

Oracle® Fusion Middleware

Upgrading Oracle GoldenGate Studio



12c (12.2.1.3.0)

E95571-02

July 2019

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

ORACLE®

Oracle Fusion Middleware Upgrading Oracle GoldenGate Studio, 12c (12.2.1.3.0)

E95571-02

Copyright © 2016, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction to Oracle GoldenGate Studio Upgrade	
	Upgrading the Oracle GoldenGate Studio to 12c	1-1
	About Upgrade Restrictions	1-2
2	Preparing to Upgrade Oracle GoldenGate Studio	
	Pre-Upgrade Checklist	2-2
	Creating a Complete Backup	2-3
	Backing Up the Schema Version Registry Table	2-4
	Cloning Your Production Environment for Testing	2-4
	Verifying Certification and System Requirements	2-5
	Verify Your Environment Meets Certification Requirements	2-6
	Verify System Requirements and Specifications	2-6
	Verify That the Database Hosting Oracle Fusion Middleware is Supported	2-7
	Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware	2-7
	Updating Policy Files when Using Enhanced Encryption (AES 256)	2-8
	Purging Unused Data	2-8
	Creating an Edition on the Server for Edition-Based Redefinition	2-8
	Creating a Non-SYSDBA User to Run the Upgrade Assistant	2-9
	Identifying Existing Schemas Available for Upgrade	2-11
3	Performing the Oracle GoldenGate Studio Upgrade	
	About the Oracle GoldenGate Studio Upgrade Process	3-1
	Stopping Servers and Processes	3-2
	Installing Oracle GoldenGate Studio	3-4
	Upgrading Product Schemas	3-5
	Starting the Upgrade Assistant	3-6
	Upgrade Assistant Parameters	3-6
	Upgrading the Schemas with the Upgrade Assistant	3-8
	Verifying the Schema Upgrade	3-11

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

About Updating the JDK Location After Installing an Oracle Fusion Middleware Product	A-1
Updating the JDK Location in an Existing Oracle Home	A-2
Updating the JDK Location in an Existing Domain Home	A-3

Preface

This document describes how to upgrade an existing Oracle GoldenGate Studio environment to 12c (12.2.1.3.0).

- [Audience](#)
Identify the target audience for your book and learn more about this document intended for.
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)
Learn about the conventions used in this document.

Audience

Identify the target audience for your book and learn more about this document intended for.

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

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

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

- For Oracle GoldenGate Studio information, see Oracle Data Integrator 12.2.1.3.0.
- For installation information, see Fusion Middleware Installation Documentation.
- For upgrade information, see Fusion Middleware 12c Upgrade Documentation.

Conventions

Learn about the conventions used in this document.

This document uses the following text conventions:

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

1

Introduction to Oracle GoldenGate Studio Upgrade

This document explains the procedure to upgrade the Oracle GoldenGate Studio from a previous 12c release to 12.2.1.3.0.

Review the topics in this chapter to understand and prepare for the upgrade.

Note:

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

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

This chapter contains the following topics:

- [Upgrading the Oracle GoldenGate Studio to 12c](#)
Oracle GoldenGate Studio is a standalone product and 12c 12.2.1.0 is the first release. You can upgrade to Oracle GoldenGate Studio 12c (12.2.1.3.0) from the previous Oracle GoldenGate Studio 12c release.
- [About Upgrade Restrictions](#)
If you are using two or more Oracle Fusion Middleware products of the same or different versions in a single, supported, Oracle Fusion Middleware configuration, you must consider the interoperability and compatibility factors before planning the upgrade.

Upgrading the Oracle GoldenGate Studio to 12c

Oracle GoldenGate Studio is a standalone product and 12c 12.2.1.0 is the first release. You can upgrade to Oracle GoldenGate Studio 12c (12.2.1.3.0) from the previous Oracle GoldenGate Studio 12c release.

Upgrading to Oracle GoldenGate Studio from 12.2.1.0 to 12.2.1.3.0 is an in-place upgrade performed by the Upgrade Assistant. You have to install the 12.2.1.3.0 product distribution in a new Oracle home, and then use the Upgrade Assistant to upgrade the 12.2.1.0 schemas and the security metadata. Oracle GoldenGate Studio 12.2.1.3.0 requires an upgraded repository; it does not work when you use the 12.2.1.0 repository.



Note:

The existing 12.2.1.0 production environment is not affected by the upgrade process.

About Upgrade Restrictions

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

Interoperability

In the context of Oracle Fusion Middleware products, interoperability is defined as the ability of two Oracle Fusion Middleware products or components of the same version (or release) to work together (interoperate) in a supported Oracle Fusion Middleware configuration. Specifically, interoperability applies when the first 4 digits of the release or version number are the same. For example, Oracle Fusion Middleware 12c (12.2.1.0) components are generally interoperable with other 12c (12.2.1.3.0) components.

Compatibility

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

For a list of products and features available in Oracle Fusion Middleware Release 12.2.1.3.0, see *Products and Features Available in Oracle Fusion Middleware in Oracle Fusion Middleware Understanding Interoperability and Compatibility*.

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

- Oracle Fusion Middleware Supported System Configurations
- Oracle® Fusion Middleware System Requirements and Specifications

2

Preparing to Upgrade Oracle GoldenGate Studio

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

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

- [Pre-Upgrade Checklist](#)
The Pre-Upgrade Checklist identifies tasks that can be performed before you begin your upgrade to ensure that you have a successful upgrade and limited downtime.
- [Creating a Complete Backup](#)
Before you start an upgrade, back up all system-critical files, including the databases that host your Oracle Fusion Middleware schemas.
- [Cloning Your Production Environment for Testing](#)
Create a copy of your actual production environment, upgrade the cloned environment, verify that the upgraded components work as expected, and then (and only then) upgrade your production environment.
- [Verifying Certification and System Requirements](#)
Review the certification matrix and system requirements documents to verify that your environment meets the necessary requirements for installation.
- [Updating Policy Files when Using Enhanced Encryption \(AES 256\)](#)
If you plan to use enhanced encryption, such as Advanced Encryption Standard (AES) 256, in your upgraded environment, Oracle recommends that you apply the latest required policy files to the JDK before you upgrade.
- [Purging Unused Data](#)
Purging unused data and maintaining a purging methodology before an upgrade can optimize the upgrade process.
- [Creating an Edition on the Server for Edition-Based Redefinition](#)
Before upgrading an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c.
- [Creating a Non-SYSDBA User to Run the Upgrade Assistant](#)
Oracle recommends that you create a non-SYSDBA user called `FMW` to run the Upgrade Assistant. This user has the privileges required to modify schemas, but does not have full administrator privileges.
- [Identifying Existing Schemas Available for Upgrade](#)
This optional task enables you to review the list of available schemas before you begin the upgrade by querying the schema version registry. The registry contains schema information such as version number, component name and ID, date of creation and modification, and custom prefix.

Pre-Upgrade Checklist

The Pre-Upgrade Checklist identifies tasks that can be performed before you begin your upgrade to ensure that you have a successful upgrade and limited downtime.

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

Note:

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

This table describes the Pre-Upgrade Checklist. It lists all the required components and describes them in detail.

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

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

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

Task	Description
<p>Required</p> <p>Verify that you install and upgrade your product on a supported hardware and software configuration.</p> <p>CAUTION: Do not attempt an upgrade if you are unable to use the latest supported operating system. As with all supported configurations, failure to comply with these requirements may cause your upgrade to fail.</p>	<p>Verify that your hardware and software configurations (including operating systems) are supported by the latest certifications and requirements documents. Also make sure to use a supported JDK version before you install the 12c product distributions.</p> <p>Oracle recommends that you verify this information right before you start the upgrade as the certification requirements are frequently updated.</p> <p>Make sure that you have applied the latest patches to your components before you upgrade.</p> <p>See Verifying Certification and System Requirements.</p>
<p>Required for 32-bit Operating Systems Only</p> <p>Migrate to a 64-bit operating system before you can upgrade.</p>	<p>This is required only if you are currently running an unsupported 32-bit operating system.</p>
<p>Optional</p> <p>Update the security policy files if you are using enhanced encryption (AES 256).</p>	<p>Some of the security algorithms used in Fusion Middleware 12c require additional policy files for the JDK.</p> <p>If you plan to use enhanced encryption, such as AES 256, Oracle recommends that you apply the latest required policy files to the JDK before you upgrade.</p> <p>See Updating Policy Files when Using Enhanced Encryption (AES 256).</p>
<p>Optional</p> <p>Purge any outdated or unused data before you upgrade.</p>	<p>To optimize performance, Oracle strongly recommends that you purge the data and objects that are not used in the upgraded environment.</p> <p>See Purging Unused Data.</p>
<p>Required for Oracle Database Users Only</p> <p>Before you upgrade an Edition-Based Redefinition (EBR) enabled schema, you must connect to the database server and create an edition on the database server for 12c (12.2.1.3.0).</p>	<p>If you are using an Edition-Based Redefinition (EBR) database, you must create the edition before you start the upgrade.</p> <p>See Creating an Edition on the Server for Edition-Based Redefinition.</p>
<p>Optional</p> <p>Create a Non-SYSDBA user to run the Upgrade Assistant.</p>	<p>Oracle recommends that you create the FMW user to run the Upgrade Assistant. The FMW user can run the Upgrade Assistant without any system administration privileges.</p> <p>See Creating a Non-SYSDBA User to Run the Upgrade Assistant.</p>
<p>Optional</p> <p>Identify the schemas that are currently in your domain before you begin.</p>	<p>It is important that you know the schemas that are in your pre-upgrade domain before you start the upgrade. You should know the schema owner names and passwords, as well as the versions of each schema.</p> <p>See Identifying Existing Schemas Available for Upgrade.</p>

Creating a Complete Backup

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

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

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

See:

- [Backing Up Your Environment in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*](#)
- [Upgrading and Preparing Your Oracle Databases for 12c in *Oracle Fusion Middleware Planning an Upgrade of Oracle Fusion Middleware*](#)
- [Backing Up the Schema Version Registry Table](#)
Your system backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table or the `FMWREGISTRY.SCHEMA_VERSION_REGISTRY$` table.

Backing Up the Schema Version Registry Table

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

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



Note:

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

Cloning Your Production Environment for Testing

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

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

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

 **Note:**

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

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

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

 **Note:**

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

Verifying Certification and System Requirements

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

 **Note:**

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

 **WARNING:**

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

- [Verify Your Environment Meets Certification Requirements](#)
Oracle has tested and verified the performance of your product on all certified systems and environments. Make sure that you are installing your product on a supported hardware or software configuration.
- [Verify System Requirements and Specifications](#)
It is important to verify that the system requirements such as disk space, available memory, specific platform packages and patches, and other operating system-specific items are met.
- [Verify That the Database Hosting Oracle Fusion Middleware is Supported](#)
You must have a supported Oracle database configured with the required schemas before you run Oracle Fusion Middleware 12c.
- [Verify That the JDK Is Certified for This Release of Oracle Fusion Middleware](#)
At the time this document was published, the certified JDK for 12c (12.2.1.3.0) was 1.8.0_131.

Verify Your Environment Meets Certification Requirements

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

Whenever new certifications occur, they are added to the appropriate certification document right away. New certifications can occur at any time, and for this reason the certification documents are kept outside of the documentation libraries and are available on Oracle Technology Network. See the Certification Matrix for 12c (12.2.1.3.0).

Verify System Requirements and Specifications

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

Use the *Oracle Fusion Middleware System Requirements and Specifications* document to verify that the requirements of the certification are met. For example, if the Certification Matrix for 12c (12.2.1.3.0) indicates that your product is certified for installation on 64-Bit Oracle Linux 7, the System Requirements and Specifications document should be used to verify that your Oracle Linux 7 system has met the required minimum specifications such as disk space, available memory, specific platform packages and patches, and other operating system-specific items. This document is updated as needed and resides outside of the documentation libraries on the Oracle Technology Network (OTN).

Note:

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

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

Verify That the Database Hosting Oracle Fusion Middleware is Supported

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

Review the Fusion Middleware database requirements before starting the upgrade to ensure that the database hosting Oracle Fusion Middleware is supported and has sufficient space to perform an upgrade. See the Certification Matrix for 12c (12.2.1.3.0).

Note:

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

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

At the time this document was published, the certified JDK for 12c (12.2.1.3.0) was 1.8.0_131.

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

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

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

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

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

Updating Policy Files when Using Enhanced Encryption (AES 256)

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

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

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

Note:

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

Purging Unused Data

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

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

Creating an Edition on the Server for Edition-Based Redefinition

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

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

 **Note:**

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

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

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

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

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

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

```
Edition created.
```

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

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

Creating a Non-SYSDBA User to Run the Upgrade Assistant

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

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

 **Notes:**

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

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

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

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

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

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

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

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

Identifying Existing Schemas Available for Upgrade

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

You can let the Upgrade Assistant upgrade all of the schemas in the domain, or you can select individual schemas to upgrade. To help decide, follow these steps to view a list of all the schemas that are available for an upgrade:

1. If you are using an Oracle database, connect to the database by using an account that has Oracle DBA privileges, and run the following from SQL*Plus:

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

2. Examine the report that is generated.

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

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

Notes:

- If your existing schemas are not from a supported version, then you must upgrade them to a supported version before using the 12c (12.2.1.3.0) upgrade procedures. Refer to your pre-upgrade version documentation for more information.
- Some components, such as Oracle Enterprise Data Quality, Oracle GoldenGate Monitor, and Oracle GoldenGate Veridata, support an upgrade from versions other than the standard Oracle Fusion Middleware supported versions.
- If you used an OID-based policy store in 11g, make sure to create a new OPSS schema before you perform the upgrade. After the upgrade, the OPSS schema remains an LDAP-based store.
- You can only upgrade schemas for products that are available for upgrade in Oracle Fusion Middleware release 12c (12.2.1.3.0). Do not attempt to upgrade a domain that includes components that are not yet available for upgrade to 12c (12.2.1.3.0).

3

Performing the Oracle GoldenGate Studio Upgrade

The Oracle GoldenGate Studio upgrade from a previous 12c release to 12c (12.2.1.3.0) is performed in-place; that is, the upgrade operations are performed on the existing 12c domain.

High-level upgrade steps:

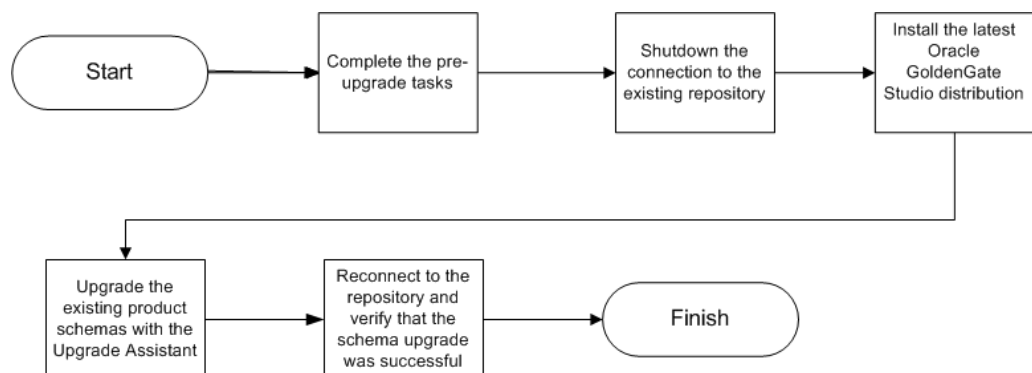
- [About the Oracle GoldenGate Studio Upgrade Process](#)
Review the flowchart and roadmap for an overview of the upgrade process for Oracle GoldenGate Studio .
- [Stopping Servers and Processes](#)
Before you run the Upgrade Assistant to upgrade your schemas and configurations, you must shut down all of the pre-upgrade processes and servers, including the Administration Server and any managed servers.
- [Installing Oracle GoldenGate Studio](#)
Before you begin your upgrade, download the Oracle GoldenGate Studio 12c (12.2.1.3.0) distribution on the target system and install it by using Oracle Universal Installer.
- [Upgrading Product Schemas](#)
After stopping servers and processes, use the Upgrade Assistant to upgrade supported product schemas to the current release of Oracle Fusion Middleware.

About the Oracle GoldenGate Studio Upgrade Process

Review the flowchart and roadmap for an overview of the upgrade process for Oracle GoldenGate Studio .

The steps you take to upgrade your existing domain varies depending on how your domain is configured and which components are being upgraded. Follow only those steps that are applicable to your deployment.

Figure 3-1 Upgrade Process Flowchart for Oracle GoldenGate Studio



The following table lists the high-level steps that you need to perform to upgrade to Oracle GoldenGate Studio 12c (12.2.1.3.0):

Table 3-1 Tasks for Upgrading Oracle GoldenGate Studio

Task	Description
<p>Required</p> <p>If you have not done so already, review the introductory topics in this guide and complete the required pre-upgrade tasks.</p>	<p>The pre-upgrade tasks include cloning your production environment, verifying system requirements and certifications, purging unused data, and creating a non-SYSD-BA user.</p> <p>For a complete list of pre-upgrade tasks, see Preparing to Upgrade Oracle GoldenGate Studio.</p>
<p>Required</p> <p>Shut down the connection to the existing 12c repository.</p>	<p>Before you start the upgrade process, shut down the connection to the 12c (12.2.1.3.0) repository.</p> <p>WARNING: Failure to shut down your servers during an upgrade may lead to data corruption.</p> <p>See Stopping Servers and Processes.</p>
<p>Required</p> <p>Download and install the Oracle GoldenGate Studio 12c (12.2.1.3.0) distribution in the new Oracle home.</p>	<p>To install the product distribution, see Installing Oracle GoldenGate Studio.</p>
<p>Required</p> <p>Upgrade the existing schemas with the Upgrade Assistant.</p>	<p>The schemas that you created during the 12c (12.2.1.x) installation are supported in 12c (12.2.1.3.0). Therefore, you do not need to create the schemas again.</p> <p>You must upgrade all the repository schemas by using the Upgrade Assistant. See Using the Oracle Fusion Middleware Upgrade Assistant to Upgrade Product Schemas.</p>
<p>Required</p> <p>Connect to the 12c (12.2.1.3.0) repository and verify your upgrade.</p>	<p>The upgrade process is complete. You can now verify the 12c (12.2.1.3.0) instance to compare the previous 12c environment with the 12c (12.2.1.3.0) and verify that the data and configuration settings are consistent in the newly upgraded environment.</p>

Stopping Servers and Processes

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

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

 **Note:**

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

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

Step 1: Stop System Components

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

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

You can stop system components in any order.

Step 2: Stop the Managed Servers

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

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

When prompted, enter your user name and password.

Step 3: Stop Oracle Identity Management Components

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

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

Step 4: Stop the Administration Server

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

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

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

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

Step 5: Stop Node Manager

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

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

Installing Oracle GoldenGate Studio

Before you begin your upgrade, download the Oracle GoldenGate Studio 12c (12.2.1.3.0) distribution on the target system and install it by using Oracle Universal Installer.

To install Oracle GoldenGate Studio:

1. Sign in to the target system.
2. Download the following from [Oracle Technology Network](#) or [Oracle Software Delivery Cloud](#) to your target system:
 - Oracle GoldenGate Studio (`fmw_12.2.1.3.0_oggstudio_generic.jar`)

 **Note:**

In case there are two jar files, download both the files.

3. Change to the directory where you downloaded the 12c (12.2.1.3.0) product distribution.
4. Start the installation program by entering the following command:
(UNIX) `JDK_HOME/bin/java -jar fmw_12.2.1.3.0_oggstudio_generic.jar`
(Windows) `JDK_HOME\bin\java -jar fmw_12.2.1.3.0_oggstudio_generic.jar`
5. On UNIX operating systems, the Installation Inventory Setup screen appears if this is the first time you are installing an Oracle product on this host.

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

 **Note:**

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

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

Support credentials then click **Search**. To configure a proxy server for the installer to access My Oracle Support, click **Proxy Settings**. Click **Test Connection** to test the connection.

Click **Next**.

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

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

9. On the Installation Type screen, select **Complete Install** and click **Next**.
10. The Prerequisite Checks screen analyzes the host computer to ensure that the specific operating system prerequisites have been met.

To view the list of tasks that are verified, select **View Successful Tasks**. To view log details, select **View Log**. If any prerequisite check fails, then an error message appears at the bottom of the screen. Fix the error and click **Rerun** to try again. To ignore the error or the warning message and continue with the installation, click **Skip** (not recommended).

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

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

Click **Install** to begin the installation.

12. On the Installation Progress screen, when the progress bar displays 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.
13. The Installation Complete screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.

Upgrading Product Schemas

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

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

- [Starting the Upgrade Assistant](#)
Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.3.0). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.
- [Upgrading the Schemas with the Upgrade Assistant](#)
Navigate through the screens in the Upgrade Assistant to upgrade the product schemas.

- [Verifying the Schema Upgrade](#)
After completing all the upgrade steps, verify that the upgrade was successful by checking that the schema version in `schema_version_registry` has been properly updated.

Starting the Upgrade Assistant

Run the Upgrade Assistant to upgrade product schemas, domain component configurations, or standalone system components to 12c (12.2.1.3.0). Oracle recommends that you run the Upgrade Assistant as a non-SYSDBA user, completing the upgrade for one domain at a time.

To start the Upgrade Assistant:

Note:

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

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

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

- [Upgrade Assistant Parameters](#)

Upgrade Assistant Parameters

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

Table 3-2 Upgrade Assistant Command-Line Parameters

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

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

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

Upgrading the Schemas with the Upgrade Assistant

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

Note:

Make sure that the administration and managed servers are stopped before you launch the Upgrade Assistant.

To upgrade product schemas with the Upgrade Assistant:

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

Note:

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

2. On the Selected Schemas screen, select **Individually Selected Schemas**. This option allows you to select only those schemas that you want to include in the upgrade. Click **Next**.

3. On the Available Components screen, select **Oracle GoldenGate Studio** and click **Next**.
4. On the Prerequisites screen, acknowledge that the prerequisites have been met by selecting all the check boxes. Click **Next**.

 **Note:**

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

5. On the OGGSTUDIO Schema screen, specify the connection credentials for the repository that needs to be upgraded.

Specify the service name and the database administrator account: DBA username and password for the database that contains the OGGSTUDIO schema and click **Connect**.

 **Note:**

Do not use SYSDBA as the database user name.

6. Specify the schema user name and the password for the selected schema.
When you see the message “Connection to database successfully completed”, click **Next**.
7. On the Studio Options screen, select **Upgrade topology and security metadata** and **Use AES-128 encryption algorithm**. Click **Next**.

 **Note:**

If you do not select **Use AES-128 encryption algorithm**, the AES-256 encryption algorithm is used for the upgrade.

8. On the OGG Studio Supervisor screen, specify the supervisor credentials for the OGG repository that you want to upgrade and click **Next**.
9. On the Examine screen, review the status of the Upgrade Assistant as it examines each schema, verifying that the schema is ready for upgrade. If the status is **Examine finished**, click **Next**.

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

 **Note:**

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

10. The Upgrade Summary screen lists the schemas that will be upgraded and/or created.

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

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

Click **Next**.

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

 **Caution:**

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

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

 **Note:**

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

Click **Next**.

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

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

 **Note:**

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

Verifying the Schema Upgrade

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

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

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

In the query result:

- Check that the number in the `VERSION` column matches the latest version number for that schema. For example, verify that the schema version number is 12.2.1.3.0.

 **Note:**

However, that not all schema versions will be updated. Some schemas do not require an upgrade to this release and will retain their pre-upgrade version number.

- The `STATUS` field will be either `UPGRADING` or `UPGRADED` during the schema patching operation, and will become `VALID` when the operation is completed.
- If the status appears as `INVALID`, the schema update failed. You should examine the logs files to determine the reason for the failure.
- Synonym objects owned by `IAU_APPEND` and `IAU_VIEWER` will appear as `INVALID`, but that does not indicate a failure.

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

A

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

Consider that you have a JDK version `jdk1.8.0_121` installed on your machine. When you install and configure an Oracle Fusion Middleware product, the utilities, such as Configuration Wizard (`config.sh|exe`), OPatch, or RCU point to a default JDK, for example, `jdk1.8.0_121`. After some time, Oracle releases a new version of the JDK, say `jdk1.8.0_131` that carries security enhancements and bug fixes.

From 12c (12.2.1.3.0) onwards, you can upgrade the existing JDK to a newer version, and can have the complete product stack point to the newer version of the JDK.

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

- [About Updating the JDK Location After Installing an Oracle Fusion Middleware Product](#)

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

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

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

The utility scripts such as `config.sh|cmd`, `launch.sh`, or `opatch` reside in the `ORACLE_HOME`, and when you invoke them, they refer to the `JAVA_HOME` variable located in `.globalEnv.properties` file. To point these scripts and utilities to the newer version of JDK, you must update the value of the `JAVA_HOME` variable in the `.global-Env.properties` file by following the directions listed in [Updating the JDK Location in an Existing Oracle Home](#).

To make the scripts and files in your Domain home directory point to the newer version of the JDK, you can follow one of the following approaches:

- Specify the path to the newer JDK on the Domain Mode and JDK screen while running the Configuration Wizard.

For example, consider that you installed Oracle Fusion Middleware Infrastructure with the JDK version `8u121`. So while configuring the WebLogic domain with the Configuration Assistant, you can select the path to the newer JDK on the Domain

Mode and JDK screen of the Configuration Wizard. Example: `/scratch/jdk/jdk1.8.0_131`.

- Manually locate the files that have references to the JDK using `grep` (UNIX) or `findstr` (Windows) commands and update each reference. See [Updating the JDK Location in an Existing Domain Home](#).

Note:

If you install the newer version of the JDK in the same location as the existing JDK by overwriting the files, then you don't need to take any action.

- [Updating the JDK Location in an Existing Oracle Home](#)
The `getProperty.sh|cmd` script displays the value of a variable, such as `JAVA_HOME`, from the `.globalEnv.properties` file. The `setProperty.sh|cmd` script is used to set the value of variables, such as `OLD_JAVA_HOME` or `JAVA_HOME` that contain the locations of old and new JDKs in the `.globalEnv.properties` file.
- [Updating the JDK Location in an Existing Domain Home](#)
You must search the references to the current JDK, for example `jdk1.8.0_121` manually, and replace those instances with the location of the new JDK.

Updating the JDK Location in an Existing Oracle Home

The `getProperty.sh|cmd` script displays the value of a variable, such as `JAVA_HOME`, from the `.globalEnv.properties` file. The `setProperty.sh|cmd` script is used to set the value of variables, such as `OLD_JAVA_HOME` or `JAVA_HOME` that contain the locations of old and new JDKs in the `.globalEnv.properties` file.

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

(UNIX) `ORACLE_HOME/oui/bin`

(Windows) `ORACLE_HOME\oui\bin`

Where, `ORACLE_HOME` is the directory that contains the products using the current version of the JDK, such as `jdk1.8.0_121`.

To update the JDK location in the `.globalEnv.properties` file:

1. Use the `getProperty.sh|cmd` script to display the path of the current JDK from the `JAVA_HOME` variable. For example:

(UNIX) `ORACLE_HOME/oui/bin/getProperty.sh JAVA_HOME`

(Windows) `ORACLE_HOME\oui\bin\getProperty.cmd JAVA_HOME`

`echo JAVA_HOME`

Where `JAVA_HOME` is the variable in the `.globalEnv.properties` file that contains the location of the JDK.

2. Back up the path of the current JDK to another variable such as `OLD_JAVA_HOME` in the `.globalEnv.properties` file by entering the following commands:

(UNIX) `ORACLE_HOME/oui/bin/setProperty.sh -name OLD_JAVA_HOME -value specify_the_path_of_current_JDK`

(Windows) `ORACLE_HOME\oui\bin\setProperty.cmd -name OLD_JAVA_HOME -value specify_the_path_of_current_JDK`

This command creates a new variable called `OLD_JAVA_HOME` in the `.global-Env.properties` file, with a value that you have specified.

3. Set the new location of the JDK in the `JAVA_HOME` variable of the `.global-Env.properties` file, by entering the following commands:

```
(UNIX) ORACLE_HOME/oui/bin/setProperty.sh -name JAVA_HOME -  
value specify_the_location_of_new_JDK  
(Windows) ORACLE_HOME\oui\bin\setProperty.cmd -name JAVA_HOME  
-value specify_the_location_of_new_JDK
```

After you run this command, the `JAVA_HOME` variable in the `.globalEnv.properties` file now contains the path to the new JDK, such as `jdk1.8.0_131`.

Updating the JDK Location in an Existing Domain Home

You must search the references to the current JDK, for example `jdk1.8.0_121` manually, and replace those instances with the location of the new JDK.

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

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

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

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

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