Oracle® Fusion Middleware

Upgrading Oracle GoldenGate

19c (19.1.0)

E98075-04

June 2021

Oracle Fusion Middleware Upgrading Oracle GoldenGate, 19c (19.1.0)

E98075-04

Contents

Preface

Audience iv
Documentation Accessibility iv
Related Information iv
Conventions iv

Part I Upgrading Oracle GoldenGate Microservices

Part II Upgrading Oracle GoldenGate Classic

Preface

This guide contains instructions for upgrading Oracle GoldenGate Classic and Microservices Architecture for Oracle and other non-Oracle databases from a previous release of Oracle GoldenGate to release 19c (19.1.0).

- Audience
- Documentation Accessibility
- Related Information
- Conventions

Audience

This guide is intended for system and database administrators who are responsible for operating Oracle GoldenGate and maintaining its performance.

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`.

Accessible Access to Oracle Support

Oracle customers who 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 Information

The Oracle GoldenGate Product Documentation Libraries are found at

https://docs.oracle.com/en/middleware/goldengate/index.html

For additional information on Oracle GoldenGate, refer to:

https://www.oracle.com/middleware/technologies/goldengate.html

https://www.oracle.com/database/technologies/high-availability/oracle-database-maa-best-practices.html

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, such as "From the File menu, select **Save**." Boldface also is used for terms defined in text or in the glossary.
italic *`italic`*	Italic type indicates placeholder variables for which you supply particular values, such as in the parameter statement: `TABLE` *`table_name`*. Italic type also is used for book titles and emphasis.
`monospace` `MONOSPACE`	Monospace type indicates code components such as user exits and scripts; the names of files and database objects; URL paths; and input and output text that appears on the screen. Uppercase monospace type is generally used to represent the names of Oracle GoldenGate parameters, commands, and user-configurable functions, as well as SQL commands and keywords.
UPPERCASE	Uppercase in the regular text font indicates the name of a utility unless the name is intended to be a specific case.
{ }	Braces within syntax enclose a set of options that are separated by pipe symbols, one of which must be selected, for example: {*`option1`* | *`option2`* | *`option3`*}.
[ ]	Brackets within syntax indicate an optional element. For example in this syntax, the `SAVE` clause is optional: `CLEANUP REPLICAT` *`group_name`* `[, SAVE` *`count`*`]`. Multiple options within an optional element are separated by a pipe symbol, for example: `[`*`option1`* `|` *`option2`*`]`.

Part I

Upgrading Oracle GoldenGate Microservices

These instructions are for upgrading Oracle GoldenGate Microservices Architecture from the previous releases of Oracle GoldenGate to the current release.

Topics:

- Overview of the Upgrade Procedure
  Learn about the prerequisites of performing an upgrade for Oracle GoldenGate.
- Obtaining the Oracle GoldenGate Distribution
- Scope of the Upgrade Procedure
  Before you start the upgrade, review the information about upgrading Extract and Replicat.
- Upgrading Oracle GoldenGate Microservices – GUI Based
  Learn the steps to upgrade Oracle GoldenGate Microservices using the GUI.
- Upgrading Oracle GoldenGate Microservices Using REST APIs
  Learn how to upgrade Oracle GoldenGate MA to Oracle GoldenGate MA 19c (19.1.0) using REST APIs.

Overview of the Upgrade Procedure

Learn about the prerequisites of performing an upgrade for Oracle GoldenGate.

For Microservices, the earliest version that can be upgraded from is Oracle GoldenGate 12*c* (12.3.0.1). As a best practice, perform a minimal upgrade first, so that you can troubleshoot more easily in the event that any problems arise. Once you know your environment is upgraded successfully, you can implement the new functionality.

The upgrade instructions also include the steps for upgrading the source or target database and Oracle GoldenGate at the same time. Following are the pre-upgrade requirements:

- Stop all Oracle GoldenGate processes.
- Upgrade the database and then start it.
- Start Oracle GoldenGate.
- Disable the DDL trigger if there is no native DDL support.

Obtaining the Oracle GoldenGate Distribution

To obtain Oracle GoldenGate, follow these steps:

1. Go to edelivery: edelivery.oracle.com

   Also see Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches (Doc ID 1557031.1) for more information.

   To access Oracle Technology Network, go to https://www.oracle.com/middleware/technologies/goldengate.html
2. Find the Oracle GoldenGate 19c (19.1.0) release and download the ZIP file onto your system.

For more information about locating and downloading Oracle Fusion Middleware products, see the Oracle Fusion Middleware Download, Installation, and Configuration Readme Files on Oracle Technology Network.

Scope of the Upgrade Procedure

Before you start the upgrade, review the information about upgrading Extract and Replicat.

Even though you may only be upgrading the source or target, rather than both, all processes are involved in the upgrade. All processes must be stopped in the correct order for the upgrade, regardless of which component you upgrade, and the trails must be processed until empty.

Oracle recommends that you begin your upgrade with the target rather than the source to avoid the necessity of adjusting the trail file format.

- Extract Upgrade Considerations
- Replicat Upgrade Considerations
- Extract Upgrade Considerations
- Replicat Upgrade Considerations

Extract Upgrade Considerations

If you are using trigger-based DDL support, you must rebuild the DDL objects, even if you plan to use the new triggerless DDL support in an integrated capture. After the upgrade, when Oracle GoldenGate is running successfully again, you can remove the trigger and DDL objects. See Upgrading a Configuration That Includes DDL Support for DDL upgrade considerations.

If you are upgrading multiple Extract processes that operate in a consolidated configuration (many sources to one target), follow the steps provided in Upgrading Oracle GoldenGate Classic Architecture for Oracle Database to upgrade one Extract at a time.

The output trail file is automatically rolled over when the Extract restarts and the integrated Extract version is upgraded.

Because the time zone is different, you may need to run the `ALTER REPLICAT` *`extseqno`* command to synchronize with newer trail files after consuming the old trail file written by Integrated Extract version 1.

After completing the upgrade, run the `UPGRADE HEARTBEATTABLE` command to add extra columns for tables and lag views. These extra columns are used to track the Extract restart position. See `UPGRADE HEARTBEATTABLE` to know more.

Replicat Upgrade Considerations

All Replicat installations should be upgraded at the same time. It is critical to ensure that all trails leading to all Replicat groups on all target systems are processed until empty, according to the upgrade instructions.

> **Caution:**
>
> The hash calculation used by the `@RANGE` function to partition data among Replicat processes has been changed. This change is transparent, and no re-partitioning of rows in the parameter files is required, so long as the upgrade is performed as directed in these instructions. To ensure data continuity, make certain to allow all Replicat processes on all systems to finish processing all of the data in their trails before stopping those processes, according to the upgrade instructions. Note that if the Replicat processes are not upgraded all at the same time, or the trails are not cleaned out prior to the upgrade, rows may shift partitions as a result of the new hash method, which may result in collision errors.

When upgrading from the 18*c* release of Oracle GoldenGate to the 19c (19.1.0) release, ensure that you do not use the `SOURCEDEF` parameter in Replicat, otherwise the Replicat will abend. However, if the trail file format is pre-12.2, then `SOURCEDEF` is still required because no metadata exists in the trail file.

For PostgreSQL: When upgrading from 12.2.0.1 to 19c (19.1.0), you need to run the `UPGRADE HEARTBEATTABLE` command from GGSCI to upgrade the heartbeattable to include the `LOGBSN` columns

Upgrading Oracle GoldenGate Microservices – GUI Based

Learn the steps to upgrade Oracle GoldenGate Microservices using the GUI.

Follow these steps to obtain the Oracle GoldenGate installation software and set up the directories for upgrade.

1. Download the latest Oracle GoldenGate MA 19c software from the Oracle Technology Network or eDelivery.
2. Upload the Oracle GoldenGate MA 19c software to a staging location on the server where a previous release of Oracle GoldenGate Microservices exists.
3. Unzip Oracle GoldenGate MA 19c software in the staging location.

   ```
   $ cd /tmp
   ```

   ```
   $ unzip ./fbo_ggs_Linux_x64_services_shiphome.zip
   ```

4. Untar the tar file that gets created after the unzip command:

   ```
   tar -xvf ggs_Linux_x64_Oracle_64bit.tar
   ```

5. Move into the unzipped files and execute the `runInstaller` command.

   ```
   $ cd ./fbo_ggs_Linux_x64_services_shiphome/Disk1
   ```

   ```
   $ ./runInstaller
   ```

6. For **Software Location**, specify where the new Oracle GoldenGate home will be located. This will not be the same location as the current Oracle GoldenGate home. Click **Next**.

7. Click **Install** to begin installing the new Oracle GoldenGate MA. When the installation is done, click **Close**.

8. At this point, you should have two Oracle GoldenGate MA home directories: one for your old home (12c or 18c) and a new home (19c).

Now, perform the following steps:

1. Verify the current version of Oracle GoldenGate Home through Service Manager.

   a. Login to the Service Manager: `http://`*`host`*`:`*`servicemanager_port`*

   b. Review the deployment section for your current Oracle GoldenGate home location.

2. Update the Service Manager and the deployments with the location of the new Oracle GoldenGate home.

   a. Click **Service Manager**, then the **Deployment name** link.

   b. Next to the deployment details, click the pencil icon. This opens the dialog box to edit the Oracle GoldenGate home.

   c. Update the Oracle GoldenGate home with the complete path to the new Oracle GoldenGate home. Also update the `LD_LIBRARY_PATH`, if required.

   d. Click **Apply**.

   e. Confirm that the Oracle GoldenGate home has been updated.

   f. Stop all Extracts, Replicats, and Distribution paths.

   g. Use the action button to restart **Service Manager** or **Deployment**.

   > **Note:**
   >
   > You can confirm that the Oracle GoldenGate home was updated by looking at the process from the operating system for Service Manager. The Service Manager process should be running from the new Oracle GoldenGate home.

3. To upgrade the associated deployments, follow the same steps for Service Manager after ensuring that all the Extract and Replicat processes in that deployment have been stopped.

Upgrading Oracle GoldenGate Microservices Using REST APIs

Learn how to upgrade Oracle GoldenGate MA to Oracle GoldenGate MA 19c (19.1.0) using REST APIs.

Follow these steps to perform the upgrade using REST APIs:

1. Download the latest Oracle GoldenGate MA 19c software from the Oracle Technology Network or eDelivery.
2. Upload the Oracle GoldenGate MA 19c software to a staging location on the server where a previous release of Oracle GoldenGate Microservices exists.
3. Unzip Oracle GoldenGate MA 19c software in the staging location.

   ```
   $ cd /tmp
   ```

   ```
   $ unzip ./fbo_ggs_Linux_x64_services_shiphome.zip
   ```

4. Untar the tar file that gets created after the unzip command:

   ```
   tar -xvf ggs_Linux_x64_Oracle_64bit.tar
   ```

5. Move into the unzipped files and execute the `runInstaller` command.

   ```
   $ cd ./fbo_ggs_Linux_x64_services_shiphome/Disk1
   ```

   ```
   $ ./runInstaller
   ```

6. For **Software Location**, specify where the new Oracle GoldenGate home will be located. This will not be the same location as the current Oracle GoldenGate home. Click **Next**.
7. Click **Install** to begin installing the new Oracle GoldenGate MA. When the installation is done, click **Close**.
8. At this point, you should have two Oracle GoldenGate MA home directories: one for your old home (12c or 18c) and a new home (19c).

Now, you are ready to update the Oracle GoldenGate MA home (`OGG_HOME`) for the Service Manager or deployments using REST API.

Upgrade a Service Manager

To upgrade the Service Manager, the following cURL command can be used to update the Oracle GoldenGate home:

```
curl -X PATCH \
  https://<hostname>:<port>/services/v2/deployments/ServiceManager \
  -H 'cache-control: no-cache' \
  -d '{"oggHome":"/opt/oracle/product/19.1.0/oggcore_1",
"status":"restart"}'
```

Upgrade a Deployment

To upgrade a Deployment:

1. Stop all Extract and Replicats within the Administration Service.

2. Stop all Distribution Paths within the Distribution Service.
3. Run this simple cURL command to update the Oracle GoldenGate home:

   ```
   curl -X PATCH \
     https:// ://<hostname>:<port>/services/v2/deployments/<deployment
   name> \
     -H 'cache-control: no-cache' \
     -d '{"oggHome":"/opt/app/oracle/product/19.1.0/oggcore_1",
      "status":"restart"}'
   ```

   > **Note:**
   >
   > You can confirm that the Oracle GoldenGate home was updated by looking at the process from the operating system for Service Manager. The Service Manager process should be running from the new Oracle GoldenGate home.

4. Start all distribution paths within the Distribution Server.
5. Start all Extracts and Replicats within the Administration Server.

Once the Service Manager or Deployment restarts, you are upgraded to the next version.

Part II

Upgrading Oracle GoldenGate Classic

Learn how to upgrade Oracle GoldenGate Classic from previous releases of Oracle GoldenGate.

Topics:

- Scope of the Upgrade Procedure
  Before you start the upgrade, review the information about upgrading Extract and Replicat.
- Upgrading Oracle GoldenGate for Oracle Database
- Upgrading Oracle GoldenGate for Heterogeneous Databases

Scope of the Upgrade Procedure

Before you start the upgrade, review the information about upgrading Extract and Replicat.

Even though you may only be upgrading the source or target, rather than both, all processes are involved in the upgrade. All processes must be stopped in the correct order for the upgrade, regardless of which component you upgrade, and the trails must be processed until empty.

Oracle recommends that you begin your upgrade with the target rather than the source to avoid the necessity of adjusting the trail file format.

- Extract Upgrade Considerations
- Replicat Upgrade Considerations
- Extract Upgrade Considerations
- Replicat Upgrade Considerations

Extract Upgrade Considerations

If you are using trigger-based DDL support, you must rebuild the DDL objects, even if you plan to use the new triggerless DDL support in an integrated capture. After the upgrade, when Oracle GoldenGate is running successfully again, you can remove the trigger and DDL objects. See Upgrading a Configuration That Includes DDL Support for DDL upgrade considerations.

If you are upgrading multiple Extract processes that operate in a consolidated configuration (many sources to one target), follow the steps provided in Upgrading Oracle GoldenGate Classic Architecture for Oracle Database to upgrade one Extract at a time.

The output trail file is automatically rolled over when the Extract restarts and the integrated Extract version is upgraded.

Because the time zone is different, you may need to run the `ALTER REPLICAT` *`extseqno`* command to synchronize with newer trail files after consuming the old trail file written by Integrated Extract version 1.

After completing the upgrade, run the `UPGRADE HEARTBEATTABLE` command to add extra columns for tables and lag views. These extra columns are used to track the Extract restart position. See `UPGRADE HEARTBEATTABLE` to know more.

Replicat Upgrade Considerations

All Replicat installations should be upgraded at the same time. It is critical to ensure that all trails leading to all Replicat groups on all target systems are processed until empty, according to the upgrade instructions.

> **Caution:**
>
> The hash calculation used by the `@RANGE` function to partition data among Replicat processes has been changed. This change is transparent, and no re-partitioning of rows in the parameter files is required, so long as the upgrade is performed as directed in these instructions. To ensure data continuity, make certain to allow all Replicat processes on all systems to finish processing all of the data in their trails before stopping those processes, according to the upgrade instructions. Note that if the Replicat processes are not upgraded all at the same time, or the trails are not cleaned out prior to the upgrade, rows may shift partitions as a result of the new hash method, which may result in collision errors.

When upgrading from the 18*c* release of Oracle GoldenGate to the 19c (19.1.0) release, ensure that you do not use the `SOURCEDEF` parameter in Replicat, otherwise the Replicat will abend. However, if the trail file format is pre-12.2, then `SOURCEDEF` is still required because no metadata exists in the trail file.

For PostgreSQL: When upgrading from 12.2.0.1 to 19c (19.1.0), you need to run the `UPGRADE HEARTBEATTABLE` command from GGSCI to upgrade the heartbeattable to include the `LOGBSN` columns

Upgrading Oracle GoldenGate for Oracle Database

These instructions are for upgrading to Oracle GoldenGate for an Oracle database.
Topics:

- Overview of the Upgrade Procedure
  Learn about the prerequisites of performing an upgrade for Oracle GoldenGate Classic.
- Prerequisites
- Obtaining the Oracle GoldenGate Distribution

- Upgrading Oracle GoldenGate Classic Architecture for Oracle Database
- Upgrading Oracle GoldenGate from OUI
- Upgrading Oracle GoldenGate using OUI – Silent
- Upgrading a Configuration That Includes DDL Support
- Upgrading Configuration that includes Berkeley Database - Oracle GoldenGate 12.2 or later

Overview of the Upgrade Procedure

Learn about the prerequisites of performing an upgrade for Oracle GoldenGate Classic.

As a best practice, perform a minimal upgrade first, so that you can troubleshoot more easily in the event that any problems arise. Once you know your environment is upgraded successfully, you can implement the new functionality.

The upgrade instructions also take into consideration the steps to take if you are upgrading the source or target database at the same time that you are upgrading Oracle GoldenGate. Take into account the following pre-upgrade requirements:

- Only if the trail file format is being changed, allow the Oracle GoldenGate processes to finish processing all current DML and DDL data in the Oracle GoldenGate trails.
- (Oracle only) Disable the DDL trigger if there is no native DDL support. Integrated Extract in Oracle 11.2.0.4 and up does not require the DDL trigger.
- When upgrading your database and Oracle GoldenGate simultaneously, you must upgrade the database first.

When upgrading your database and Oracle GoldenGate simultaneously, you must upgrade the database first.

Prerequisites

Before performing the upgrade procedure, read the *Release Notes for Oracle GoldenGate* to determine whether the new release affects the following in your configuration:

- New default process behavior.
- Parameters that changed or were deprecated.
- Parameters that were added to support a desired new feature or database type.
- Parameter default values that have changed.
- New data type support that might require changes to `TABLE` or `MAP` statements.
- Interaction with native database components that might require database change.

For Oracle Database., if you are using symbolic links that point to the dir directories, you need to use the parameter `ALLOWOUTPUTDIR` within GLOBALS.

You also need to see the sections Patching for Classic Architecture, Patching Oracle Platform with OPatch, Preparing to Patch 19c, and Installing Patches for Release in the *Installing Oracle GoldenGate* guide.

You can prevent startup delays that can cause lag by having all of your parameter changes made ahead of time, so that they are ready when you restart the processes. You should not make parameter changes while a process is running, but you can:

1. Make a copy of the parameter file.
2. Make edits to the copy.
3. After you shut down the processes during the upgrade procedure, copy the old parameter file to a new name (to save it as backup).
4. Copy the new parameter file to the old parameter file's name.

- Upgrade Considerations if Using Character-Set Conversion
- Upgrade Considerations if Using Quoted Object Names

Upgrade Considerations if Using Character-Set Conversion

Both the TRAILCHARSET and SOURCEDEF parameters are deprecated because Extract writes the source database character set with the column level. By default, these parameters are ignored but use the database character set and column character set from the table metadata. SOURCECHARSET parameter is only required when you need to override the source database character set and must use it with the OVERRIDE option.

Upgrade Considerations if Using Quoted Object Names

Oracle GoldenGate treats strings that are within single quotes as literals. Oracle GoldenGate has supported double-quoted object names since release 11.2 but did not fully implement the rule of single quotes for literals until release 12.1. Supporting double quotes for object names and single quotes for literals brings Oracle GoldenGate into compliance with SQL-92 rules and is now enabled by default. The USEANSISQLQUOTES parameter, which forced the SQL-92 standard in previous releases, is now deprecated.

The change to default SQL-92 rules affects object names in parameter files, input to SQLEXEC clauses, OBEY files, conversion functions, user exits, and commands. You have the following options as a result of this change:

- Retain non-SQL-92 quote rules: Oracle GoldenGate retains backward compatibility to enable the retention of current parameter files that do not conform to SQL-92 rules. To retain non-SQL-92 rules, add the NOUSEANSISQLQUOTES parameter to the GLOBALS file before you perform the upgrade and retain that parameter going forward. NOUSEANSISQLQUOTES affects Extract, Replicat, DEFGEN, and GGSCI.
- Upgrade your parameter files to use SQL-92 rules: Oracle GoldenGate provides the convprm conversion tool which you can run to convert your parameter files to be in conformance with SQL-92 rules. Run the convprm tool before you start the upgrade process.

- Overview of the convprm Tool
- Running convprm

Overview of the convprm Tool

The following describes the convprm tool:

- It is a command line program which can be run either manually or scripted.
- It converts string literals from double-quoted character strings to single-quoted character strings, but leaves double-quoted object names intact. It can distinguish between an object name and a string literal even when both are represented as a sequence of characters delimited with double quotes.
- It escapes quotation marks. Quotation marks must be escaped when the character that is used to delimit the string appears in the literal string itself. For example, the sentence "This character "" is a double quote" contains an escaped quote mark. The sentence 'This character '' is a single quote' contains an escaped single quote mark. When converting from double quotes to single quotes, `convprm` removes one of the repeated double quotes from escaped double quotes and escapes the single quotes that are embedded in the character sequence.
- It issues a warning message if `NOUSEANSISQLQUOTES` is specified in the `GLOBALS` file. The message states that the converted parameter file is incompatible with `NOUSEANSISQLQUOTES`, but the parameter file was updated anyway.
- It recursively converts the files that are included through an `OBEY` or `INCLUDE` parameter.
- It creates a backup of the initial parameter file in the same directory as the original file. The backup has the name of the original file with the `.bck` suffix. The creation of a backup file can be disabled when you run the `convprm` tool.
- It converts the character set. The character set for the new parameter file is taken from the `CHARSET` parameter in the original parameter file. Absent that parameter, the character set is taken from the `CHARSET` parameter in the `GLOBALS` file. Absent a `GLOBALS` parameter, the new parameter file is written in the character set of the local operating system.

Table 1 provides examples of the conversion outcome.

Table 1 Comparison of Input Requirements for [NO]USEANSISQLQUOTES

Input variable	String literal with old syntax	String literal with new syntax
Double quotes are escaped in the old syntax but not in the new syntax.	"abc""def"	'abc"def'
Single quotes are escaped in the new syntax but not in the old syntax.	"abc'def"	'abc''def'

Running convprm

To use the `convprm` tool:

1. Run `convprm` with the following syntax:

   ```
   convprm [options] input_files
   ```

   where:

   - `{-h | --help}` displays usage.
   - `{-v |--version}` displays version information.
   - `{-i | --follow-include}` recursively converts files included through an `OBEY` or `INCLUDE` parameter.

- {-n |--no-backup} does not create a copy of the original file.
- {-s | --silent} does not display status messages.
- {-q | --quotes} performs quote conversion. This is the default.
- {-d | --dry-run} does not change the parameter file or create a backup file. It only prints out what would happen as the result of the conversion.
- *input_files* is a list of the parameter files that are to be converted. Separate each file name with a white space, for example:

  ```
  convprm [options] extfin extacct exthr
  ```

2. Examine the parameter file to make certain the conversion completed successfully. Status messages are displayed at the beginning, during, or at the end of the file conversion process.

On errors, the process abends in the same way as other Oracle GoldenGate executables. All error messages that cause the converter to fail are sent to the Oracle GoldenGate error log.

If you are currently using the USEANSISQLQUOTES parameter, you may remove it or leave it in the parameter files. It is now the default.

Obtaining the Oracle GoldenGate Distribution

To obtain Oracle GoldenGate, follow these steps:

1. Go to edelivery: edelivery.oracle.com

   Also see Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches (Doc ID 1557031.1) for more information.

   To access Oracle Technology Network, go to https://www.oracle.com/middleware/technologies/goldengate.html
2. Find the Oracle GoldenGate 19c (19.1.0) release and download the ZIP file onto your system.

For more information about locating and downloading Oracle Fusion Middleware products, see the Oracle Fusion Middleware Download, Installation, and Configuration Readme Files on Oracle Technology Network.

Upgrading Oracle GoldenGate Classic Architecture for Oracle Database

These instructions contain the procedure for performing the minimal upgrade.

Before proceeding, make certain that you have reviewed the following preparation information:

- Prerequisites
- Overview of the Upgrade Procedure
- Scope of the Upgrade Procedure
- Upgrading a Configuration That Includes DDL Support

> **Note:**
>
> If you are using integrated capture and plan to upgrade from trigger-based DDL capture to new native DDL capture, **do not** remove the DDL trigger until prompted. Extract needs to mine DDL to the point where the redo COMPATIBLE level. For example, if Extract is behind by a week when the database is upgraded, Extract does not immediately switch to native DDL capture. It must be allowed to process the previous redo first, then Extract upgrades to native DDL capture automatically.

1. Use the following command in GGSCI to determine the oldest archive log that you might need to restore when Extract starts. The Recovery Checkpoint field shows the oldest log needed for recovery.

   ```
   GGSCI> INFO EXTRACT group, SHOWCH
   ```

   It's best to perform upgrade activities outside of the peak hours. If there are large and long running transactions, consider that on the source system, the new Extract may need to start processing from normal recovery checkpoint, rather than the bounded recovery checkpoint, if the first record of the oldest open transaction at the time that you stop Extract is in a log that is not on the system.

   The amount of time it takes to restart an Extract is directly related to the amount of uncomitted data at the time the Extract was stopped.

   You have two options:

   - You can restore the archives back to, and including, the one shown in the recovery checkpoint shown with INFO EXTRACT.
   - You can clear out the long-running transactions that apply to the Extract that you are upgrading. This can be done by skipping the transactions or by forcing them to the trail as committed transactions. Skipping a transaction may cause data loss, and forcing a transaction to the trail may add unwanted data to the trail if the transaction is rolled back. To skip or force a transaction, follow these steps:

     a. View open transactions with the following command in GGSCI. Record the transaction ID of any transaction that you want to clean up.

     ```
     GGSCI> SEND EXTRACT group, SHOWTRANS
     ```

     b. Clean up old transactions with the SEND EXTRACT command, using either the SKIPTRANS option to skip a transaction or the FORCETRANS option to force a transaction in its current state to the trail as a committed transaction.

     ```
     GGSCI> SEND EXTRACT group, {SKIPTRANS | FORCETRANS}
     transaction_ID [THREAD n] [FORCE]
     ```

     c. After you are finished cleaning up the long-running transactions, force a Bounded Recovery checkpoint.

     ```
     GGSCI> SEND EXTRACT group, BR BRCHECKPOINT IMMEDIATE
     ```

> **Note:**
>
> A forced checkpoint is necessary because the skipped or forced transaction is not cleaned up from the Bounded Recovery checkpoint and will be shown if `SHOWTRANS` is issued again. This is a known issue. For more information about `SEND EXTRACT`, see *Reference for Oracle GoldenGate*.

2. (Classic Extract in Oracle only) If you are using classic Extract, then after the DML and DDL have been stopped, run the following query. This query provides the minimum SCN needed to ensure all transactions are accounted for. Query the oldest SCN of any open transaction. If there are no open transactions, the current SCN of the database can be used.

```
SELECT MIN(SCN) AS INSTANTIATION_SCN
 FROM (SELECT MIN(START_SCN) AS SCN
         FROM gv$transaction
         UNION ALL
         SELECT CURRENT_SCN
         FROM gv$database);
```

> **Note:**
>
> You can avoid stopping DML operations when performing the Oracle GoldenGate upgrade, but you should stop DML when there is a requirement to run any SQL, such as DDL or any other upgrade SQL scripts.

The result of the query is:

```
YES = reached the logend
```

3. (Classic Extract in Oracle only) Record the SCN from the query result.

4. (Classic Extract for Source system in Oracle only) Issue the `SEND EXTRACT` command with the `LOGEND` option until it shows there is no more redo data to capture. You can skip this step, if you are using an integrated Extract.

   (Source system if using integrated Extract) Wait for the Extract recovery checkpoint to progress past the current SCN from the query result. To determine whether Extract is past that SCN in its checkpoints, view the Extract report file.

```
GGSCI> SEND EXTRACT group LOGEND
```

5. (Source system) In GGSCI, stop Extract and data pumps.

```
GGSCI> STOP EXTRACT group
```

6. If you are using integrated Extract, then you can skip this step. If you are using classic Extract, then follow these steps on the source system, only if currently using trigger-based DDL capture support:

   a. Run SQL*Plus and log in as a user that has `sysdba` privileges.

   b. Disconnect all sessions that ever issued DDL to avoid an error (ORA-04021). Ensure that no DDL sessions can be started for the duration of this upgrade.

c. From the Oracle GoldenGate directory, run the `ddl_disable` script to disable the Oracle GoldenGate DDL trigger.

d. Run the `ddl_remove` script to remove the Oracle GoldenGate DDL trigger and other associated objects and provide the name of the Oracle GoldenGate DDL schema.

   A `ddl_remove_spool.txt` log file is generated that logs the script output and a `ddl_remove_set.txt` file that logs current user environment settings for use in debugging.

e. Run the `marker_remove` script to remove the Oracle GoldenGate marker support system and provide the name of the Oracle GoldenGate DDL schema.

   A `marker_remove_spool.txt` file is generated that logs the script output and a `marker_remove_set.txt` file for use in debugging.

7. (Target systems) In GGSCI, stop all Replicat processes.

   ```
   GGSCI> STOP REPLICAT group
   ```

8. (Source and target systems) In GGSCI, stop Manager on the source and target systems.

   ```
   GGSCI> STOP MANAGER
   ```

9. When updating target systems only, or if updating the target side before the source side, you *must* use `STOP` to stop all data pumps and any primary Extracts that write directly to those targets on any source running on this target. Any static collectors that may have been started that must be stopped, as well. To verify that there are no `server` processes running, use process checking shell commands, such as `ps` and `grep`.

10. You need to use an out-of-place upgrade, which implies that you retain the existing installation in parallel while you run the upgrade. See Installing the Oracle GoldenGate Classic Architecture with Oracle Database and Patching for Classic Architecturein *Installing Oracle GoldenGate* for details.

11. In GGSCI, start the Oracle GoldenGate processes on the source and target systems in the following order.

    ```
    GGSCI> START MANAGER
    GGSCI> START EXTRACT group
    GGSCI> START EXTRACT pump
    GGSCI> START REPLICAT group
    ```

    If you need to restore any log files, Extract abends with an error that indicates the log to restore. Restore the logs back to, and including that log, and then restart Extract.

If you made copies of the parameter files to make parameter changes, move the new parameter files into the Oracle GoldenGate directory where the old parameter files were stored then rename them to the same names as the old parameter files. If you are using case-sensitivity support, ensure that you either add `NOUSEANSISQLQUOTES` to your parameter files, or that you ran the `convprm` utility to convert the quotes as required. See "Upgrade Considerations if Using Character-Set Conversion" for more information.

Upgrade Considerations for older Oracle GoldenGate Releases

- (Target systems, if upgrading Replicat from version 11.2.1.0.0 or earlier) In GGSCI on each target system, issue the following commands to upgrade the Replicat checkpoint tables on those systems. This step updates the table definition to add columns that support the Oracle GoldenGate 18c (18.1.0) release.

  ```
  GGSCI> DBLOGIN [{SOURCEDB} data_source] |
  [, database@host:port] | {USERID {/ | user id}
  [, PASSWORD password]   [algorithm ENCRYPTKEY {keyname | DEFAULT}]
  |USERIDALIAS alias
  [DOMAIN domain]}
  GGSCI> UPGRADE CHECKPOINTTABLE [owner.table]
  ```

  `owner.table` can be omitted if the checkpoint table was created with the name listed with `CHECKPOINTTABLE` in the GLOBALS file.

- (Source system) On the source system, if `ADD TRADATA` is used to add supplemental logging to a table then run the `ulg.sql script` as `sysdba`. This script converts the existing table-level supplemental log groups to the new format required by the new release. The script should run without error however, if errors occur, contact Oracle Support. This is not required if using `ADD SCHEMATRANDATA`.

Upgrading Oracle GoldenGate from OUI

You can use Oracle Universal Installer (OUI) on any of the Linux, UNIX, and Windows platforms that OUI supports and which Oracle GoldenGate supports. OUI is supported for Oracle versions 11g and later. An instance of Oracle GoldenGate can be installed for only one Oracle version in any given Oracle home. You can install multiple instances of Oracle GoldenGate for the same or different database versions on the same host.

The installer registers the Oracle GoldenGate home directory with the central inventory that is associated with the selected database. The inventory stores information about all Oracle software products installed on a host, provided the product was installed using OUI.

To perform the upgrade using OUI, perform the following steps:

1. Unzip and untar the installation file.
2. From the unzipped directory, run the **runInstaller** program on UNIX or Linux, or `run setup.exe` on Windows.
3. On the **Select Installation Option** page, select the Oracle GoldenGate build to install, and then click **Next** to continue.
4. On the **Specify Installation Details** page, specify the following:
   - For **Software Location**, specify the Oracle GoldenGate installation directory. It can be a new or existing directory. The default location is under installing user's home directory, but Oracle recommends changing it to a local directory that is not mounted and has no quotas. The specified directory cannot be a registered home in the Oracle central inventory. If installing in a cluster, install Oracle GoldenGate on shared storage that is accessible by all of the cluster nodes.

- (Optional) Select **Start Manager** to perform configuration functions, such as creating the Oracle GoldenGate sub-directories in the installation folder, setting library paths, and starting Manager on the specified port number. To proceed, a database must exist on the system. When Start Manager is selected, the **Database Location and Manager Port** fields are displayed.
  - The database must have a registered home in the Oracle central inventory. The installer registers the Oracle GoldenGate home directory with the central inventory.
  - For Manager Port, accept the default port number or enter a different unreserved, unrestricted port number for the Manager process to use for inter-process communication. The default port is the first available one starting with 7809. If you are installing multiple instances of Oracle GoldenGate on the same system, each must use a different port number.

5. Click **Next** to continue. In case of upgrading existing Oracle GoldenGate Installation, OUI prompts that the selected Software location has files or directories. Click on **Yes**.
6. The **Create Inventory** page is displayed if this is the first Oracle product to be installed from OUI on a host that does not have a central inventory.
   - For **Inventory Directory**, specify a directory for the central inventory. It can be a new directory or an existing directory that is empty and has the amount of disk space shown on the screen. The directory cannot be on a shared drive.
   - Select an operating system group in which the members have write permission to the inventory directory. This group is used to add inventory information to the Oracle GoldenGate sub-folder.
7. On the **Summary** page, confirm that there is enough space for the installation and that the installation selections are correct. Optionally, click **Save Response File** to save the installation information to a response file. You can run the installer from the command line with this file as input to duplicate the results of a successful installation on other systems. You can edit this file or create a new one from a template.
8. Click Install to begin the installation or **Back** to go back and change any input specifications. When Upgrading existing Oracle GoldenGate Installation, OUI will notify that the software location has files or directories. Click **Yes** to continue. You are notified when the installation is finished.
9. If you created a central inventory directory, you are prompted to run the `INVENTORY_LOCATION/orainstRoot.sh` script. This script must be executed as the root operating system user. This script establishes the inventory data and creates sub-directories for each installed Oracle product (in this case, Oracle GoldenGate).

Upgrading Oracle GoldenGate using OUI – Silent

These instructions apply to new installations, as well as upgrades.

You can perform a silent installation from the command console if the system has no X-Windows interface or to perform an automated installation. Silent installations can ensure that multiple users in your organization use the same installation options when they install your Oracle products.

You perform a silent installation by running a response file. You can create a response file by selecting the **Save Response File** option during an interactive OUI session or by editing a template. To run a response file, issue the following command.

```
runInstaller -silent -nowait -responseFile
absolute_path_to_response_file
```

The response files and the template are stored in the response subdirectory of the Oracle GoldenGate installation directory. The Oracle GoldenGate response file contains a standard set of Oracle configuration parameters in addition to parameters that are specific to Oracle GoldenGate. These parameters correspond to the fields in the interactive session.

> **Note:**
>
> If you are upgrading an existing Oracle GoldenGate installation with the silent option, you might get the following warning:
>
> ```
> WARNING:OUI-10030:You have specified a non-empty directory to install
> this product. It is recommended to specify either an empty or a non-
> existent directory.
> You may, however, choose to ignore this message if the directory
> contains Operating System generated files or subdirectories like
> lost+found. Do you want to proceed with installation in this Oracle
> Home?
> ```
>
> Press **ENTER** to continue.

Upgrading a Configuration That Includes DDL Support

This section contains considerations and steps you should take when DDL support is active in the Oracle GoldenGate environment.

DDL support in Oracle GoldenGate offers two options:

- The Integrated mode supports two DDL capture methods:
  - If the source database is Oracle 11.2.0.4 or later, DDL capture support is integrated into the database logmining server and does not require the use of the Oracle GoldenGate DDL trigger and supporting objects, as long as the database `COMPATIBLE` parameter is set to 11.2.0.4 or higher.
  - If the source database is earlier than Oracle 11.2.0.4, the Oracle GoldenGate trigger and supporting DDL objects must be used when Extract is in integrated mode.
- Classic capture requires the use of the Oracle GoldenGate DDL trigger and supporting objects regardless of the release number of the source Oracle database. (Source system) If you plan to use trigger-based DDL support for Oracle Database, use the following sub-steps to rebuild the Oracle GoldenGate DDL trigger environment to a clean state:
  1. Run SQL*Plus and log in as a user that has sysdba privileges.

2. Run the marker_setup script to reinstall the Oracle GoldenGate marker support system and provide the name of the Oracle GoldenGate schema.
3. Run the `ddl_setup` script and provide the name of the Oracle GoldenGate DDL schema.
4. Run the `role_setup` script to recreate the Oracle GoldenGate DDL role.
5. Grant the role that you created to all Oracle GoldenGate users under which the following Oracle GoldenGate processes run: Extract, Replicat, GGSCI, and Manager. You may need to make multiple grants if the processes have different user names.
6. Run the `ddl_enable.sql` script to enable the Oracle GoldenGate DDL trigger.

Table 2 shows possible DDL upgrade paths and guidelines.

Table 2 Possible Upgrade Paths to Oracle GoldenGate and Requirements for DDL Support

Upgrade from:	To: Classic capture using trigger method	To: Integrated capture, no trigger[1]
Classic capture using trigger method (all 11.2.1 database versions)	Cannot be used for a container database. Upgrade Oracle GoldenGate per these upgrade instructions.	Can be used for a container database. 1. Source database must be 11.2.0.4 or higher. 2. Source database `COMPATIBLE` setting must be 11.2.0.4 or higher. 3. Upgrade Oracle GoldenGate per these upgrade instructions.
Integrated capture using trigger method (all 11.2.1 database versions)	Cannot be used for a container database. No DDL upgrade path.	Can be used for a container database. 1. Source database must be 11.2.0.4 or higher. 2. Source database `COMPATIBLE` setting must be 11.2.0.4 or higher. 3. Upgrade Oracle GoldenGate per these upgrade instructions.

[1] An upgrade of the database to 11.2.0.4 or 12.1 automatically takes a data dictionary snapshot in the redo stream as part of the patch set upgrade.

Upgrading Configuration that includes Berkeley Database - Oracle GoldenGate 12.2 or later

When you are upgrading Oracle GoldenGate from release 12.1.2.1 to 12.3.0.1 and have enabled monitoring and the datastore is created by the Performance Metrics server, the best practice is to purge the data store before performing the upgrade. After the upgrade, the datastore is recreated. For more information about purging a datastore, see How to Purge the Datastore. in the *Using the Oracle GoldenGate Microservices Architecture*.

From Oracle GoldenGate 12.3.0.1 onward, all operations related to the datastore have been removed and are taken care of by the Performance Metrics server. To know more, see Monitoring Performance.

Upgrading Oracle GoldenGate for Heterogeneous Databases

These instructions are for upgrading Oracle GoldenGate Classic Architecture in the following heterogeneous database environments:

- DB2 for i
- DB2 LUW
- DB2 z/OS
- MySQL
- PostgreSQL
- SQL Server
- Teradata
- TimesTen

Topics:

- Overview of the Upgrade Procedure for Heterogeneous Databases
- Obtaining the Oracle GoldenGate Distribution
- Upgrading Oracle GoldenGate for Heterogeneous Databases

Overview of the Upgrade Procedure for Heterogeneous Databases

The upgrade performs a minimal feature upgrade to deploy only the core Oracle GoldenGate functionality, without implementing any of the major new features. It ensures easy troubleshooting of any upgrade related issues that may occur after the upgrade. After upgrading the Oracle GoldenGate environment successfully, you can implement the new functionality.

If you are upgrading multiple Extract processes that operate in a consolidated configuration (many sources to one target), you must upgrade one Extract at a time. All Replicat installations must be upgraded at the same time. It is critical to ensure that all trails leading to all Replicat groups on all target systems are processed until empty.

Note:

The hash calculation used by the `@RANGE` function to partition data among Replicat processes has been changed. This change is transparent, and no re-partitioning of rows in the parameter files is required. To ensure data continuity, ensure that you allow all Replicat processes on all systems to finish processing all the data in their trails before stopping those processes. If the Replicat processes are not upgraded all at the same time, or the trails are not cleaned out prior to the upgrade, rows may shift partitions as a result of the new hash method, which may result in collision errors.

Obtaining the Oracle GoldenGate Distribution

To obtain Oracle GoldenGate, follow these steps:

1. Go to edelivery: edelivery.oracle.com

   To access Oracle Technology Network, go to https://www.oracle.com/middleware/technologies/goldengate.html

2. Find the Oracle GoldenGate 19c (19.1.0) release and download the ZIP file onto your system.

For more information about locating and downloading Oracle Fusion Middleware products, see the Oracle Fusion Middleware Download, Installation, and Configuration Readme Files on Oracle Technology Network.

Upgrading Oracle GoldenGate for Heterogeneous Databases

Even though you may only be upgrading Extract or Replicat, rather than both, all processes are involved in the upgrade. All processes must be stopped in the correct order for the upgrade, regardless of which component you upgrade, and the trails must be processed until empty.

1. (Source and target systems) Back up the current Oracle GoldenGate installation directory on the source and target systems, and any working directories that you have installed on a shared drive in a cluster (if applicable).

2. (Source and target systems, as applicable) Expand the Oracle GoldenGate upgrade build into a new directory on each system (not the current Oracle GoldenGate directory). Do not create the sub-directories; just complete the steps to the point where the installation files are expanded.

3. Stop all user activity that generates DML and DDL on objects in your Oracle GoldenGate configuration and ensure that there are no outstanding open transactions against the database.

   For SQL Server CDC Extract on a Source system, manually stop the CDC Capture job for the database.

4. (Source system) In GGSCI on the source system, issue the `SEND EXTRACT` command with the `LOGEND` option until it shows `YES`, indicating that there is no more data in the transaction log to process.

For SQL Server CDC Extract on Source system, monitor that the current read position of the Extract is no longer updating, by repeatedly running `SEND EXTRACT` group `STATUS` for a few seconds, and observe that the LSN value for the current read position is no longer updating.

```
GGSCI> SEND EXTRACT group LOGEND
```

5. (Source system) In GGSCI, stop Extract and data pumps.

   ```
   GGSCI> STOP EXTRACT group
   ```

6. (Target systems) In GGSCI on each target system, issue the `SEND REPLICAT` command with the `STATUS` option until it shows a status of "`At EOF`" to indicate that it finished processing all of the data in the trail. This must be done on all target systems until all Replicat processes return "`At EOF`."

   ```
   GGSCI> SEND REPLICAT group STATUS
   ```

7. (Target systems) In GGSCI, stop all Replicat processes.

   ```
   GGSCI> STOP REPLICAT group
   ```

8. (Source and target systems) In GGSCI, stop Manager on the source and target systems and close GGSCI.

   ```
   GGSCI> STOP MANAGER
   ```

9. If you want to upgrade the source or target database, or both, do so at this time according to the upgrade instructions provided for that database. Ensure that you start the databases after the upgrade, but do not permit transactions on the objects in the Oracle GoldenGate configuration.

   For MySQL, if you upgrade from Oracle GoldenGate 19c (19.1.0) and the database is MySQL 5.7, then no change is required. However, if you upgrade from Oracle GoldenGate 19c (19.1.0) and the database is MySQL 8.0, then you need to perform the following steps:

   a. Enable logging of full metadata because it's mandatory for MySQL 8.0 and higher, regardless of DDL or DML replication. Logging of full metadata can be enabled by setting the value of MySQL server variable `binlog_row_metadata` to `FULL` inside the MySQL configuration file (`my.cnf` for Linux and `my.ini` for Windows). You need to bounce the server after changing the configuration file for the settings to take effect.

   b. Run the DDL uninstall scripts to disable old DDL solutions if DDL replication was enabled previously.

      The script name:

      For Windows: `ddl_install.bat`

      For Linux: `ddl_install.sh`

   c. To uninstall, run the following script:

      ```
      bash$ ./ddl_install.sh uninstall mysql userid password port
      ```

10. (Source and target systems) Move the expanded Oracle GoldenGate files from the new directory to your existing Oracle GoldenGate directory on the source and target systems overwriting the existing files.

11. (DB2 for i) Run `ggos400install` without arguments. For an upgrade, no arguments are necessary. However, if you change the library, the old library is left on the system until you remove it.

12. (Source and target systems) Start GGSCI.

13. (Target systems, if upgrading Replicat from version 11.2.1.0.0 or earlier only) In GGSCI on each target system, issue the following commands to upgrade the Replicat checkpoint tables on those systems. This step updates the table definition.

    ```
    GGSCI> DBLOGIN {
            [SOURCEDB data_source] |
            [, database@host:port] |USERID {/ | userid}
            [, PASSWORD password]
            [algorithm ENCRYPTKEY {keyname | DEFAULT}] |USERIDALIAS alias
    [DOMAIN domain] |
            [SYSDBA | SQLID sqlid]
            [SESSIONCHARSET character_set]}

    GGSCI> UPGRADE CHECKPOINTTABLE [owner.table]
    ```

    > **Note:**
    >
    > *owner.table* can be omitted if the checkpoint table was created with the name listed with CHECKPOINTTABLE in the GLOBALS file.

14. (SQL Server Source system) Follow the requirements for installing Oracle GoldenGate for SQL Server by reviewing the Installing on SQL Server in the *Installing Oracle GoldenGate* guide.

15. (SQL Server Classic Extract on Source system) DELETE TRANDATA against any tables previously enabled with it, including the gg_heartbeat and gg_heartbeat_seed tables if using the Oracle GoldenGate heartbeat implementation, and then ADD TRANDATA again to the tables.

    ```
    GGSCI> DBLOGIN {[SOURCEDB data_source] | |USERID {/ | userid}[,
    PASSWORD password] |USERIDALIAS alias [DOMAIN domain]
    GGSCI> DELETE TRANDATA schema.tablename
    GGSCI> ADD TRANDATA schema.tablename
    ```

16. (SQL Server CDC Extract on Source system) Run ADD TRANDATA again on any tables previously enabled with it, including any filter table and the gg_heartbeat and gg_heartbeat_seed tables if using the Oracle GoldenGate heartbeat implementation.

    ```
    GGSCI> DBLOGIN {[SOURCEDB data_source] | |USERID {/ | userid}[,
    PASSWORD password] |USERIDALIAS alias [DOMAIN domain]
    GGSCI> ADD TRANDATA schema.tablename
    ```

17. (Source and target systems) From Oracle GoldenGate 19c onward, you don't need to perform explicit ETROLLOVER because Oracle GoldenGate allows Extract to write into the trail in the same format as any existing trail.

18. (Source system if not upgrading Extract) Add the SOURCECHARSET parameter to the Replicat parameter file. Specify the character set of the source database with this parameter.

19. (Source system if not upgrading Replicat) If you are not upgrading Replicat on the target systems at this time, add the following parameter to the Extract parameter file to specify the version of Oracle GoldenGate that is running on the target. This parameter causes Extract to write a version of the trail that is compatible with the older version of Replicat.

    ```
    {EXTTRAIL | RMTTRAIL} file_name FORMAT RELEASE major.minor
    ```

    By default, Extract writes trail file in the exact same format with existing trail file format that is written before upgrade. The `FORMAT RELEASE` parameter is only required if you want to write trail in a newer format than the existing one.

20. If you made copies of the parameter files to make parameter changes, move the new parameter files into the Oracle GoldenGate directory where the old parameter files were stored, and give them the same names as the old parameter files. If using case-sensitivity support, make certain that you either added `NOUSEANSISQLQUOTES` to your parameter files, or that you ran the `convprm` utility to convert the quotes as required. See "Upgrade Considerations if Using Character-Set Conversion" for more information.

21. In GGSCI, start the Oracle GoldenGate processes on the source and target systems in the following order.

    ```
    GGSCI> START MANAGER
    GGSCI> START EXTRACT group
    GGSCI> START EXTRACT pump
    GGSCI> START REPLICAT group
    ```

22. If you need to restore any log files, Extract will abend with an error that indicates the log to restore. Restore the logs back to, and including that log, and then restart Extract.