Understanding the Oracle Data Integrator Standard Upgrade Topologies"

1.1 Understanding the Valid Starting Points for the Oracle Data Integrator Upgrade

You can upgrade to Oracle data Integrator 12*c* (12.1.3) from the following supported starting points:

- Oracle Data Integrator 11*g* Release 1 (11.1.1.6.0)

- Oracle Data Integrator 11*g* Release 1 (11.1.1.7.0)

- Oracle Data Integrator 12*c* (12.1.2)

The upgrade procedures in this guide explain how to upgrade an existing Oracle Data Integrator 11*g* domain and existing 12.1.2 domain to Oracle Fusion Middleware 12*c* (12.1.3). If your domain 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 11*g* Release 1 (11.1.1.6.0), you must first upgrade your software to one of the following supported versions before you can upgrade to 12*c* (12.1.3):

- To upgrade to 11*g* Release 1 (11.1.1.6.0), see the *Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator* in the 11*g* Release 1 (11.1.1.6.0) documentation library.

- To upgrade to 11*g* Release 1 (11.1.1.7.0), see the *Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator* in the 11*g* Release 1 (11.1.1.7.0) documentation library.

Note: IF your current starting point is 11.1.1.6, Oracle recommends upgrading to 11.1.1.7 first, and then upgrading to 12c.

> **Note:** Upgrade to 12*c* (12.1.3) is only supported from 11*g* Release 1 (11.1.1.6.0) or 11*g* Release 1 (11.1.1.7.0).

1.2 Key Differences Between Oracle Data Integrator 11g and Oracle Data Integrator 12c

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

- Section 1.2.1, "Standalone Agents are Managed by the WebLogic Management Framework"

- Section 1.2.2, "Standalone Agents are Installed in Their Own Directories"

- Section 1.2.3, "Repository Upgrade to 12.1.3 Now Validates Name Uniqueness for Objects"

To understand what's new in general in 12*c*, see "New and Changed Features for 12c (12.1.3)" 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" in *Upgrading to the Oracle Fusion Middleware Infrastructure*.

1.2.1 Standalone Agents are Managed by the WebLogic Management Framework

In 12*c*, 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*.

1.2.2 Standalone Agents are Installed in Their Own Directories

In 12*c*, standalone agents are installed in a separate directory from other ODI components.

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

1.2.3 Repository Upgrade to 12.1.3 Now Validates Name Uniqueness for Objects

The upgrade checks for duplicated names (according to the namespace rules, that is, within the parent) for a number of objects, during the repository upgrade to 12.1.3. The check is done at the beginning (before the actual upgrade is started) and if duplicates are detected, then you must print a report with the list of duplicates in the upgrade log and error out. You should then connect to the 11g repository using Studio, to manually fix the duplicates (either by rename or delete), and then restart the upgrade.

In 12.1.3, the following objects are required to have unique names:

- Interface

- Folder

- Procedure/ KM

> **Note:** The Knowledge Modules must have unique names within the parent scope. That can be either its own project or for global Knowledge Modules.

- Package
- Profile

1.3 Understanding the Oracle Data Integrator Standard Upgrade Topologies

Using the Oracle Fusion Middleware Oracle Data Integrator 11*g* software, you can create a variety of production topologies to suit the needs of your applications, your organization, and your application users.

As a result, it is difficult to provide exact upgrade instructions for every possible Oracle Data Integrator 11*g* installation. To solve this problem, this upgrade documentation provides detailed instructions for upgrading several typical Oracle Data Integrator configurations. These typical topologies are referred to as 12*c* 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.

This guide explains step-by-step how to upgrade the following specific upgrade topologies:

- Oracle Data Integrator Standard Upgrade Topology for Java EE Agents
- Oracle Data Integrator Standard Upgrade Topology for Standalone Agents not Registered with a WebLogic Domain
- Oracle Data Integrator Standard Upgrade Topology for Standalone Agents Registered with a WebLogic Domain

1.3.1 Oracle Data Integrator Standard Upgrade Topology for Java EE Agents

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

The upgrade roadmap and procedures for this topology are in Chapter 2.

Figure 1–1   Oracle Data Integrator Java EE Agent Upgrade Topology
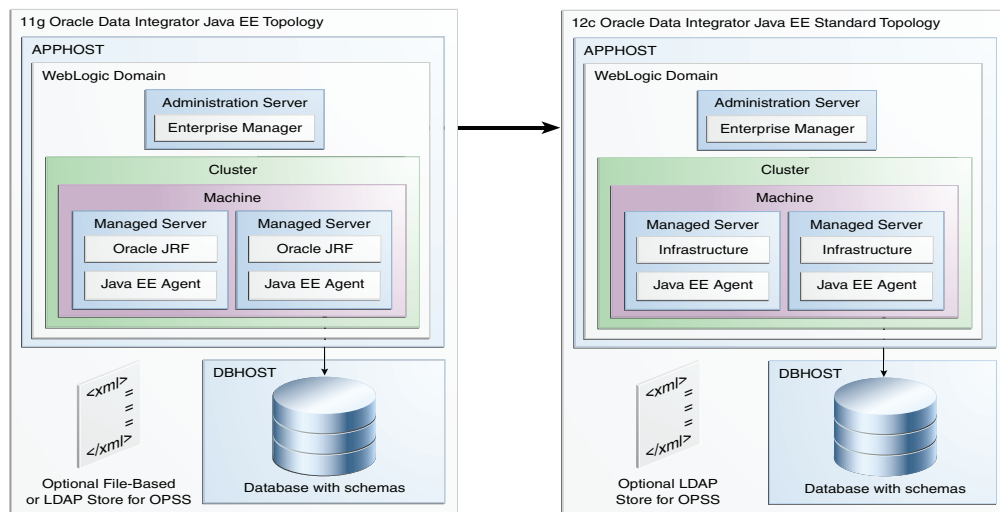
All elements in this topology illustration are described in Table 1–1.

Table 1–1   Description of the Elements in the Oracle Data Integrator Java EE Standard Upgrade Topology

Element	Description and Links to Additional Documentation
11*g* Oracle Data Integrator Java EE Topology	This is the label for the left side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 11*g* Oracle Data Integrator installer.
	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).
	This document describes, step-by-step, how to upgrade this topology to an equivalent topology in 12*c*.
12c Oracle Data Integrator Java EE 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 12*c* Oracle Data Integrator distribution.
	Like the 11*g* 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.
	Unlike the 11*g* topology, only an LDAP based store can be used for OPSS; file-based stores are not allowed in 12*c*.
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.
Database with Schemas	Represents a supported database, where the Oracle Fusion Middleware schemas have been created using the Repository Creation Utility.

Table 1–1   (Cont.) Description of the Elements in the Oracle Data Integrator Java EE Standard Upgrade Topology

Element	Description and Links to Additional Documentation
WebLogic Domain	A logically related group of Java components (in this case, the Administration Server, Managed Servers, Java EE agent, and other related software components).
	For more information, see "What is an Oracle WebLogic Server Domain" in *Understanding Oracle Fusion Middleware*.
Administration Server	The central control entity of a domain which maintains the domain's configuration objects and distributes configuration changes to Managed Servers.
	For more information, see "What is the Administration Server" in *Understanding Oracle Fusion Middleware*.
Enterprise Manager	Oracle Enterprise Manager Fusion Middleware Control.
	For more information, see "Oracle Enterprise Manager Fusion Middleware Control" in *Understanding Oracle Fusion Middleware*.
Cluster	A collection of multiple WebLogic Server instances running simultaneously and working together.
	For more information, see "Understanding Managed Servers and Managed Server Clusters" in *Understanding Oracle Fusion Middleware*.
Machine	Logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are also the logical glue between WebLogic Managed Servers and the Node Manager; in order to start or stop a Managed Server with Node Manager, the Managed Server must be associated with a machine.
Managed Server	Host for your applications, application components, Web services, and their associated resources.
	For more information, see "Understanding Managed Servers and Managed Server Clusters" in *Understanding Oracle Fusion Middleware*.
Java EE Agent	A Java EE agent is a JEE application that is deployed and runs on a Managed Server configured in a WebLogic domain.
	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 *Developer's Guide for Oracle Data Integrator*.
Oracle JRF	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.
	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.

Table 1–1 (Cont.) Description of the Elements in the Oracle Data Integrator Java EE Standard Upgrade Topology

Element	Description and Links to Additional Documentation
Infrastructure	Oracle Fusion Middleware 12*c* term (similar to Oracle JRF) that refers to the collection of services that include the following:
	■ Metadata repository (MDS)
	This contains metadata for Oracle Fusion Middleware components, such as the Oracle Application Developer Framework.
	For more information, see "What is the Metadata Repository" in *Understanding Oracle Fusion Middleware*.
	■ Oracle Application Developer Framework (ADF)
	■ Oracle Web Services Manager (OWSM)

1.3.2 Oracle Data Integrator Standard Upgrade Topology for Standalone Agents not Registered with a WebLogic Domain

In 11*g* Release 1 (11.1.1.6.0) and 11*g* Release 1 (11.1.1.7.0), the following standalone agent configurations (not registered with a WebLogic domain) were possible:

■ Standalone agent as a standalone Oracle instance

■ Standalone agent managed by OPMN

Figure 1–2 shows the 11*g* Oracle Data Integrator standard upgrade topology for standalone agents (not registered with a WebLogic domain) and the resulting Oracle Fusion Middleware 12*c* topology as it appears after you complete the upgrade procedures in this guide.

The upgrade roadmap and procedures for this topology are in Chapter 3.

Figure 1–2 Oracle Data Integrator Standard Upgrade Topology for Standalone Agents not Registered to a WebLogic Domain
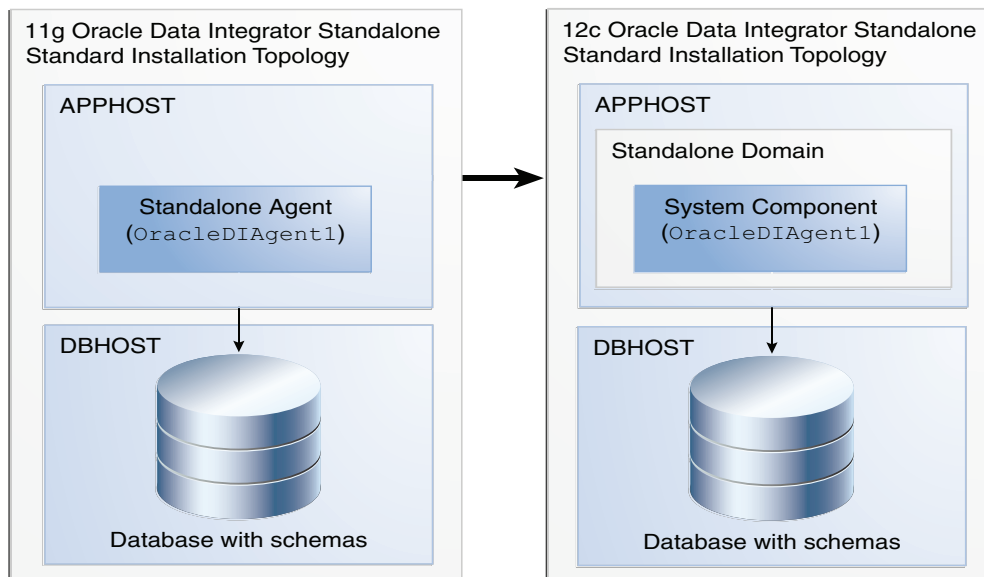
Most of the elements in this topology illustration are described in Table 1–1.

Additional elements and those that are different from the ones in Figure 1–1 are described below in Table 1–2.

Table 1–2   Description of the Elements in the Standalone Agent Standard Upgrade Topology

Element	Description and Links to Additional Documentation
11*g* Oracle Data Integrator Standalone Topology	This is the label for the left side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 11*g* Oracle Data Integrator installer.
	It consists of a single standalone agent (`OracleDIAgent1`) on a single machine. The standalone agent may or may not be managed by OPMN; the upgrade procedures will vary slightly depending on whether or not your 11*g* standalone agent is managed by OPMN.
	A relational database for the Master and Work Repository is also required and is shown in the figure.
	This document describes, step-by-step, how to upgrade this topology to an equivalent topology in 12*c*.
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 12*c* 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.
Standalone Agent	A standalone agent is an Oracle Data Integrator agent that runs in a separate Java Virtual Machine (JVM) process.
	n 11*g*, the standalone agent is created directly as part of the installation.
System Component	In 12*c*, a standalone domain must be created before a standalone agent can be created. A system component corresponds to a standalone agent managed with the WebLogic Management Framework.
Standalone Domain	For more information, see "Standalone Domain" in *Administering Oracle HTTP Server*.

1.3.3 Oracle Data Integrator Standard Upgrade Topology for Standalone Agents Registered with a WebLogic Domain

In 11*g* Release 1 (11.1.1.7.0), it was possible to configure a standalone agent managed by OPMN inside a WebLogic domain.

Figure 1–3 shows the 11*g* Oracle Data Integrator standard upgrade topology for standalone agents (not registered with a WebLogic domain) and the resulting Oracle Fusion Middleware 12*c* topology as it appears after you complete the upgrade procedures in this guide.

The upgrade roadmap and procedures for this topology are in Chapter 3.

Figure 1–3   Oracle Data Integrator Standard Upgrade Topology for Standalone Agents Registered with a WebLogic Domain
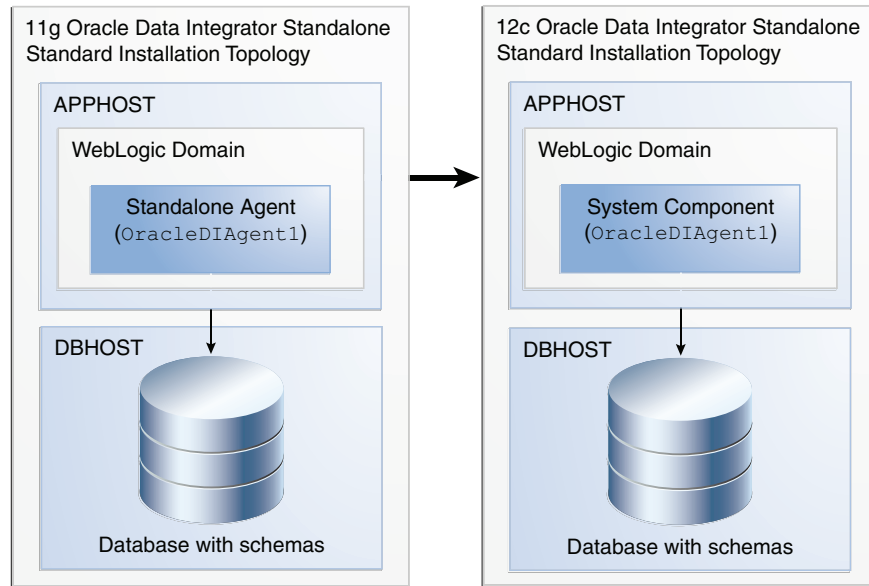
Most of the elements in this topology illustration are described in Table 1–1 and Table 1–2.

Additional elements and those that are different from the ones in the preceding figures are described below in Table 1–2.

Table 1–3   Description of the Elements in the Standard Upgrade Topology for Standalone Agents Registered to a WebLogic Domain

Element	Description and Links to Additional Documentation
11*g* Oracle Data Integrator Standalone Topology	This is the label for the left side of the figure. It shows a typical single-host topology created using the Oracle Fusion Middleware 11*g* Oracle Data Integrator installer.
	It consists of a single standalone agent (`OracleDIAgent1`) on a single machine. The standalone agent is managed by OPMN and is registered to the WebLogic domain in which it resides.
	A relational database for the Master and Work Repository is also required and is shown in the figure.
	This document describes, step-by-step, how to upgrade this topology to an equivalent topology in 12*c*.
12*c* 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 12*c* Oracle Data Integrator distribution.
	It consists of a single standalone agent (`OracleDIAgent1`) configured in a WebLogic domain, along with a relational database for the Master and Work Repository.

2

Upgrading the Oracle Data Integrator Java EE Agent Environment from 11g to 12.1.3

This chapter describes the tasks required to upgrade the 11*g* Oracle Data Integrator Java EE agent environment to Oracle Fusion Middleware 12*c* (see Section 1.1 for valid 11*g* starting points).

The following topics are covered in this chapter:

- Section 2.1, "Understanding the Java EE Agent Environment Upgrade Roadmap"
- Section 2.2, "Completing and Verifying the Upgrade"

2.1 Understanding the Java EE Agent Environment Upgrade Roadmap

Figure 2–1 provides an overview of the upgrade process for the Java EE agent topology.

Figure 2–1   Java EE Agent Upgrade Process Flow

Table 2–1 describes each of the steps in the upgrade process flowchart which is shown in Figure 2–1. The table also provides information on where to go to get more information on each step in the process.

Table 2–1   Oracle Data Integrator J2EE Agent Upgrade Process

Task	Description	More Information
Review and understand the topics in Chapter 1 of this document.	This chapter provides important information you should read and understand before beginning your upgrade process.	Chapter 1
Plan your upgrade.	Understand important concepts and tasks when planning your upgrade.	Section 6.1.1
Backup your existing 11*g* environment.	Before you begin your upgrade, make a complete backup of your existing 11*g* environment.	Section 6.1.2
If necessary, upgrade the repository database to a supported database version	The Oracle Data Integrator Repositories are stored into database schemas. Before you can upgrade these schemas, you must make sure the database is a supported version.	Section 6.1.3
If necessary, re-associate your file-based security store.	If you are using a file-based security store in your 11g environment, then reassociate the file-based store with the database-based repository and OPSS schema.	Section 6.1.4

Table 2–1 (Cont.) Oracle Data Integrator J2EE Agent Upgrade Process

Task	Description	More Information
Copy (clone) existing Master and Work Repositories before running the Upgrade Assistant	Oracle recommends that you perform repository upgrades on cloned repository schemas.	Section 6.1.5
Verify that the work repositories are attached to the correct master repository schema and host	The Upgrade Assistant upgrades all work repositories attached to master repository. Each work repository must be attached to the correct master repository schema and host before performing the upgrade.	Section 6.1.7
Create a database backup of the ODI schema that will be upgraded.	Creating a backup is mandatory if the repository schemas have not been cloned and you are attempting to upgrade a non-cloned schema. Performing a backup of the ODI schemas is particularly important if the upgrade fails and corrupts the content. With a backup, you can delete the corrupted schemas and re-clone the originals to complete the upgrade.	Section 6.1.8
Install and Configure the necessary Oracle Fusion Middleware 12*c* products	Install Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12*c*. As part of the product installation, you should also follow the instructions to run RCU and create any new 12*c* schemas (for example, Service Table). The Upgrade Assistant is available as part of the 12*c* product installation.	Section 6.2
Run Upgrade Assistant to Upgrade the Oracle Data Integrator Master Repository and Work Repository schema.	The Upgrade Assistant upgrades the Oracle Data Integrator 11*g* repository schemas to Oracle Data Integrator 12*c*.	Section 6.3
Upgrade your Java EE environment.	After upgrading the 11*g* schemas, run the reconfiguration wizard to reconfigure your 11*g* domain. Running the reconfiguration wizard concludes the upgrade process for the Java EE agent environment. Note: If any JRF component upgrade is needed, run the Upgrade Assistant again after the Reconfiguration Wizard.	Section 6.4

2.2 Completing and Verifying the Upgrade

To complete and verify the upgrade, perform the tasks in this section.

- Section 2.2.1, "Configuring and Starting the Node Manager"
- Section 2.2.2, "Starting the Administration Server"
- Section 2.2.3, "Starting the Managed Servers"

2.2.1 Configuring and Starting the Node Manager

Refer to the following to configure and start the Node Manager:

- For more information about configuring the Node Manager, see "Completing the Node Manager Configuration" in *Upgrading Oracle WebLogic Server*.

■ For information about starting the Node Manager, see "Starting and Stopping Node Manager" in *Administering Oracle Fusion Middleware*.

2.2.2 Starting the Administration Server

Start Administration Server, since server was down while upgrade was being done.

For more information, see "Starting and Stopping Administration Servers" in *Administering Oracle Fusion Middleware*.

■ Administration Console URL:

```
http://administration_server_host:administration_server_port/console
```

■ Fusion Middleware Control:

```
http://administration_server_host:administration_server_port/em
```

> **Note:** After upgrade, you must run your administration tools from the new 12*c* Oracle home and not from the 11*g* Oracle home.

2.2.3 Starting the Managed Servers

Refer to "Starting the Managed Server" in *Installing and Configuring Oracle Data Integrator* for instructions on how to start the Managed Server in the domain.

3

Upgrading the Oracle Data Integrator Standalone Agent Environment (no WebLogic Domain) from 11g to 12.1.3

This chapter describes the tasks required to upgrade the 11*g* Oracle Data Integrator standalone agent environment to Oracle Fusion Middleware 12*c* (see Section 1.1 for valid 11*g* starting points).

The following upgrade scenarios are covered:

- Upgrading an 11*g* Release 1 (11.1.1.6.0) or 11*g* Release 1 (11.1.1.7.0) standalone agent not managed by OPMN to 12*c*.

- Upgrading an 11*g* Release 1 (11.1.1.6.0) or 11*g* Release 1 (11.1.1.7.0) standalone agent managed by OPMN to 12*c*.

The following topics are covered in this chapter:

- Section 3.1, "Understanding the Standalone Agent Upgrade Process"

- Section 3.2, "Verifying the Upgrade"

3.1 Understanding the Standalone Agent Upgrade Process

Figure 3–1 provides an overview of the upgrade process for the standalone agent topology.

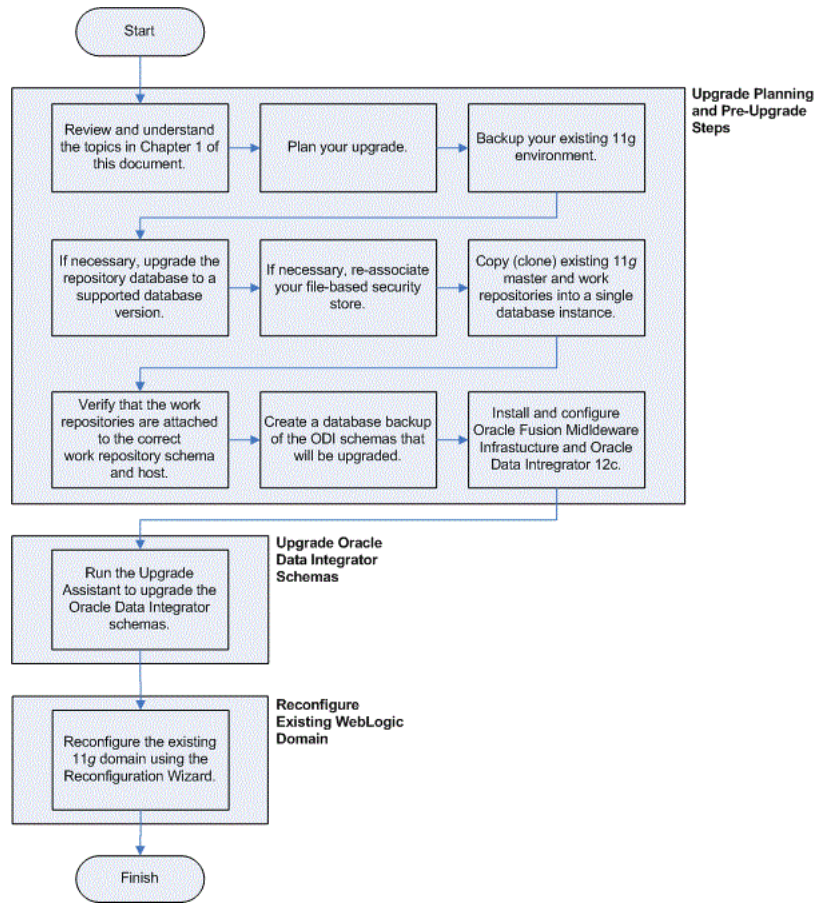**Figure 3–1   Upgrade Process Flowchart for Standalone Agents (no WebLogic Domain)**

Table 3–1 describes each of the steps in the upgrade process flowchart which is shown in Figure 3–1. The table also provides information on where to go to get more information on each step in the process.

Table 3–1   Oracle Data Integrator Standalone Agent Upgrade Process

Task	Description	More Information
Review and understand the topics in Chapter 1 of this document.	This chapter provides important information you should read and understand before beginning your upgrade process.	Chapter 1
Plan your upgrade.	Understand important concepts and tasks when planning your upgrade.	Section 6.1.1
Backup your existing 11*g* environment.	Before you begin your upgrade, make a complete backup of your existing 11*g* environment.	Section 6.1.2
If necessary, upgrade the repository database to a supported database version	The Oracle Data Integrator Repositories are stored into database schemas. Before you can upgrade these schemas, you must make sure the database is a supported version.	Section 6.1.3
Copy (clone) existing Master and Work Repositories before running the Upgrade Assistant	Oracle recommends that you perform repository upgrades on cloned repository schemas.	Section 6.1.5

Table 3–1   (Cont.) Oracle Data Integrator Standalone Agent Upgrade Process

Task	Description	More Information
Verify that the work repositories are attached to the correct work repository schema and host	The Upgrade Assistant upgrades all work repositories attached to master repository. Each work repository must be attached to the correct work repository schema and host before performing the upgrade.	Section 6.1.7
Create a database backup of the ODI schema that will be upgraded.	Creating a backup is mandatory if the repository schemas have not been cloned and you are attempting to upgrade a non-cloned schema. Performing a backup of the ODI schemas is particularly important if the upgrade fails and corrupts the content. With a backup, you can delete the corrupted schemas and re-clone the originals to complete the upgrade.	Section 6.1.8
Install and Configure Oracle Data Integrator 12*c*.	Install Oracle Data Integrator 12*c*. This procedure includes creating the necessary database schemas. The Upgrade Assistant is available as part of the ODI product installation.	Section 6.2
Run Upgrade Assistant to Upgrade the Oracle Data Integrator Master Repository and Work Repository schema.	The Upgrade Assistant upgrades the Oracle Data Integrator 11*g* repository schemas to Oracle Data Integrator 12*c*.	Section 6.3
If your standalone agent is managed by OPMN, run the Upgrade Assistant again to upgrade the Oracle Data Integrator standalone agent.	This step configures the 11*g* standalone agent infrastructure to 12*c*.	Section 6.5

3.2 Verifying the Upgrade

Perform the tasks in this section to verify that the upgrade was successful.

- Section 3.2.1, "Starting the Node Manager"
- Section 3.2.2, "Starting the Standalone Agent"

3.2.1 Starting the Node Manager

For more information about starting the Node Manager, see "Starting the Node Manager" in *Installing and Configuring Oracle Data Integrator*.

3.2.2 Starting the Standalone Agent

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

4

Upgrading the Oracle Data Integrator Standalone Agent Environment (with WebLogic Domain) from 11g to 12.1.3

This chapter describes the tasks required to upgrade the 11*g* Oracle Data Integrator standalone agent environment to Oracle Fusion Middleware 12*c* (see Section 1.1 for valid 11*g* starting points).

The following specific upgrade scenario is covered:

- Upgrading an 11*g* Release 1 (11.1.1.7.0) standalone agent managed by OPMN and registered to a WebLogic domain to 12*c*.

The following topics are covered in this chapter:

- Section 4.1, "Understanding the Standalone Agent Upgrade Process"
- Section 4.2, "Verifying the Upgrade"

4.1 Understanding the Standalone Agent Upgrade Process

Figure 4–1 provides a roadmap of the tasks required to upgrade your standalone agent that is managed by OPMN and registered to a WebLogic domain.

Figure 4–1   Upgrade Process for a Standalone Agent Managed by OPMN and Registered to a WebLogic Domain
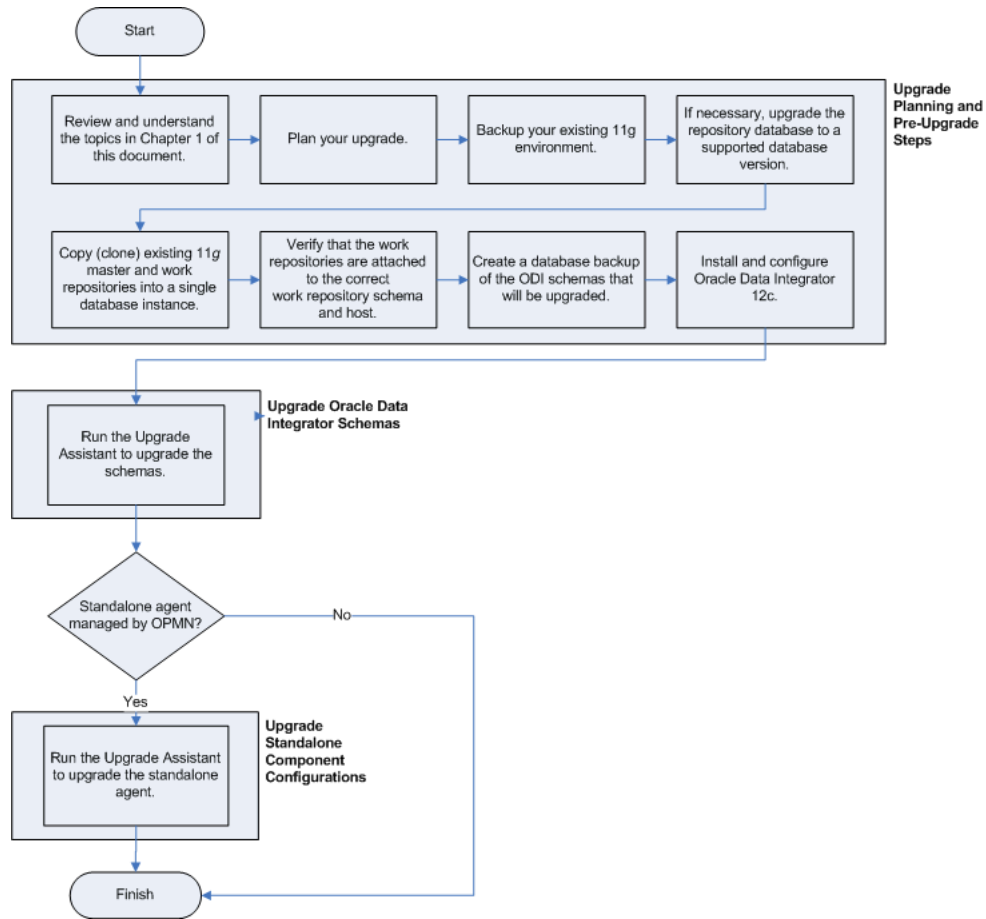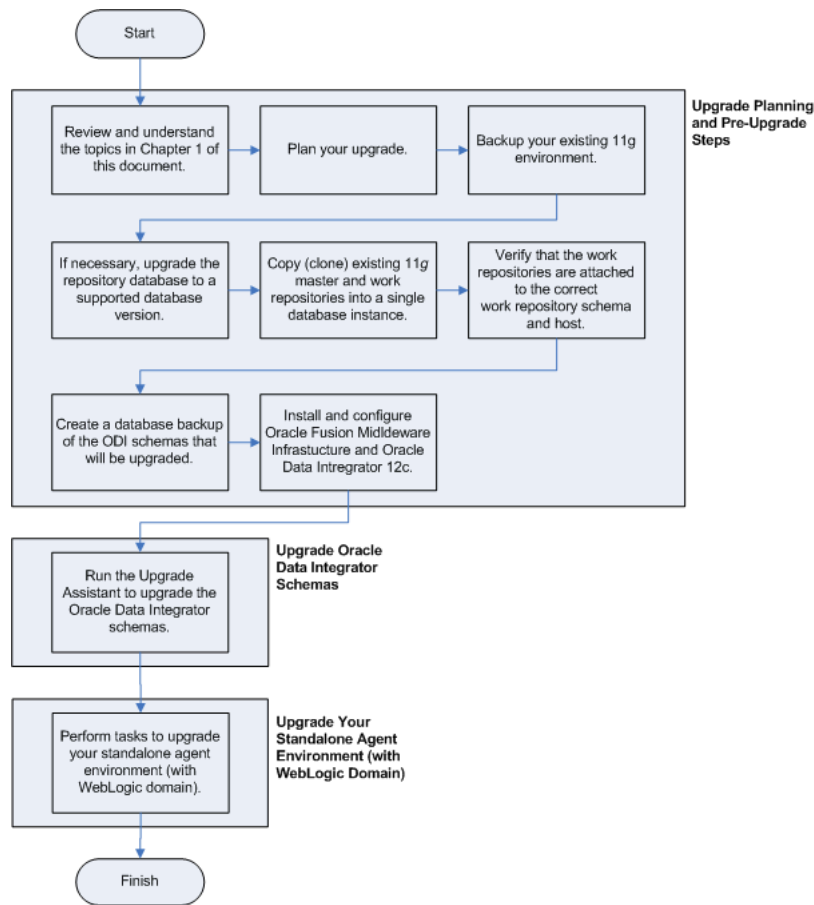

Table 4–1 describes each of the steps in the upgrade process flowchart which is shown in Figure 4–1. The table also provides information on where to go to get more information on each step in the process.

Table 4–1   Oracle Data Integrator Upgrade Process

Task	Description	More Information
Review and understand the topics in Chapter 1 of this document.	This chapter provides important information you should read and understand before beginning your upgrade process.	Chapter 1
Plan your upgrade.	Understand important concepts and tasks when planning your upgrade.	Section 6.1.1
Backup your existing 11g environment.	Before you begin your upgrade, make a complete backup of your existing 11g environment.	Section 6.1.2
If necessary, upgrade the repository database to a supported database version	The Oracle Data Integrator Repositories are stored into database schemas. Before you can upgrade these schemas, you must make sure the database is a supported version.	Section 6.1.3
Copy (clone) existing Master and Work Repositories before running the Upgrade Assistant	Oracle recommends that you perform repository upgrades on cloned repository schemas.	Section 6.1.5

Table 4–1 (Cont.) Oracle Data Integrator Upgrade Process

Task	Description	More Information
Verify that the work repositories are attached to the correct master repository schema and host	The Upgrade Assistant upgrades all work repositories attached to master repository. Each work repository must be attached to the correct master repository schema and host before performing the upgrade.	Section 6.1.7
Create a database backup of the ODI schema that will be upgraded.	Creating a backup is mandatory if the repository schemas have not been cloned and you are attempting to upgrade a non-cloned schema. Performing a backup of the ODI schemas is particularly important if the upgrade fails and corrupts the content. With a backup, you can delete the corrupted schemas and re-clone the originals to complete the upgrade.	Section 6.1.8
Install and Configure the necessary Oracle Fusion Middleware 12*c* products	Install Oracle Fusion Middleware Infrastructure and Oracle Data Integrator 12*c*. The Upgrade Assistant is available as part of the 12*c* product installation.	Section 6.2
Run Upgrade Assistant to Upgrade the Oracle Data Integrator Master Repository and Work Repository schema.	The Upgrade Assistant upgrades the Oracle Data Integrator 11*g* repository schemas to Oracle Data Integrator 12*c*.	Section 6.3
Upgrade your standalone agent environment (with WebLogic domain).	After upgrading the 11*g* schemas, you can upgrade your standalone agent environment (with WebLogic domain). Tasks to do so include: ■ Reconfiguring the domain with the reconfiguration wizard. ■ Running the Upgrade Assistant to upgrade your component configurations for a managed WebLogic domain.	Section 6.6

4.2 Verifying the Upgrade

To complete and verify the upgrade, perform the following tasks:

- Section 4.2.1, "Configuring and Starting the Node Manager"

- Section 4.2.2, "Restarting the Administration Server"

- Section 4.2.3, "Starting the Colocated Standalone Agent"

4.2.1 Configuring and Starting the Node Manager

Refer to the following to configure and start the Node Manager:

- For more information about configuring the Node Manager, see "Completing the Node Manager Configuration" in *Upgrading Oracle WebLogic Server*.

- For information about starting the Node Manager, see "Starting and Stopping Node Manager" in *Administering Oracle Fusion Middleware*.

4.2.2 Restarting the Administration Server

Stop and restart the Administration Server.

For more information, see "Starting and Stopping Administration Servers" in *Administering Oracle Fusion Middleware*.

- Administration Console URL:

  `http://`*`administration_server_host`*`:`*`administration_server_port`*`/console`

- Fusion Middleware Control:

  `http://`*`administration_server_host`*`:`*`administration_server_port`*`/em`

> **Note:** After upgrade, you must run your administration tools from the new 12*c* Oracle home and not from the 11*g* Oracle home.

4.2.3 Starting the Colocated Standalone Agent

You must start the colocated standalone agent to verify the upgrade. For more information, see "Configuring the Domain for the Colocated Standalone Agent" in *Installing and Configuring Oracle Data Integrator*.

5

Upgrading the Oracle Data Integrator from 12.1.2 to 12.1.3

This chapter describes the tasks required to upgrade an 12.1.2 Oracle Data Integrator environment to 12.1.3.

This chapter contains the following topics:

- Section 5.1, "Upgrading the Oracle Data Integrator Standalone Agent Environment"
- Section 5.2, "Upgrading the Oracle Data Integrator Java EE Agent Environment"
- Section 5.3, "Upgrading the Oracle Data Integrator Colocated Standalone Agent Environment"
- Section 5.4, "Verifying the Upgrade"

5.1 Upgrading the Oracle Data Integrator Standalone Agent Environment

Upgrade Oracle Data Integrator Standalone agent environment from 12.1.2 to 12.1.3 using Upgrade Assistant.

To Upgrade Oracle Data Integrator Standalone agent:

Table 5–1   Oracle Data Integrator Standalone Agent Upgrade Process

Task	Description	More Information
Review and understand the topics in Chapter 1 of this document.	This chapter provides important information you should read and understand before beginning your upgrade process.	Chapter 1
Plan your upgrade.	Understand important concepts and tasks when planning your upgrade.	Section 6.1.1
Backup your existing 12.1.2 environment.	Before you begin your upgrade, make a complete backup of your existing 12.1.2 environment.	Section 6.1.2
If necessary, upgrade the repository database to a supported database version.	The Oracle Data Integrator Repositories are stored into database schemas. Before you can upgrade these schemas, you must make sure the database is a supported version.	Section 6.1.3
Copy (clone) existing Master and Work Repositories before running the Upgrade Assistant	Oracle recommends that you perform repository upgrades on cloned repository schemas.	Section 6.1.5

Table 5–1   (Cont.) Oracle Data Integrator Standalone Agent Upgrade Process

Task	Description	More Information
Verify that the work repositories are attached to the correct work repository schema and host	The Upgrade Assistant upgrades all work repositories attached to master repository. Each work repository must be attached to the correct work repository schema and host before performing the upgrade.	Section 6.1.7
Create a database backup of the ODI schema that will be upgraded.	Creating a backup is mandatory if the repository schemas have not been cloned and you are attempting to upgrade a non-cloned schema. Performing a backup of the ODI schemas is particularly important if the upgrade fails and corrupts the content. With a backup, you can delete the corrupted schemas and re-clone the originals to complete the upgrade.	Section 6.1.8
Install and Configure Oracle Data Integrator 12.1.3.	Install Oracle Data Integrator 12.1.3. This procedure includes creating the necessary database schemas. The Upgrade Assistant is available as part of the ODI product installation.	Section 6.2
Run Upgrade Assistant to Upgrade the Oracle Data Integrator Master Repository and Work Repository schema.	The Upgrade Assistant upgrades the Oracle Data Integrator 12.1.2 repository schemas to Oracle Data Integrator 12.1.3.	Section 6.3
Review the configuration summary on the Configuration Summary screen, then click **Reconfig** to reconfigure the domain.	The Reconfiguration Wizard reconfigures the domain.	Section 6.5

5.2 Upgrading the Oracle Data Integrator Java EE Agent Environment

Upgrade Oracle Data Integrator Java EE agent environment from 12.1.2 to 12.1.3 using Upgrade Assistant.

To Upgrade Oracle Data Integrator Standalone agent:

Table 5–2   Oracle Data Integrator Java EE Upgrade Process

Task	Description	More Information
Review and understand the topics in Chapter 1 of this document.	This chapter provides important information you should read and understand before beginning your upgrade process.	Chapter 1
Plan your upgrade.	Understand important concepts and tasks when planning your upgrade.	Section 6.1.1
Backup your existing 12.1.2 environment.	Before you begin your upgrade, make a complete backup of your existing 12.1.2 environment.	Section 6.1.2
If necessary, upgrade the repository database to a supported database version.	The Oracle Data Integrator Repositories are stored into database schemas. Before you can upgrade these schemas, you must make sure the database is a supported version.	Section 6.1.3
Copy (clone) existing Master and Work Repositories before running the Upgrade Assistant	Oracle recommends that you perform repository upgrades on cloned repository schemas.	Section 6.1.5

Table 5–2   (Cont.) Oracle Data Integrator Java EE Upgrade Process

Task	Description	More Information
Verify that the work repositories are attached to the correct work repository schema and host	The Upgrade Assistant upgrades all work repositories attached to master repository. Each work repository must be attached to the correct work repository schema and host before performing the upgrade.	Section 6.1.7
Create a database backup of the ODI schema that will be upgraded.	Creating a backup is mandatory if the repository schemas have not been cloned and you are attempting to upgrade a non-cloned schema. Performing a backup of the ODI schemas is particularly important if the upgrade fails and corrupts the content. With a backup, you can delete the corrupted schemas and re-clone the originals to complete the upgrade.	Section 6.1.8
Install and Configure Oracle Data Integrator 12.1.3.	Install Oracle Data Integrator 12.1.3. This procedure includes creating the necessary database schemas. The Upgrade Assistant is available as part of the ODI product installation.	Section 6.2
Run Upgrade Assistant to Upgrade the Oracle Data Integrator Master Repository and Work Repository schema.	The Upgrade Assistant upgrades the Oracle Data Integrator 12.1.2 repository schemas to Oracle Data Integrator 12.1.3.	Section 6.3
Review the configuration summary on the Configuration Summary screen, then click **Reconfig** to reconfigure the domain.	The Reconfiguration Wizard reconfigures the domain.	Section 6.4

5.3 Upgrading the Oracle Data Integrator Colocated Standalone Agent Environment

Upgrade Oracle Data Integrator Colocated Standalone agent environment from 12.1.2 to 12.1.3 using Upgrade Assistant.

To Upgrade Oracle Data Integrator Colocated Standalone agent:

Table 5–3   Oracle Data Integrator Colocated Standalone Upgrade Process

Task	Description	More Information
Review and understand the topics in Chapter 1 of this document.	This chapter provides important information you should read and understand before beginning your upgrade process.	Chapter 1
Plan your upgrade.	Understand important concepts and tasks when planning your upgrade.	Section 6.1.1
Backup your existing 12.1.2 environment.	Before you begin your upgrade, make a complete backup of your existing 12.1.2 environment.	Section 6.1.2
If necessary, upgrade the repository database to a supported database version.	The Oracle Data Integrator Repositories are stored into database schemas. Before you can upgrade these schemas, you must make sure the database is a supported version.	Section 6.1.3

Table 5–3   (Cont.) Oracle Data Integrator Colocated Standalone Upgrade Process

Task	Description	More Information
Copy (clone) existing Master and Work Repositories before running the Upgrade Assistant	Oracle recommends that you perform repository upgrades on cloned repository schemas.	Section 6.1.5
Verify that the work repositories are attached to the correct work repository schema and host	The Upgrade Assistant upgrades all work repositories attached to master repository. Each work repository must be attached to the correct work repository schema and host before performing the upgrade.	Section 6.1.7
Create a database backup of the ODI schema that will be upgraded.	Creating a backup is mandatory if the repository schemas have not been cloned and you are attempting to upgrade a non-cloned schema. Performing a backup of the ODI schemas is particularly important if the upgrade fails and corrupts the content. With a backup, you can delete the corrupted schemas and re-clone the originals to complete the upgrade.	Section 6.1.8
Install and Configure Oracle Data Integrator 12.1.3.	Install Oracle Data Integrator 12.1.3. This procedure includes creating the necessary database schemas. The Upgrade Assistant is available as part of the ODI product installation.	Section 6.2
Run Upgrade Assistant to Upgrade the Oracle Data Integrator Master Repository and Work Repository schema.	The Upgrade Assistant upgrades the Oracle Data Integrator 12.1.2 repository schemas to Oracle Data Integrator 12.1.3.	Section 6.3
Upgrade your standalone agent environment (with WebLogic domain).	After upgrading the 11g schemas, you can upgrade your standalone agent environment (with WebLogic domain).	Section 6.6
Review the configuration summary on the Configuration Summary screen, then click **Reconfig** to reconfigure the domain.	The Reconfiguration Wizard reconfigures the domain.	Section 6.6.1

5.4 Verifying the Upgrade

- To complete and verify the upgrade of the Oracle Data Integrator Java EE Agent Environment from 12.1.2 to 12.1.3, refer to Section 2.2, "Completing and Verifying the Upgrade,".

- To complete and verify the upgrade of the Oracle Data Integrator Standalone Agent Environment (no WebLogic Domain) from 12.1.2 to 12.1.3, refer to Section 3.2, "Verifying the Upgrade,".

- To complete and verify the upgrade of the Oracle Data Integrator Standalone Agent Environment (with WebLogic Domain) from 12.1.2 to 12.1.3, refer toSection 4.2, "Verifying the Upgrade,".

6

Upgrading the Existing Oracle Data Integrator 11*g* Environment

This chapter provides specific instructions for the tasks involved in upgrading the existing Oracle Data Integrator 11*g* environment to Oracle Data Integrator 12*c* (see Section 1.1 for valid 11*g* starting points).

If you are upgrading an existing 11*g* environment to a newer 11*g* version of Oracle Data Integrator, see the *Oracle Fusion Middleware Patching Guide* in the 11*g* documentation library.

This chapter contains the following sections:

- Section 6.1, "Preparing to Upgrade Oracle Data Integrator"
- Section 6.2, "Installing Oracle Fusion Middleware 12c Products"
- Section 6.3, "Upgrading the Master and Work Repository Schema"
- Section 6.4, "Upgrading the Java EE Agent Environment"
- Section 6.5, "Upgrading the Standalone Agent Environment (no WebLogic Domain)"
- Section 6.6, "Upgrading the Standalone Agent Environment (with WebLogic Domain)"
- Section 6.7, "Troubleshooting the Upgrade"

6.1 Preparing to Upgrade Oracle Data Integrator

This section contains important tasks that should be performed before you begin your upgrade to Oracle Data Integrator 12*c* (12.1.3).

- Section 6.1.1, "Planning the Upgrade"
- Section 6.1.2, "Backing Up the Existing 11g Environment"
- Section 6.1.3, "Upgrading the Database Containing the ODI Repository"
- Section 6.1.4, "Re-Associating a File-Based Security Store"
- Section 6.1.5, "Cloning Existing Master and Work Repositories"
- Section 6.1.6, "Updating the Work Repository JDBC URL"
- Section 6.1.7, "Verifying that Work Repositories are Attached to the Correct Schemas"
- Section 6.1.8, "Creating a Backup of the ODI Repositories to be Upgraded"

6.1.1 Planning the Upgrade

Before you begin, you should read *Planning an Upgrade of Oracle Fusion Middleware*, which provides a high-level overview of how to plan and prepare for the upgrade to Oracle Fusion Middleware 12*c*.

6.1.2 Backing Up the Existing 11g Environment

Before you begin, you should make a complete backup of the 11*g* environment. For details, see "Backup and Recovery Strategies for Upgrade" in the *Planning an Upgrade of Oracle Fusion Middleware*.

6.1.3 Upgrading the Database Containing the ODI Repository

The database that contains the Oracle Data Integrator repositories must be supported by Oracle Fusion Middleware 12*c*. For the latest list of certified databases, see *System Requirements and Supported Platforms for Oracle Fusion Middleware 12c (12.1.3)* on the Oracle Fusion Middleware Supported System Configurations page.

For instructions on verifying that your Oracle database meets the requirements of Oracle Fusion Middleware 11*g*, see "Upgrading and Preparing Your Oracle Databases for 12c (12.1.3)" in *Planning an Upgrade of Oracle Fusion Middleware*. Oracle recommends that you refer to your database-specific upgrade documentation for additional information.

> **Note:** If your database was supported in Oracle Data Integrator 11*g* but is no longer supported in Oracle Data Integrator 12*c*, use the ODI 11*g* version to do the following before you upgrade:
>
> 1. Export the ODI 11*g* repositories from the unsupported database systems/versions.
> 2. Import Master and Work Repositories into new repositories created with the 11*g* version into supported database systems/versions.
>
> For additional information, see "Repository-Level Export/Import" in the *Developer's Guide for Oracle Data Integrator*.

6.1.4 Re-Associating a File-Based Security Store

If you are using a file-based security store in your existing 11g environment, you must perform the following tasks before you begin the upgrade process.

Task 1 Creating 11g OPSS and IAU Schemas

Create new 11*g* Oracle Platform Security Services (OPSS) and Audit Schemas (IAU) schemas in a supported Database using the 11*g* Repository Creation Utility.

For more information about creating 11*g* schemas, see "Obtaining and Running Repository Creation Utility" in the *Oracle Fusion Middleware Repository Creation Utility User's Guide* for 11*g* Release 1 (11.1.1.7.0).

Task 2 Reassociating the 11g Security Store with the Database-Based Security Store and OPSS Schema

If you are using a file-based security store in your 11*g* environment, then reassociate the file-based store with the database-based repository and OPSS schema.

For information about reassociating OPSS schema with Database-based repository, see "Using a DB-Based OPSS Security Store" in the *Oracle Fusion Middleware Application Security Guide* in the 11*g* Release 1 (11.1.1.7.0) documentation library.

Task 3 Configuring the Audit Data Store
If the audit data store is file based, then you must enable audit loading on the database to change from storing audit records in a file to using a database audit data store.

For information about enabling audit loading, see "Configure the Audit Data Store and Bus-Stop Storage" in *Securing Applications with Oracle Platform Security Services*.

6.1.5 Cloning Existing Master and Work Repositories

Oracle recommends that you clone (copy) each original ODI Master and Work repository before beginning the upgrade process. During the Master Repository upgrade process, the Upgrade Assistant will prompt you for the location and credentials of the cloned Master Repository and Work Repositories.

The following sections provide basic schema cloning procedures for databases that are supported to host ODI repositories. For detailed information, refer to your database-specific documentation.

- Schema Cloning Process for Oracle Database
- Schema Cloning Process for MySQL Database
- Schema Cloning Process for Microsoft SQL Server Database
- Schema Cloning Process for IBM DB2 Universal Database

> **Note:**   The goal of this section is to stress the importance of creating a clone (or copy) of each of the 11*g* repositories before you begin the upgrade process. The cloning procedures documented in this section are sample procedures for each of the 11*g* supported databases. You are not restricted to using these procedures. Always use the cloning procedures that suit your specific needs.

6.1.5.1 Schema Cloning Process for Oracle Database

To clone Oracle Database schemas for ODI:

1. Export ODI 11*g* Master and Work schemas using the Oracle Export Utility (`exp`). For example:

```
exp userid=odi_master_11g/odi_master_11g file=/tmp/odi_master_11g.dmp
exp userid=odi_work_11g/odi_work_11g file=/tmp/odi_work_11g.dmp
exp userid=odi_work1_11g/odi_work1_11g file=/tmp/odi_work1_11g.dmp
```

   Export ODI 11*g* Master and Work schemas using the Datapump Utilities (`expdp`). For example:

```
expdp odi_tmp/odi_tmppwd schemas=odiw11117 dumpfile=odiw11117.dmp
```

2. Using SQL*Plus, create Master and Work clone schemas and grant connect/resource privileges. For example:

```
create user odi_master_11g_cp identified by odi_master_11g_cp;
create user odi_work_11g_cp identified by odi_work_11g_cp;
create user odi_work1_11g_cp identified by odi_work1_11g_cp;
grant connect,resource to odi_master_11g_cp, odi_work_11g_cp, odi_work1_11g_cp;
```

3. Using Oracle Import (imp), import the ODI 11*g* Master and Work schema dump into the cloned Master and Work schemas. For example:

```
imp userid='system/manager' touser=odi_master_11g_cp fromuser=odi_master_11g
file=/tmp/odi_master_11g.dmp
imp userid='system/manager' touser=odi_work_11g_cp fromuser=odi_work_11g
file=/tmp/odi_work_11g.dmp
imp userid='system/manager' touser=odi_work1_11g_cp fromuser=odi_work1_11g
file=/tmp/odi_work1_11g.dmp
```

Import ODI 11*g* Master and Work schemas using the Datapump Utilities (impdp). For example:

```
impdp ODI_TMP/ODI_TMPPWD dumpfile=odim11117 remap_tablespace=repo11117:odi11g
remap_schema=odim11117:odim1113
```

Note that with impdp it is also possible to modify the schema name and tablespace for data storage. The remap_xx parameters are optional.

6.1.5.2 Schema Cloning Process for MySQL Database

The following steps can be used to clone MySQL database schemas:

1. Export the ODI 11*g* Master and Work schemas using mysqldump. For example:

```
mysqldump -h localhost -u root -p DEV_ODI_REPO > /scratch/dump.sql
```

2. Restore the ODI schema into a new schema using mysql. For example:

First, create a cloned schema:

```
mysql -h localhost -u root -p
create schema NEW_ODI_REPO default character set=utf8 default collate=utf8_bin;
```

Then, import the ODI schema into the cloned schema.

```
mysql -h localhost -u root -p NEW_ODI_REPO < /scratch/dump.sql
```

3. Create a login for the cloned schema using mysql. For example:

```
mysql -h localhost -u root -p
grant all on NEW_ODI_REPO.* to NEW_ODI_REPO1@'localhost' identified by
'password';
grant process on *.* to NEW_ODI_REPO1@'localhost'
```

6.1.5.3 Schema Cloning Process for Microsoft SQL Server Database

To clone Microsoft SQL 2005/2008 database schemas:

1. Export the ODI 11*g* Master and Work schemas using SQL Management Studio.

Example:

```
BACKUP DATABASE [odi_11g] TO DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Backup\odi_11g.bak' WITH INIT, NOSKIP;
```

2. Restore Master and Work schemas into the new database using SQL Management Studio.

Using SQL Management Studio Express perform the following:

1. Restore the Master and Work schemas.

2. Print logical names of files used to store the database.

3. Move the files used to store database.

Example:

```
RESTORE DATABASE [odi_11g_cp] FROM DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Backup\odi_11g.bak'
WITH FILE = 1, MOVE N'odi_11g' TO N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\odi_11g_cp.mdf',
MOVE N'odi_11g_log' TO N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\odi_11g_cp_log.ldf', NOUNLOAD;
go
```

3. Create login and user for cloned Master and Work schemas using SQL Management Studio.

Using SQL Management Studio Express, create logins and users to access cloned Master and Work schemas. Be sure to select the correct database instance in SQL Management Studio Express, as these commands are applied to the selected database instance.

Example:

```
create login odi_11g_cp with password=N'odi_11g_cp',
default_database=odi_11g_cp,  check_expiration = off, check_policy = off;
go
USE odi_11g_cp
go
create user odi_11g_cp  for login odi_11g_cp;
go
USE odi_11g_cp
go
```

4. To move the old schema to the new schema location, run the following SQL script:

NOTE: In the example below, the old schema name is `odi_11g` and the new schema name is `odi_11g_cp`.

```
CREATE SCHEMA [odi_11g_cp] AUTHORIZATION odi_11g_cp
go
.
DECLARE @OldSchema AS varchar(255)
DECLARE @NewSchema AS varchar(255)
.
SET @OldSchema = 'odi_11g'
SET @NewSchema = 'odi_11g_cp'
.
DECLARE @sql AS varchar(MAX)
SET @sql = CHAR(13) + CHAR(10)
.
SELECT @sql = @sql + 'ALTER SCHEMA [' + @NewSchema + '] TRANSFER [' +
TABLE_SCHEMA + '].[' + TABLE_NAME + ']' + CHAR(13) + CHAR(10)
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = @OldSchema
.
EXEC (@sql)
go
```

5. To finalize the schema move, run the following SQL query:

```
DROP SCHEMA [odi_11g]
go
drop user odi_11g;
go
```

```
alter user odi_11g_cp with default_schema = odi_11g_cp;
go
grant create table, create view, create procedure,create function to
odi_11g_cp;
go
```

6.1.5.4 Schema Cloning Process for IBM DB2 Universal Database

Chose one of the following procedures to clone IBM's DB2 Universal Database schemas:

- Same Host Cloning Process for ODI 11g Master and Work Schemas
- Different Host Cloning Process for ODI 11g Master and Work Schemas

> **Note:** The Page size for database has to be 32768 (32k) and operating system users `ODI_MASTER_11G_CP` and `ODI_WORK_11G_CP` have to be created manually.

6.1.5.4.1 Same Host Cloning Process for ODI 11g Master and Work Schemas Use the following steps to clone IBM DB2 schemas on the same host or platform:

1. Create DB2 Database using Command Line Processor.

   Example:

   ```
   db2 CREATE DATABASE ODI12 AUTOMATIC STORAGE YES  ON 'C:\' DBPATH ON 'C:\' USING
   CODESET IBM-1252 TERRITORY US COLLATE USING SYSTEM PAGESIZE 32768
   ```

2. Copy ODI 11*g* Master and Work schemas using DB2 Database Movement Tool to new schema.

   Master Schema Example:

   ```
   db2move ODI11G COPY -sn odi_master_11g -co TARGET_DB ODI11GCP USER db2admin
   USING welcome SCHEMA_MAP ((odi_master_11g,odi_master_11g_cp)) TABLESPACE_MAP
   ((USERSPACE1,USERSPACE1),SYS_ANY) owner odi_master_11g_cp
   ```

   Work Schema Example:

   ```
   db2move ODI11G COPY -sn odi_work_11g -co TARGET_DB ODI11GCP USER db2admin USING
   welcome SCHEMA_MAP ((odi_work_11g,odi_work_11g_cp)) TABLESPACE_MAP
   ((USERSPACE1,USERSPACE1),SYS_ANY) owner odi_work_11g_cp
   ```

6.1.5.4.2 Different Host Cloning Process for ODI 11g Master and Work Schemas Use the following steps to clone IBM DB2 schemas on different hosts or platforms:

1. Export DDL and Data from Master and Work schemas using DB2 Database Movement Tool and DDL Extracting Tool.

   DB2 Database Movement Tool produces PC/IXF files with data and `db2move.lst` file with list of tables, Files are produced in the folder where the tool was called. The DDL Extracting Tool produces `db2master.sql` and `db2work.sql` with SQL queries to recreate database structure.

   Example:

   ```
   db2move ODI11G export -sn odi_master_11g,odi_work_11g
   db2look -d ODI11G -z odi_master_11g -e -o c:/db2master.sql
   db2look -d ODI11G -z odi_work_11g -e -o c:/db2work.sql
   ```

2. Transfer exported files to new location.

1. Ensure that the PC/IXF files were transferred in binary mode, and that the `db2move.1st` file and the `db2master.sql` and `db2work.sql` files were transferred in ASCII mode.

2. Place the PC/IXF files where the DB2 Database Movement Tools is located.

3. Create DB2 database using Command Line Processor.

   Example:

   ```
   db2 CREATE DATABASE ODI11G AUTOMATIC STORAGE YES  ON 'C:\' DBPATH ON 'C:\'
   USING CODESET IBM-1252 TERRITORY US COLLATE USING SYSTEM PAGESIZE 32768
   ```

4. Import the exported DDL to the new database using the Command Line Processor.

   Example:

   ```
   db2 -tvf c:/db2backup/db2master.sql
   db2 -tvf c:/db2backup/db2work.sql
   ```

5. Import exported data to new database using DB2 Database Movement Tool.

   Example:

   ```
   db2move ODI11G load
   ```

6. Verify that cloned schemas are intact; some tables may be in "check pending" state (because of check constraint).

   Use command `set integrity` to move to the normal state.

   Example:

   ```
   db2 set integrity for table_name immediate checked
   ```

6.1.6 Updating the Work Repository JDBC URL

When you clone an Oracle Data Integrator repository, the cloned master repository has embedded in it the work repository connection details of the original source repository. When cloning, always connect to the Master Repository of the cloned repository. Open each work repository and change the schema name and password and/or the JDBC URL

Without these procedures, the original schemas may be upgraded unintentionally.

Example 1: A source repository schema named CUST1, a work repository schema named CUST1 (workrep1), and a work repository schema named CUST2 (workrep2) are all being restored to the same database instance.

To update the work repository JDBC URL:

1. Run a simple clone using Oracle Datapump by exporting the schemas CUST1 and CUST2 using the following commands:

   ```
   Expdb System/Password Directory=Dumpdir Schemas=cust1,cust2
   Dumpfile=Filename.dmp
   ```

2. Clone CUST1 and CUST2 to create new schemas NEWCUST1, NEWCUST2:

   ```
   Impdp System/Password Directory=Dumpdir Schemas=cust1,cust2 Remap_
   schemas=cust1:newcust1, cust2:newcust2 Dumpfile=Filename.dmp
   ```

3. Connect to the master repository only of NEWCUST1.

4. Open `workrep1` and update the `schema_name` from `CUST1` to `NEWCUST1`.

5. Open `workrep2` and update the `schema_name` from `CUST2` to `NEWCUST2`.

Example 2: The source repository schema named CUST1, a work repository schema named CUST1 (workrep1),and a work repository schema named CUST2 (workrep2). In this example there is no renaming of schemas. All schemas are restored to a different database instance.

To restore the schemas to new database instance:

1. Run the following command:

   ```
   Impdp System/Password Directory=Dumpdie Schemas=cust1,cust2
   Dumpfile=Filename.dmp
   ```

2. Connect to the master repository only of CUST1.

3. Open `workrep1` and update the JDBC URL to point to the new database instance.

4. Open `workrep2` and update the JDBC URL to point to the new database instance.

If you restore schemas to a new database instance and also rename them, update both the `schema_names` and `JDBC_URL`.

6.1.7 Verifying that Work Repositories are Attached to the Correct Schemas

After cloning the repositories, you should check the repository connections to see verify that the cloned master repository points to the correct cloned work repository schema. The upgrade process retrieves the work repository information from its master repository; in order to have a successful upgrade of the work repositories, you must ensure that the repositories are attached to the correct schema and host before you upgrade.

To verify this:

> **Note:** The documentation links in this section refer to 11*g* Release 1 (11.1.1.7.0) documentation.

1. Connect to the ODI master repository using your existing ODI client (pre-upgraded version).

   For information, see "Connecting to the Master Repository" in the *Developer's Guide for Oracle Data Integrator*.

2. Validate that the work repositories are attached to the correct work repository schema and host.

   For more information, see: "Connecting to a Work Repository" and "Attaching and Deleting a Work Repository" in the *Developer's Guide for Oracle Data Integrator*.

6.1.8 Creating a Backup of the ODI Repositories to be Upgraded

Oracle recommends that you create a backup for each ODI Master and Work repository. The backup enables you to restore to a pre-upgrade state if necessary. For more information, see "Backup Strategies for Upgrade," in *Planning an Upgrade of Oracle Fusion Middleware*.

If an upgrade fails you also need to restore the contents of `schema_version_registry` table back to its pre-upgrade state; therefore, the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table must be included as part of any backup.

The Upgrade Assistant Prerequisite screen prompts you to indicate whether the backup of the ODI repositories has been completed. It is important to note, however, that the Upgrade Assistant will not validate that a backup has been created.

> **Caution:** This is a critical step of the upgrade process; especially if the repositories were not cloned. In the event that upgrade results are unsatisfactory, the repository is locked and unable to be used.
>
> Having a backup copy of the ODI repositories not only ensures that you will not lose important data, but it is the only way to restore after a failed upgrade; there is no capability to restart the upgrade from a failed state.
>
> For more information on creating a backup, refer to your database backup and recovery documentation.

6.2 Installing Oracle Fusion Middleware 12c Products

Before beginning your upgrade, use the Oracle Universal Installer to install the 12*c* version of your Oracle Fusion Middleware products. Follow the instructions in the following section in *Installing and Configuring Oracle Data Integrator*:

1. "Planning Your Oracle Data Integrator Installation" to understand important information about the 12*c* Java EE agent installation topology.

   This chapter introduces some important 12*c* concepts and also provides information for where to obtain the necessary product distributions. Note that for Java EE agent environment, Oracle Fusion Middleware Infrastructure is also required as a prerequisite for installing Oracle Data Integrator.

2. "Installing Oracle Data Integrator" to install Oracle Data Integrator for your environment.

   Be sure to follow the installation instructions for your particular environment.

3. "Creating the Oracle Data Integrator Master and Work Repository Schema" to run RCU and create the schemas that are new in 12*c* (for example, the Service Table schema).

You do not need to run the configuration wizard to configure a new domain; the Upgrade Assistant will take care of this for you in an upgrade scenario.

6.3 Upgrading the Master and Work Repository Schema

Follow the instructions in this section to use the Upgrade Assistant to upgrade your Master and Work Repository schema.

- Section 6.3.1, "Stopping the Servers and Processes"
- Section 6.3.2, "Starting the Upgrade Assistant"
- Section 6.3.3, "Navigating the Upgrade Assistant Screens"

6.3.1 Stopping the Servers and Processes

Before running the Upgrade Assistant to upgrade your schemas, make sure all servers and processes in the domain are stopped.

For more information, see "Starting and Stopping Oracle Fusion Middleware" in *Administering Oracle Fusion Middleware*.

6.3.2 Starting the Upgrade Assistant

To start the Upgrade Assistant, go to the *ORACLE_HOME*/oracle_common/upgrade/bin directory, and enter the following command:

On UNIX operating systems:

```
./ua
```

On Windows operating systems:

```
ua.bat
```

6.3.3 Navigating the Upgrade Assistant Screens

After the Upgrade Assistant is started (Section 6.3.2), follow the instructions in this section to navigate through the screens in the Upgrade Assistant to upgrade your product schemas.

Note: The Upgrade Assistant uses the data and structure of the ODI Master repository to determine if a repository has already been upgraded. The Upgrade Assistant will return a message stating that the repository has already been upgraded if the following conditions exist:

- The schema version registry has valid state and version for the repository

- The repository is 12*c*

- The version of the repository is equal to or greater than the version of ODI SDK used by the Upgrade Assistant

To debug or view the repository catalog information, use the following query on the schema_version_registry table, which is stored in the Admin user (not in the ODI schema/repository).

In Oracle databases, the name of this table is SYSTEM.SCHEMA_VERSION_REGISTRY$ and it is stored in the SYSTEM schema. There is also a view named SYSTEM.SCHEMA_VERSION_REGISTRY and a public synonym SCHEMA_VERSION_REGISTRY that points to the view:

```
SELECT COMP_ID,COMP_NAME,MRC_NAME,OWNER,VERSION,STATUS,UPGRADED
FROM schema_version_registry;
```

On DB2/400 operating systems, the Admin user is QSECOFR, and the schema_version_registry table is located in the schema 'NULLID'.

Rows with the component "ODI" are used to track ODI repositories.

- Task 1, "Introducing the Upgrade Assistant"

- Task 2, "Selecting an Upgrade Operation"

- Task 3, "Selecting Component Schemas"

- Task 4, "Specifying the Weblogic Directory"

- Task 5, "Verifying Prerequisites"

- Task 6, "Specifying Database and Schema Credentials"
- Task 7, "Selecting the ODI Upgrade Option"
- Task 8, "Specifying Supervisor Account Credentials"
- Task 9, "Selecting the ODI Upgrade Key"
- Task 10, "Completing the Upgrade Validation"
- Task 11, "Beginning and Completing the Upgrade"
- Task 12, "Verifying the Schema Version"

Task 1 Introducing the Upgrade Assistant

The Welcome screen contains important reminders to consider before proceeding with your upgrade. Make sure you read these and verify that you are ready to proceed.

Task 2 Selecting an Upgrade Operation

Select **Schemas**. The Upgrade Assistant will list the schemas available for upgrade on the next screen.

The title of the screen will change to "Schemas" when you select **Schemas**.

Task 3 Selecting Component Schemas

The Available Components screen lists the schemas that are available for upgrade.

Select Oracle Data Integrator to upgrade the Master and Work Repository schema. Note that any dependent schema will be automatically selected.
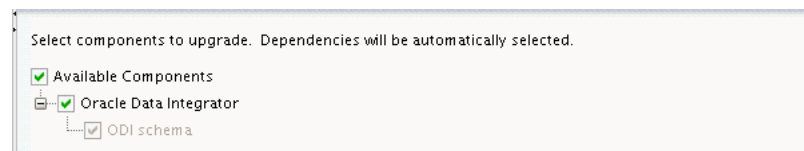
> **Tip:** More information about this screen can be found in Available Components in *Upgrading with the Upgrade Assistant*.

Task 4 Specifying the Weblogic Directory

In the **Domain Directory** screen enter the WebLogic directory for the domain to be upgrade.

Task 5 Verifying Prerequisites

The Prerequisites screen lists items that you must check and verify before you can proceed.

You must check the boxes before you can continue. The Upgrade Assistant will not verify that the prerequisites have been met.

Task 6 Specifying Database and Schema Credentials

On the Select Schemas screen, enter the connection credentials for the database containing the schema you want to upgrade. Click **Connect** to connect to your database.

Then, specify the schema user name and password for the Master and Work Repository.

> **Tip:** More information about this screen can be found in Select Schemas in *Upgrading with the Upgrade Assistant*.

Task 7 Selecting the ODI Upgrade Option

On the ODI Options screen, select all of the options on this screen.
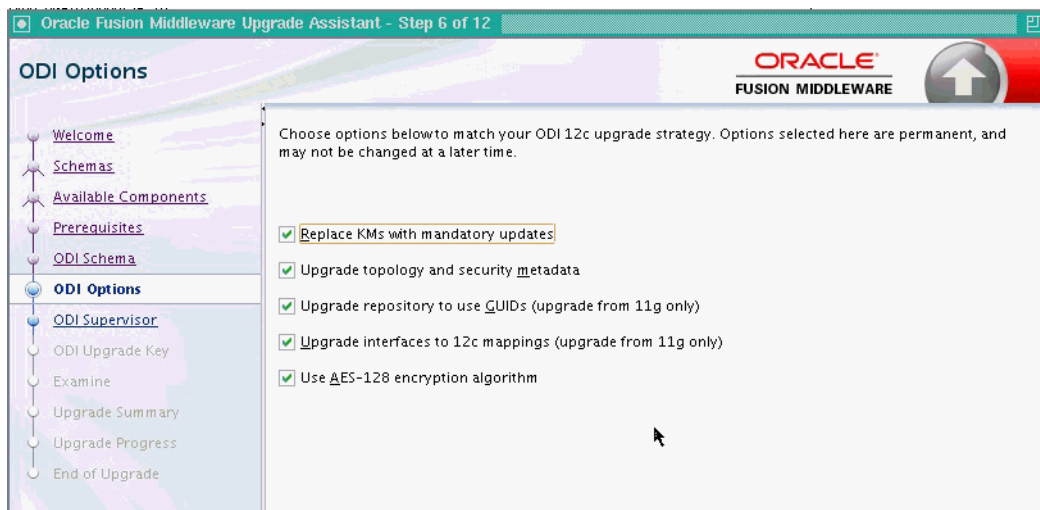
Each option is described in the following table.

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. If you do not wish to go through the ODI upgrade, see "Using Compatibility Mode" in *Developing Integration Projects with Oracle Data Integrator*.
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 *Developing Integration Projects with Oracle Data Integrator*.

Option	Description
Upgrade repository to use GUIDs	This selection sets the repository to 12*c* full mode. All objects are referenced using 12*c* GUID rather than the internal ID.
	You should leave this option checked in order to take advantage of the truly universally unique identification scheme in Oracle data Integrator 12*c*.
	Note: This option only applies when upgrading from 11g repository. For the 12.1.2 repository, the option is ignored, and the repository stays in the same mode as before the upgrade.
	If you have custom Knowledge Modules and procedures that use `odiRef` substitution APIs, which take internal identifiers as parameters such as `getFlexFieldValue()`, you may choose to not select this option, leaving your repository in "11*g* compatibility mode." Scenarios generated from objects using such Knowledge Modules and procedures continue to work in "11*g* compatibility mode" but will not work in 12*c* full mode. "11g compatibility mode" can be used to smoothly transition the custom Knowledge Modules and procedures to use the new `odiRef` substitution APIs (the ones that take GUIDs as parameters). After all custom Knowledge Modules and procedures have been modified to use the new `odiRef` substitution APIs, the repository can be switched to 12*c* full mode.
	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 greyed out.
Upgrade interfaces to use 12c mappings - losing 11g SDK compatibility	This selection converts all 11*g* interfaces to 12*c* mappings. Once converted to 12*c* mappings, all of the existing scenarios must be regenerated before use. There is no ability to use existing 11*g* SDK applications; they must be upgraded to use the 12*c* SDK.
	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.
	If this option is not selected, some conversion to 12*c* mappings are performed but the resulting mappings are left in "11g compatibility mode."After these interfaces are modified using the 11*g* SDK, they can then be converted to 12*c* mappings using the ODI Studio graphical interface or the 12*c* SDK.
	Oracle recommends leaving this option checked, unless you have significant amount of Java code that uses the 11*g* SDK to read or update existing interfaces. This option only applies when upgrading from 11g repository. For 12.1.2 repository the option is ignored, and the repository stays in the same mode as before upgrade.
	NOTE: In order for this migration to work properly, all interfaces in 11*g* repository must be valid (for example, they should not return any errors when validating from 11*g* Studio, for example). If an 11*g* interface is not valid, the Upgrade Assistant will try to migrate it into a 12*c* 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 11*g* repository as valid to start with.
	The upgrade process does not stop even if some 11*g* interfaces fail during the migration; the upgrade will continue until all interfaces are processed.

Option	Description
Use AES-128 encryption algorithm	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

> **Tip:** For more information about this screen, see "ODI Options" in *Upgrading with the Upgrade Assistant*.

Table 6–1 contains descriptions of the combinations of options that may or may not be selected on this screen.

Table 6–1   Possible ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	
X	X	X	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.
X		X	With this combination, the repository stays in ID compatibility mode, which means that odiRef 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 odiRef APIs with numeric IDs as arguments.
		X	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.
X	X		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.

Table 6–1   (Cont.) Possible ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	
X			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.
			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 6–2 describes the Invalid ODI Options screen combinations.

Table 6–2   Invalid ODI Options screen combinations

Replace Knowledge Modules with mandatory updates	Upgrade repository to use GUIDs	Upgrade interfaces to use 12c mappings	
	X		With this selection, most Knowledge Modules will not function correctly since they will be using deprecated odiRef APIs, which use legacy numeric IDs.
	X	X	With this combination, most Knowledge Modules will not function correctly since they will be using deprecated odiRef 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 affect 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 affect on the other options.

Task 8 Specifying Supervisor Account Credentials

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

The installed Supervisor account is SUPERVISOR (all CAPS). Check with your ODI administrator for proper account name and password.

> **Tip:** For more information about this screen, see "ODI Supervisor" in *Upgrading with the Upgrade Assistant*.

Task 9 Selecting the ODI Upgrade Key

The upgrade key is used to convert 11*g* IDs for repository objects into unique GUIDs.

> **Note:** This task does not apply to an upgrade from ODI 12.1.2. to ODI 12.1.3.

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 12*c*, 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 of the fact that 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 product with 1001 as the ID in the 11*g* 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 12*c* 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 11*g* 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.

> **Tip:** For more information, see "ODI Upgrade Key" in *Upgrading with the Upgrade Assistant*.

Task 10 Completing the Upgrade Validation

On the Examine screen, the Upgrade Assistant performs a series of validations before upgrading the selected components. Ensure that all validations have succeeded.

> **Tip:** More information about the options on this screen can be found in Examine in *Upgrading with the Upgrade Assistant*.

Task 11 Beginning and Completing the Upgrade

Click **Upgrade** on the Upgrade Summary screen to begin the upgrade. The Upgrade Progress screens shows information about the progress of the upgrade, and the Upgrade Success screen summarizes the upgrade.

Task 12 Verifying the Schema Version

To verify that the schema has been upgraded, run SQL*Plus from the database host and use the following command:

```
select owner, version, status from schema_version_registry where owner = 'prefix_
ODI_REPO';
```

Replace `prefix` with the custom prefix of your repository schema created in RCU. Below is an example:

```
select owner, version, status from schema_version_registry where owner = DEV1213_
ODI_REPO
OWNER                           VERSION                        STATUS
----------------------------    ---------------------------    -----------
DEV1213_ODI_REPO                12.1.3.0.0                     VALID
```

In the output, verify that the schema version number is "12.1.3.0.0" in the "VERSION" column.

6.4 Upgrading the Java EE Agent Environment

After the repository schemas are upgraded, run the reconfiguration wizard to complete the upgrade of the Java EE agent environment.

- Section 6.4.1, "Starting the Reconfiguration Wizard"
- Section 6.4.2, "Navigating the Reconfiguration Wizard Screens to Reconfigure the Domain"

6.4.1 Starting the Reconfiguration Wizard

To start the reconfiguration wizard:

1. Log in to the system where the 12*c* Oracle Data Integrator software was installed.

2. Go to the following directory location:

   On UNIX operating systems:

   ```
   ORACLE_HOME/oracle_common/common/bin
   ```

On Windows operating systems:

```
ORACLE_HOME\oracle_common\common\bin
```

where `ORACLE_HOME` is the location where Oracle Data Integrator is installed.

3. Start the reconfiguration wizard.

On UNIX operating systems:

```
./reconfig.sh -log=log_file
```

On Windows operating systems:

```
reconfig.cmd -log=log_file
```

Specify the full path and file name in place of `log_file`; creating this log file can be very helpful if you need to troubleshoot the reconfiguration process.

Note: When you run the reconfiguration wizard, the following error message might be displayed 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
```

6.4.2 Navigating the Reconfiguration Wizard Screens to Reconfigure the Domain

Follow the instructions in this section to navigate through the screens in the reconfiguration wizard.

- Task 1, "Selecting the 11g Domain"

- Task 2, "Viewing the Reconfiguration Setup Progress"

- Task 3, "Selecting the Domain Mode and JDK"

- Task 4, "Configuring JDBC Data Sources"

- Task 5, "Testing the JDBC Data Sources"

- Task 6, "Selecting the Database Configuration Type"

- Task 7, "Configuring JDBC Component Schema"

- Task 8, "Configuring Node Manager"

- Task 9, "Configuring Credentials"

- Task 10, "Configuring the Keystore"

- Task 11, "Configuring the ODI Server"

- Task 12, "Completing Domain Reconfiguration"

Task 1 Selecting the 11g Domain

Use the Select Domain screen to specify the full path to the domain location of your 11*g* Oracle Data Integrator domain. You can also click **Browse** and use the file manager window to help you select the domain location.

Task 2 Viewing the Reconfiguration Setup Progress

The Reconfiguration Setup Progress screen shows you the progress of the application of reconfiguration templates.

Task 3 Selecting the Domain Mode and JDK

Domain mode cannot be changed.

Select the JDK to use in the domain or click **Browse** to navigate to the JDK you want to use.

> **Note:**   Oracle Fusion Middleware 12c requires Java SE 7. For more information, see "Verifying Certification and System Requirements" in *Planning an Installation of Oracle Fusion Middleware*.

Task 4 Configuring JDBC Data Sources

The JDBC Data Sources screen is displayed if you created custom data sources for a database-based OPSS security store or Audit Data store in 11*g*.

Use this screen to configure the JDBC data sources defined in your domain source.

For information about the fields on this page, click Help, or refer to "JDBC Data Sources" in *Upgrading Oracle WebLogic Server*.

Task 5 Testing the JDBC Data Sources

Use the JDBC Data Sources Test screen to test the data source connections you configured.

For information about the fields on this page, click Help, or refer to "JDBC Data Sources Test" in *Upgrading Oracle WebLogic Server*.

Task 6 Selecting the Database Configuration Type

Select **Manual Configuration**, and click **Next**.

If not already selected, select **RCU Data**.

Provide the database credentials to retrieve the schema information for the upgradable schemas that are included in the domain. If you select this option, the fields on this screen are activated. Fill in each field using the connection information that you specified for the STB component in the Repository Creation Utility (RCU).

> **Note:**   If in the earlier installation you created an ODI repository using ODI Studio you do not have the STB schema created. Go to "Starting the Repository Creation Utility" in *Installing and Configuring Oracle Data Integrator*. Create the dependent schema for ODI by first selecting the ODI schema in RCU so that all the dependent schemas get selected. Deselect the ODI schema and provide the custom ODI repository connectivity from this step.

When you have provided the connection information, click **Get RCU Configuration** to retrieve the schema information.

For more information, click **Help**, or refer to "Database Configuration Type" in *Upgrading Oracle WebLogic Server*.

Task 7 Configuring JDBC Component Schema

Specify the data source settings for each of the schemas listed on the JDBC Component Schema screen by selecting the check box adjacent to each schema name.

> **Note:**   You must specify the 11*g* schema details for those schemas that you upgraded in Section 6.3. For the others, specify the 12*c* schema details.

For information about the fields on this page, click **Help**, or refer to "JDBC Component Schema" in *Upgrading Oracle WebLogic Server*.

Task 8 Configuring Node Manager

The Node Manager screen is displayed only if the domain you are reconfiguring is currently using a Per Domain Default Location Node Manager.

Select **Migrate Existing Configuration** and provide the location of the per domain default location.

Enable **Apply Oracle Recommended Defaults**.

Provide Node Manager Credentials. This is a new User that is being created to administer the Node Manager. The password is required during start-up, for any components now handled by the Node Manager (including OHS).

Note: When upgrading a domain and changing from a per-host Node Manager configuration to a per-domain Node Manager configuration, if you are using custom scripts to start and stop the WebLogic Server environment, you must manually update the scripts to change the Node Manager home location to the new domain-based location.

For more information on configuring the Node Manager, see "Default Node Manager Configuration" in *Administering Node Manager for Oracle WebLogic Server*.

Task 9 Configuring Credentials

Use this screen to provide credentials for each key in the domain. Credentials that have been previously defined in the domain are already included in the table.

For more information on credentials, see "Understanding Identities, Policies, Credentials, Keys, Certificates, and Auditing" in *Securing Applications with Oracle Platform Security Services*.

Task 10 Configuring the Keystore

Use the Keystore screen to specify the path to the trusted certificate for each keystore, the path to each keystore's private key, the password for the private key, and the path to the Identity Certificate for the private key.

When you click in the Trusted Certificate , Private Key, or Identity Certificate fields, a browse icon appears on the right of the field. Click this icon to browse for the appropriate file.

Task 11 Configuring the ODI Server

Use the ODI Server Configuration screen to configure the colocated ODI agents in your domain.

Table 6–3   Fields for ODI Server Configuration screen

Field	Description
System Component	From this drop-down list, select the ODI agent to configure.
Server Listen Address	Enter the host name or IP address of the host systyem on which this agent resides. Do not use **localhost**.
Server Listen port	Enter the listen port to be used for the ODI agent.
Supervisor Name	Enter the ODI username that has Supervisor privileges.
Supervisor Password	Enter the Supervisor user's password.
Preferred Data Source	From this drop-down list, select the data source to be used for the selected ODI agent.

Task 12 Completing Domain Reconfiguration

Review the configuration summary on the Configuration Summary screen, then click **Reconfig** to reconfigure the domain.

When reconfiguration is finished, click **Finish** to dismiss the reconfiguration wizard.

6.5 Upgrading the Standalone Agent Environment (no WebLogic Domain)

After the repository schemas are upgraded, follow the instructions in this section to upgrade the standalone agent (no WebLogic domain) environment.

- Section 6.5.1, "Starting the Upgrade Assistant"
- Section 6.5.2, "Upgrading the Standalone System Component Configurations"

6.5.1 Starting the Upgrade Assistant

To start the Upgrade Assistant, go to the *ORACLE_HOME*/oracle_ common/upgrade/bin directory, and enter the following command:

On UNIX operating systems:

```
./ua
```

On Windows operating systems:

```
ua.bat
```

6.5.2 Upgrading the Standalone System Component Configurations

After the Upgrade Assistant is started (Section 6.5.1), follow the instructions in this section to navigate through the Upgrade Assistant screens and upgrade the standalone system component configurations.

- Task 1, "Introducing the Upgrade Assistant"
- Task 2, "Selecting an Upgrade Operation"
- Task 3, "Viewing Components to be Upgraded"
- Task 4, "Verifying Prerequisites"
- Task 5, "Specifying 11g Instance Directories"
- Task 6, "Creating a Node Manager"
- Task 7, "Completing the Upgrade Validation"

-

Task 1 Introducing the Upgrade Assistant

The Welcome screen contains important reminders to consider before proceeding with your upgrade. Make sure you read these and verify that you are ready to proceed.

Task 2 Selecting an Upgrade Operation

Beginning with 12c (12.1.2), standalone system components will have their own standalone domain. When you are upgrading your 11*g* standalone system components (which had no previous domain associations), you must first create a new standalone domain for your system components.

On the Standalone Components screen, select **Standalone System Component Configurations**, then select **Create a New Domain**.

In the Domain Directory field, specify the full path of the Domain you are creating. It is recommended 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.
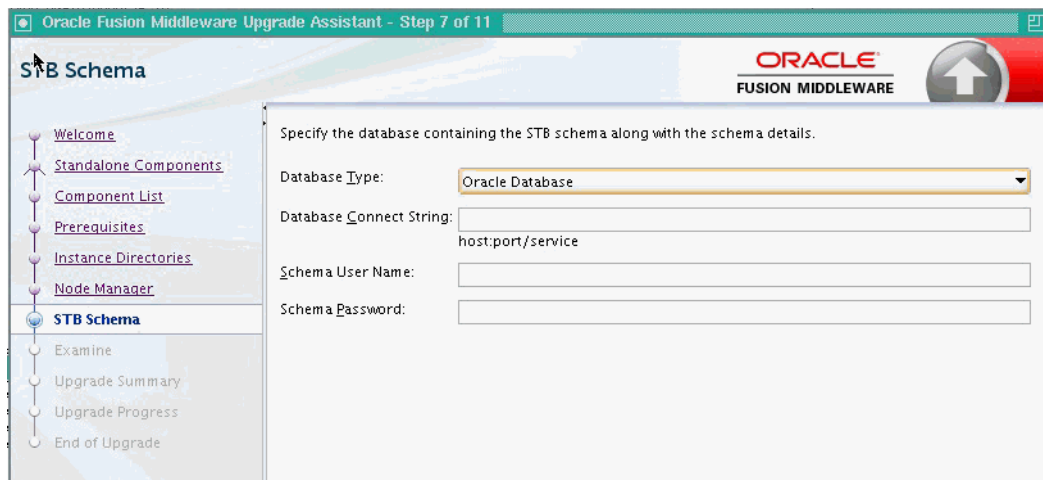
> **Tip:** The **Update an Existing Domain** option might be used in a situation where an upgrade was already performed and the domain created, either from another system component upgrade, or from a partial previous Oracle Data Integrator upgrade. These are examples of a situation where you would not need to create a new domain.
>
> More information about this screen can be found in Standalone Components in *Upgrading with the Upgrade Assistant*.

Task 3 Viewing Components to be Upgraded

The Component List screen lists the components that will be upgraded:

- System Components Infrastructure
- Oracle Data Integrator

Task 4 Verifying Prerequisites

The Prerequisites screen lists items that you must check and verify before you can proceed.

You must check the boxes before you can continue. The Upgrade Assistant will not verify that the prerequisites have been met.

Task 5 Specifying 11g Instance Directories

On the Instance Directories screen, specify the location of one or more 11*g* Oracle instance directories to be upgraded.

Task 6 Creating a Node Manager

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.

> **Tip:** More information about the options on this screen can be found in Node Manager in *Upgrading with the Upgrade Assistant*.

Task 7 Completing the Upgrade Validation

Ensure that all validations have succeeded.

> **Tip:** More information about the options on this screen can be found in Examine in *Upgrading with the Upgrade Assistant*.

Task 8 Beginning and Completing the Upgrade

Click **Upgrade** on the Upgrade Summary screen to begin the upgrade. The Upgrade Progress screens shows information about the progress of the upgrade, and the Upgrade Success screen summarizes the upgrade.

6.6 Upgrading the Standalone Agent Environment (with WebLogic Domain)

After your repository schemas are upgraded, follow the instructions in this section to upgrade the standalone agent (with WebLogic domain) environment.

- Section 6.6.1, "Reconfiguring the Domain Using the Reconfiguration Wizard"
- Section 6.6.2, "Starting the Upgrade Assistant"
- Section 6.6.3, "Upgrading the WebLogic Domain Component Configurations"

6.6.1 Reconfiguring the Domain Using the Reconfiguration Wizard

Follow the instructions in Section 6.4 to reconfigure the 11*g* or 12.1.2 domain using the reconfiguration wizard.

6.6.2 Starting the Upgrade Assistant

To start the Upgrade Assistant, go to the `ORACLE_HOME`/oracle_ common/upgrade/bin directory, and enter the following command:

On UNIX operating systems:

```
./ua
```

On Windows operating systems:

```
ua.bat
```

6.6.3 Upgrading the WebLogic Domain Component Configurations

After the upgrade assistant is started (Section 6.6.2), follow the instructions in this section to navigate through the Upgrade Assistant screens and upgrade the WebLogic domain component configurations.
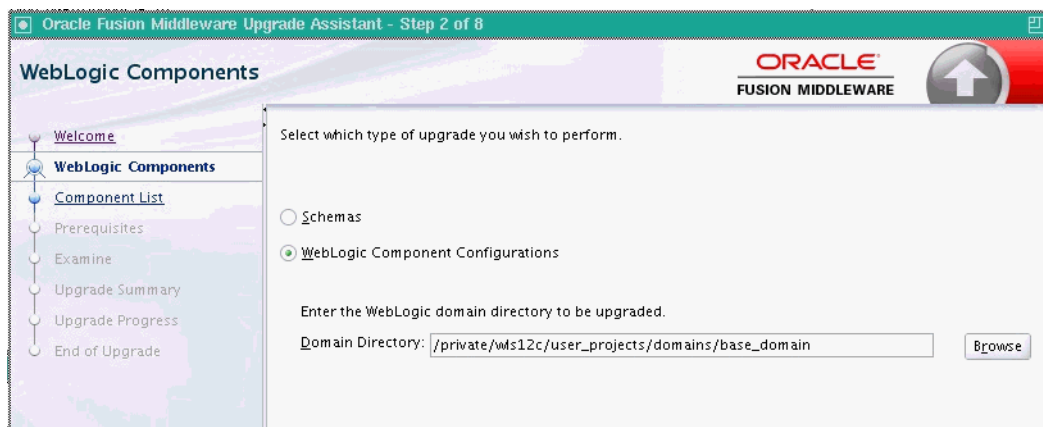
- Task 1, "Introducing the Upgrade Assistant"

- Task 2, "Selecting an Upgrade Operation"

- Task 3, "Viewing Components to be Upgraded"

- Task 4, "Verifying Prerequisites"

- Task 5, "Completing the Upgrade Validation"

- Task 6, "Beginning and Completing the Upgrade"

Task 1 Introducing the Upgrade Assistant

The Welcome screen contains important reminders to consider before proceeding with your upgrade. Make sure you read these and verify that you are ready to proceed.

Task 2 Selecting an Upgrade Operation

Select the **WebLogic Component Configurations** option to upgrade component configurations for your managed WebLogic Server domain. You will be prompted to supply the connection details required to connect to the WebLogic Administration Server that is managing the domain.

> **Tip:**   More information about this screen can be found in WebLogic Component Configurations in *Upgrading with the Upgrade Assistant*.

Task 3 Viewing Components to be Upgraded

The Component List screen lists the components that will be upgraded.

Task 4 Verifying Prerequisites

The Prerequisites screen lists items that you must check and verify before you can proceed.

You must check the boxes before you can continue. The Upgrade Assistant will not verify that the prerequisites have been met.

Task 5 Completing the Upgrade Validation

On the Examine screen, the Upgrade Assistant performs a series of validations before upgrading the selected components. Ensure that all validations have succeeded.

> **Tip:** More information about the options on this screen can be found in Examine in *Upgrading with the Upgrade Assistant*.

Task 6 Beginning and Completing the Upgrade

Click **Upgrade** on the Upgrade Summary screen to begin the upgrade. The Upgrade Progress screens shows information about the progress of the upgrade, and the Upgrade Success screen summarizes the upgrade.

6.7 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*.

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 will have no effect on other conversions.

Do the following when you encounter such interface upgrade errors:

- If the interfaces highlighted by UA log file are not used and can be ignored, then you can continue to use the upgraded 12c repository

- If the interface should be converted to a valid 12c mapping, each erroneous interface needs to be fixed in the 11g environment and re-converted using one of these two methods:

  - Export only those interfaces, import them into upgraded 12c repository. The import process will internally upgrade the 11g interfaces into 12c mappings.

  - Repeat the entire upgrade with fixed interfaces. This is recommended if the number of erroneous interfaces is large.

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 re-converted using one of these two methods:

– Export only those interfaces, import them into upgraded 12c repository. The import process internally upgrades the 11g interfaces into 12c mappings. For more information on the importing an interface into 12c repository, see *Administering Oracle Data Integrator*.

– Repeat the entire upgrade with fixed interfaces. This is recommended if the number of erroneous interfaces is large.

A

Appendix

This chapter lists the Substitution API that are impacted by GUID changes. Upgrading to GUIDs impacts some Substitution API methods which use internal IDs as parameters.

Table A–1 lists the new Substitution API methods.

Table A–1   The 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("ID_TABLE_PK")	getFK("GUID_TABLE_PK") /* Return the Primary Table GUID */
	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("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("DEST_I_ CONNECT")	getInfo("DEST_ CONNECT_ GUID") /* Return the Destination connection GUID */
	getInfo("SRC_I_CONNECT")	getInfo("SRC _CONNECT _ GUID") /* Return the Source connection GUID */
	getInfo("DEST_I_ PSCHEMA")	getInfo("DEST _PSCHEMA _ GUID") /* Return the Destination Physical Schema GUID */

Table A–1 (Cont.) The New Substitution API Methods

Impacted ODIRef Methods	Previous ODIRef Method	New ODIRef Method
	getInfo("SRC_I_PSCHEMA")	getInfo("SRC _PSCHEMA _ GUID") /* Return the Source Physical Schema GUID */
	getInfo("DEST_I_ LSCHEMA")	getInfo("DEST _LSCHEMA _ GUID") /* Return the Destination Logical Schema GUID */
	getInfo("SRC_I_LSCHEMA")	getInfo("SRC _LSCHEMA _ GUID") /* Return the Source Logical Schema GUID */
	getInfo("DEST_I_TECHNO")	getInfo("DEST _TECHNO _ GUID") /* Return the Destination Technology GUID */
	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 */