Oracle® GoldenGate

Upgrading Oracle GoldenGate for Windows and UNIX

12*c* (12.1.2)

E40011-05

May 2015

Describes the procedures to upgrade Oracle GoldenGate for Windows and UNIX.

Oracle GoldenGate Upgrading Oracle GoldenGate for Windows and UNIX, 12*c* (12.1.2)

E40011-05

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, 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 failsafe, 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.

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

Contents

Preface

This guide contains instructions for upgrading Oracle GoldenGate for Windows and UNIX.

Audience

This guide is intended for the person or persons who are responsible for operating Oracle GoldenGate and maintaining its performance. This audience typically includes, but is not limited to, systems administrators and database administrators.

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 Oracle GoldenGate documentation set includes the following components:

Windows, UNIX, and Linux Platforms

- *Installing and Configuring Oracle GoldenGate for DB2 for i*
- *Installing and Configuring Oracle GoldenGate for DB2 LUW*
- *Installing and Configuring Oracle GoldenGate for DB2 z/OS*
- *Installing and Configuring Oracle GoldenGate for Informix*
- *Installing and Configuring Oracle GoldenGate for MySQL*
- *Installing and Configuring Oracle GoldenGate for NonStop SQL/MX*
- *Installing and Configuring Oracle GoldenGate for SQL Server*
- *Installing and Configuring Oracle GoldenGate for Oracle TimesTen*
- *Installing and Configuring Oracle GoldenGate for Oracle Database*

- *Installing and Configuring Oracle GoldenGate for Sybase*
- *Installing and Configuring Oracle GoldenGate for Teradata*
- *Administering Oracle GoldenGate for Windows and UNIX*
- *Reference for Oracle GoldenGate for Windows and UNIX*
- *Logdump Reference for Oracle GoldenGate*
- *Upgrading Oracle GoldenGate for Windows and UNIX*
- *Error Messages Reference for Oracle GoldenGate for Windows and UNIX*

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`*`]`.

1

Overview of Upgrading Oracle GoldenGate

This manual contains instructions for upgrading Oracle GoldenGate from version 11*g* to version 12*c*. This chapter contains an overview of this process and important pre-upgrade considerations that should be addressed.

This chapter includes the following sections:

- Who Should Perform the Upgrade
- Before You Upgrade
- Upgrade Considerations if Using Character-Set Conversion
- Upgrade Considerations if Using Quoted Object Names

1.1 Who Should Perform the Upgrade

Upgrades to Oracle GoldenGate should be performed by an administrator who has thorough knowledge of the commands that are necessary to control processes and obtain status information.

1.2 Before You Upgrade

Before performing the upgrade procedure, read the *Release Notes for Oracle GoldenGate for Windows and UNIX* (`http://docs.oracle.com/goldengate/1212/gg-winux/GRLWU/index.html`) 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.

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.

> **Note:** The sample files provided with the Oracle GoldenGate installation are overwritten with each upgrade. If you modify any Oracle delivered sample files, Oracle recommends that you make a copy of these file then modify the copy to avoid issues with changed files being overwritten.

1.3 Upgrade Considerations if Using Character-Set Conversion

The Replicat parameter TRAILCHARSET is renamed to SOURCECHARSET in version 12.1.2, but Oracle GoldenGate continues to support TRAILCHARSET as an alias. When upgrading Oracle GoldenGate, you can keep TRAILCHARSET or change it to SOURCECHARSET. One of those parameters is required to support character-set conversion from a pre-11.2.1 Extract if you are not upgrading that Extract at this time. If you are upgrading both Extract and Replicat to version 12.1.2, the information about the source character set is written to the trail automatically, and neither TRAILCHARSET nor SOURCECHARSET is needed.

1.4 Upgrade Considerations if Using Quoted Object Names

As of version 12.1.2, Oracle GoldenGate treats strings that are within single quotes as literals. Oracle GoldenGate has supported double-quoted object names since version 11.2 but did not fully implement the rule of single quotes for literals until version 12. 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.

1.4.1 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 1–1 provides examples of the conversion outcome.

Table 1–1 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'

1.4.2 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.

2 Upgrading to Version 12.1.2 for Oracle Database

These instructions are for upgrading Oracle GoldenGate to version 12*c* for an Oracle Database.

This chapter includes the following sections:

2.1 Overview of the Upgrade Procedure for Oracle Database

The upgrade instructions perform a minimal upgrade that deploys only the updated core functionality, without implementing any of the major new 12.1.2 features at this time, such as integrated Replicat, updated globalization support, new SQL-92 defaults, and enhanced security features. It is best practice to perform a minimal upgrade first, so that you can trouble shoot 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. Taken into account are the following pre-upgrade requirements:

- Allow the Oracle GoldenGate processes to finish processing all current DML and DDL data in the Oracle GoldenGate trails.
- Disable the DDL trigger.
- Upgrade the databases before upgrading Oracle GoldenGate.

2.2 Upgrading a Configuration that Includes DDL Support

This section contains considerations and steps you should take when DDL support is active in the current Oracle GoldenGate environment. DDL support in Oracle GoldenGate v12 offers two options:

- Integrated mode in version 12c of Oracle GoldenGate 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 Oracle version of the source database.

Table 2–1 shows possible DDL upgrade paths and guidelines.

Table 2–1 Possible Upgrade Paths to 12c and Requirements for DDL Support

Upgrade from:	To: Classic capture 12c using trigger method	To: Integrated capture 12c, 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.

2.3 Understanding the Scope of the Upgrade Procedure

Before performing the upgrade, review the following information about upgrading Extract and Replicat. 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.

2.3.1 Extract Upgrade Considerations

Most of these instructions pertain to the Extract portion of the upgrade. 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 integrated capture. After the upgrade, when Oracle GoldenGate is running successfully again, you can follow the steps in this documentation to remove the trigger and DDL objects. See Section 2.2, "Upgrading a Configuration that Includes DDL Support" for DDL upgrade considerations.

If you are upgrading multiple Extract processes that operate in a consolidation configuration (many sources to one target), follow these steps fully to upgrade one Extract at a time.

If you are using integrated capture, you will need to roll over the local and remote trails at the appropriate point in the upgrade steps. The 11.2.1 integrated Extract (version 1) captures `TIMESTAMP WITH LOCAL TIME ZONE` data as UTC, but the 12.1.2 integrated Extract (version 2) captures it as the local time of the database time zone.

2.3.2 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.

2.4 Downloading Oracle GoldenGate

Download the appropriate Oracle GoldenGate build to each system that will be part of the Oracle GoldenGate configuration.

> **Note:** If you have not upgraded to Oracle GoldenGate release 12.1.2.1.0 or greater, you must do so using the following steps.
>
> Once you have successfully upgraded, you use the same steps to apply any OPatches.

1. Using a browser, navigate to `http://support.oracle.com`.
2. Log in with your Oracle ID and password.
3. Select the **Patches and Upgrades** tab.
4. On the Search tab, click **Product or Family**.
5. In the `Product` field, type **Oracle GoldenGate**.
6. From the Release drop-down list, select the release version that you want to download.
7. Make certain that **Platform** is displayed as the default in the next field, and then select the platform from the drop-down list.
8. Leave the last field blank.
9. Click **Search**.

10. In the Patch Advanced Search Results list, select the patch that best meets your search criteria, making certain that the Oracle GoldenGate patch that you select corresponds to the version of the database that will be used.

    When you select the build, a dialog box pops up under the build description, and then you are advanced to the download page.

11. Click the Patch file name link for each patch that you want to download. The File Download dialog box appears.

12. Select either **Open with** or **Save File**:

To...	Select...
Extract the patch immediately	**Open with**, then select the desired file extraction utility and extract the files to a location on your file system.
Save the patch for later extraction	Select **Save file**, then save to a directory on your file system.

Note: Before installing the software, see *Release Notes for Oracle GoldenGate for Windows and UNIX* for any new features, parameter changes, upgrade requirements, or bug fixes that affect your current configuration. Review the readme file for known issues.

2.5 Preparing to Patch

Once the patch is downloaded, you need to prepare your environment before you can install it (you need to meet these prerequisites prior to deinstalling, too). To do so, ensure that your system meets the following requirements:

1. Ensure that the Oracle GoldenGate version on which you are installing the patch or from which you are rolling back the patch is Oracle GoldenGate release 12.1.2 for Oracle.

2. Use the latest version of OPatch. If you do not have the latest version, follow the instructions outlined in the My Oracle Support note 224346.1, available at:

   `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=224346.1`

3. Set the `ORACLE_HOME` environment variable to the Oracle GoldenGate installation directory, which is referred to as `ORACLE_HOME`.

4. Set the `PATH` environment variable to include the location of the unzip executable, and the `ORACLE_HOME` and the `ORACLE_HOME/OPatch` directories present in the Oracle GoldenGate home.

5. Verify the Oracle Inventory, which OPatch accesses to install the patches. To verify the inventory, run the following command:

   ```
   $ opatch lsinventory
   ```

   If the command displays any errors, contact Oracle Support and resolve the issue.

2.6 Installing the Patch

To install the patch, use this procedure:

1. Maintain a location for storing the contents of the patch ZIP file. In the rest of the document, this location (absolute path) is referred to as `PATCH_TOP_DIR`.
2. Extract the contents of the patch ZIP file to the location you created in step 1.

   For example:

   ```
   $ PATCH_TOP_DIR p14309369_112101_Solaris86-64.zip
   ```

3. Stop Oracle GoldenGate by doing the following:
   1. Run GGSCI.
   2. Stop Manager by running the command:

      ```
      STOP MANAGER
      ```

   3. Stop all Oracle GoldenGate processes by running the command:

      ```
      STOP ER *
      ```

4. Navigate to the `PATCH_TOP_DIR/`*Patch_number* directory:

   ```
   $ cd PATCH_TOP_DIR/Patch_number
   ```

5. Install the patch by running the following command:

   ```
   $ opatch apply
   ```

   > **Note:** When OPatch starts, it validates the patch and ensures that there are no conflicts with the software already installed in the `ORACLE_HOME` of the Oracle GoldenGate release. OPatch categorizes conflicts into the following types:
   >
   > - **Conflicts with a patch already applied to the ORACLE_HOME.** In this case, stop the patch installation and contact Oracle Support Services.
   > - **Conflicts with a patch already applied to the ORACLE_HOME that is a subset of the patch you are trying to apply.** In this case, continue with the patch installation because the new patch contains all the fixes from the existing patch in the ORACLE_HOME. The subset patch will automatically be rolled back prior to the installation of the new patch.

6. If the patch includes new parameters, edit the respective parameter files.
7. Set `ORACLE_HOME` back to the Oracle database home directory.
8. Start the processes from the Oracle GoldenGate home by doing the following:
   1. Ensure GGSCI is running.
   2. Start Manager by running this command:

      ```
      START MANAGER
      ```

   3. Start all Oracle GoldenGate processes by running this command:

      ```
      START ER *
      ```

2.7 Performing the Upgrade to Version 12.1.2 for Oracle Database

These instructions contain the procedure for performing the minimal upgrade.

Before proceeding, make certain you have reviewed the following preparation information in this manual:

- Section 1, "Overview of Upgrading Oracle GoldenGate"
- Section 2.1, "Overview of the Upgrade Procedure for Oracle Database"
- Section 2.2, "Upgrading a Configuration that Includes DDL Support"
- Section 2.3, "Understanding the Scope of the Upgrade Procedure"

> **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 is advanced to 11.2.0.4 (or higher). For example, if Extract is behind by a week when the database is upgraded to 11.2.0.4, Extract does not immediately switch to native DDL capture. It must be allowed to process the pre-11.2.0.4 redo first, then Extract upgrades to native DDL capture automatically.

1. (Source system) The new Extract might need to start processing from the 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. 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
   ```

   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 for Windows and UNIX*.

2. (Source system) Stop user activity that generates both DDL and DML on objects in the Oracle GoldenGate configuration.

3. After DDL and DML is stopped, run the following query. Record the current SCN from the query result.

   ```
   select current_scn from v$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 as specified in Step 7, 17, and 19.

4. Record the current SCN from the query result.

5. (Source system, if currently using classic capture) In GGSCI on the source system, issue the SEND EXTRACT command with the LOGEND option until it shows there is no more redo data to capture.

   (Source system if using integrated capture) 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
   ```

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

   ```
   GGSCI> STOP EXTRACT group
   ```

7. (Source system if not upgrading Target) 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. Before making the changes to the extract parameter file, please take a backup of the Oracle GoldenGate parameter files on source

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

   Where *version* specifies an Oracle GoldenGate release, *version. major* is the major version number, and *.minor* is the minor version number. For example, 11.2 or 12.1. Ensure that you include the period in *major.minor*.

8. (Source system, if currently using trigger-based DDL capture support) Follow these steps:

   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 ORA-04021 error. 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.

9. (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
   ```

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

    ```
    GGSCI> STOP REPLICAT group
    ```

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

    ```
    GGSCI> STOP MANAGER
    ```

12. 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`.

13. (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). You do not need to backup up the `dirdat` folder.

14. If you want to upgrade the source or target database, or both, do so at this time according to the Oracle database upgrade instructions for the new version. Ensure that you start and mount the databases after the upgrade though *do not* permit DML or DDL transactions on the objects in the Oracle GoldenGate configuration.

15. (Source and target systems) If you are using an Oracle GoldenGate version prior to 11.2.1, grant write permission to the installation user on the existing Oracle GoldenGate directory.

    For example, on UNIX or Linux, run the following command:

    ```
     $ chmod -R u+w existing_Oracle_GoldenGate_directory
    ```

16. (Source and target systems) Install Oracle GoldenGate version 12.1.2.0 using Oracle Universal Installer (OUI) into your existing Oracle GoldenGate directory on the source and target systems. Ensure that you unselect the **Start Manager** option during the OUI installation; otherwise, this may overwrite your existing

parameter files. See *Installing and Configuring Oracle GoldenGate for Oracle Database* for instructions.

> **Note:** Oracle recommends that you install with the interactive OUI option for upgrade purposes, rather than use the silent installation.

17. (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 12*c* (12.1.2) 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]
```

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

18. (Source system) On the source system, run the `ulg.sql` script as `sysdba`. This script converts the existing supplemental log groups to the new format required by the new release. This step is required even if you plan to use the new triggerless capture method. The script should run without error; if errors occur, contact Oracle Support.

19. (Source and target systems) Updates to the trail version, the checkpoint storage format, and integrated capture require that you perform the following steps:

    a. In GGSCI, alter the primary Extract process and the associated data-pump Extract processes to write to a new trail sequence number. The command should return "`Rollover performed`."

    ```
    GGSCI> ALTER EXTRACT group ETROLLOVER
    ```

    b. In GGSCI, issue the `INFO EXTRACT` command with `DETAIL` for the primary Extract and the data pumps to verify the trail sequence number.

    ```
    GGSCI> INFO EXTRACT group, DETAIL
    ```

    c. In GGSCI, reposition the data pumps and the Replicat processes to start at the new trail sequence number.

    ```
    GGSCI> ALTER EXTRACT pump, EXTSEQNO seqno, EXTRBA RBA

    GGSCI> ALTER REPLICAT group, EXTSEQNO seqno, EXTRBA RBA
    ```

20. (Source system) If you plan to use new trigger-based DDL support for Oracle Database, use the following sub-steps to rebuild the Oracle GoldenGate DDL trigger environment to a clean state:

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

    b. Run the `marker_setup` script to reinstall the Oracle GoldenGate marker support system and provide the name of the Oracle GoldenGate schema.

    c. Run the `ddl_setup` script and provide the name of the Oracle GoldenGate DDL schema.

    d. Run the `role_setup` script to recreate the Oracle GoldenGate DDL role.

   e. 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.

   f. Run the `ddl_enable.sql` script to enable the Oracle GoldenGate DDL trigger.

21. You may now restore DDL and DML activity on the source database.

22. 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" on page 1-2 for more information.

23. 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 will abend with an error that indicates the log to restore. Restore the logs back to, and including that log, and then restart Extract.

3

Upgrading to Version 12.1.2 for Non-Oracle Databases

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

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

This chapter includes the following sections:

3.1 Overview of the Upgrade Procedure for Non-Oracle Databases

The upgrade instructions perform a minimal upgrade only to deploy the updated core Oracle GoldenGate functionality, without implementing any of the major new 12.1.2 features at this time, such as coordinated Replicat, updated globalization support, new SQL-92 defaults, and enhanced security features. The best practice is to perform a minimal upgrade first, so that you can trouble shoot more easily in the event that any problems arise. Once you know your environment is upgraded successfully, you can implement the new functionality.

If you are upgrading multiple Extract processes that operate in a consolidation configuration (many sources to one target), follow these steps fully to upgrade one Extract at a time.

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 these 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.

3.2 Downloading Oracle GoldenGate

Download the appropriate Oracle GoldenGate build to each system that will be part of the Oracle GoldenGate configuration.

1. Using a browser, navigate to `http://support.oracle.com`.
2. Log in with your Oracle ID and password.
3. Select the **Patches and Upgrades** tab.
4. On the Search tab, click **Product or Family**.
5. In the `Product` field, type **Oracle GoldenGate**.
6. From the Release drop-down list, select the release version that you want to download.
7. Make certain that **Platform** is displayed as the default in the next field, and then select the platform from the drop-down list.
8. Leave the last field blank.
9. Click **Search**.
10. In the Patch Advanced Search Results list, select the patch that best meets your search criteria, making certain that the Oracle GoldenGate patch that you select corresponds to the version of the database that will be used.

    When you select the build, a dialog box pops up under the build description, and then you are advanced to the download page.
11. Click the Patch file name link for each patch that you want to download. The File Download dialog box appears.
12. Select either **Open with** or **Save File**:

To...	Select...
Extract the patch immediately	**Open with**, then select the desired file extraction utility and extract the files to a location on your file system.
Save the patch for later extraction	Select **Save file**, then save to a directory on your file system.

Note: Before installing the software, see *Release Notes for Oracle GoldenGate for Windows and UNIX* for any new features, parameter changes, upgrade requirements, or bug fixes that affect your current configuration. Review the readme file for known issues.

3.3 Performing the Upgrade to Version 12.1.2 for Non-Oracle 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.

WARNING: (Sybase database) To proceed with these instructions and upgrade to Oracle GoldenGate version 12.1.2 for Sybase, the Sybase page size must be at least 4K.

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 version 12.1.2 of Oracle GoldenGate 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. (Source system) Stop user activity on objects in the Oracle GoldenGate configuration.
4. (Source system) In GGSCI on the source system, issue the `SEND EXTRACT` command with the `LOGEND` option until it shows there is no more redo data to capture.

   ```
   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.

   ```
   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 by the database vendor. Make certain to start the databases after the upgrade, but do not permit transactions on the objects in the Oracle GoldenGate configuration.

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.

11. (DB2 for IBM 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. For more information about `ggos400install`, see *Installing and Configuring Oracle GoldenGate for DB2 for i.*

12. (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.

13. (Source and target systems) Updates to the trail version, the checkpoint storage format, and integrated capture require that you perform the following steps:

    a. In GGSCI, alter the primary Extract process and the associated data-pump Extract processes to write to a new trail sequence number. The command should return "`Rollover performed`."

    ```
    GGSCI> ALTER EXTRACT group ETROLLOVER
    ```

    b. In GGSCI, issue the `INFO EXTRACT` command with `DETAIL` for the primary Extract and the data pumps to verify the trail sequence number.

    ```
    GGSCI> INFO EXTRACT group, DETAIL
    ```

    c. In GGSCI, reposition the data pumps and the Replicat processes to start at the new trail sequence number.

    ```
    GGSCI> ALTER EXTRACT pump, EXTSEQNO seqno, EXTRBA RBA

    GGSCI> ALTER REPLICAT group, EXTSEQNO seqno, EXTRBA RBA
    ```

14. (SQL/MX, only if upgrading from Oracle GoldenGate version 10.4.x) From the Oracle GoldenGate installation directory on the source system, run the convchk program for the primary SQL/MX Extract to convert the checkpoints to the new format.

    ```
    ./convchk group
    ```

    If the checkpoints are not converted, the primary SQL/MX Extract process abends at startup and ignores all data in the transaction log.

15. (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. For more information, see *Reference for Oracle GoldenGate for Windows and UNIX*.

16. (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
    ```

    where: *version* specifies an Oracle GoldenGate release version. *major* is the major version number, and *minor* is the minor version number, such as 11.2. Make certain to include the dot.

17. (Oracle GoldenGate upgrades on DB2 z/OS) You might need to re-set the "a" attribute for the Extract process and its DLLs to APF-authorize them to use the privileged API. For more information, see the installation instructions in *Installing and Configuring Oracle GoldenGate for DB2 z/OS*.

18. 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" on page 1-2 for more information.

19. 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
    ```

20. 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.

4

Deploying New Features of Version 12.1.2

This chapter contains instructions for deploying advanced new features of Oracle GoldenGate after the initial upgrade has been performed successfully. At this point, you can make configuration changes that support the new features that you want to use. See *Release Notes for Oracle GoldenGate for Windows and UNIX* for information about new features in version 12.1.2.

> **Note:** These instructions assume that you tested 12.1.2 and that all processes are running properly post-upgrade.

This chapter contains the following topics.

- Upgrading Extract from Classic Mode to Integrated Mode (Oracle)
- Upgrading from Oracle Nonintegrated Replicat to Integrated Replicat (Oracle)
- Upgrading to Coordinated Replicat (All Databases)
- Deploying Feature Upgrades that Require Only Parameter Changes

4.1 Upgrading Extract from Classic Mode to Integrated Mode (Oracle)

Extract in integrated mode receives logical change records (LCRs) from a database logmining server. These instructions assume that Extract is currently running in Oracle GoldenGate version 11g classic capture mode and that you want to upgrade to integrated capture mode, either with the current trigger-based capture method or with the new triggerless method of mining DDL through the logmining server.

> **Note:** If you are using the `LOGRETENTION` option, where Extract works with the Oracle Recovery Manager (RMAN) to manage the logs, this upgrade automatically disables that feature and log retention is managed as part of integrated capture.

1. Back up the current Oracle GoldenGate working directories.
2. While the Oracle GoldenGate processes continue to run in their current configuration, so that they keep up with current change activity, copy the Extract parameter file to a new name.
3. Perform the following configuration tasks on the server where Extract will be running:

- Grant the appropriate privileges to the Extract user. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.
- Configure server resources to support shared memory for the logmining server. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.
- Edit the new parameter file to add the required parameters to support integrated capture. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.
- (If this will be a downstream mining deployment) Prepare the mining database. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.

4. To support the upgrade to integrated capture, the transaction log that contains the start of the oldest open transaction must be available on the system where Extract will be running. To determine the oldest open transaction, issue the `SEND EXTRACT` command with the `SHOWTRANS` option. You can use the `FORCETRANS` or `SKIPTRANS` options of this command to manage specific open transactions, with the understanding that skipping a transaction may cause data loss and forcing a transaction to commit to the trail may add unwanted data if the transaction is rolled back by the user applications. Review these options in *Reference for Oracle GoldenGate for Windows and UNIX* before using them.

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

5. Follow the instructions for switching from classic capture to integrated capture in *Administering Oracle GoldenGate for Windows and UNIX*.

Note: (Oracle database 11.2.0.4) Make certain to set the `ENABLE_GOLDENGATE_REPLICATION` database initialization parameter to `TRUE` before restarting Oracle GoldenGate processes. Otherwise, the database may not support some integration features of Oracle GoldenGate (classic or integrated).

4.2 Upgrading from Oracle Nonintegrated Replicat to Integrated Replicat (Oracle)

When operating in integrated mode with an Oracle target, Replicat constructs and transmits logical change records (LCRs) to an inbound server in the target database, which takes advantage of parallel apply capabilities to apply the data to the target database.

Note: Do not configure the switch between Replicat modes to occur immediately after Extract recovers from a failure or is repositioned to a different location in the transaction log.

1. Back up the current Oracle GoldenGate working directories on the target system where the Replicat that you want to upgrade is located.
2. While the Oracle GoldenGate processes continue to run in their current configuration, so that they keep up with current change activity, copy the Replicat parameter file to a new name.
3. Perform the following configuration tasks:

- Grant the appropriate privileges to the Replicat user. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.
- (Optional) Edit the new parameter file to add the DBOPTIONS parameter with the INTEGRATEDPARAMS option to specify parallelism. The default is 4. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.

```
GGSCI> EDIT PARAMS group
```

4. Follow the instructions for switching from nonintegrated Replicat to integrated Replicat in *Administering Oracle GoldenGate for Windows and UNIX*.

> **Note:** (Oracle database 11.2.0.4) Make certain to set the ENABLE_GOLDENGATE_REPLICATION database initialization parameter to TRUE before restarting Oracle GoldenGate processes. Otherwise, the database may not support some integration features of Oracle GoldenGate (classic or integrated).

4.3 Upgrading to Coordinated Replicat (All Databases)

This procedure upgrades a regular Replicat configuration (non-coordinated) to a coordinated configuration. This procedure assumes you are replacing a configuration that partitions data across multiple Extract and Replicat processes with a configuration that uses one Extract and one coordinated Replicat. The coordinated Replicat replaces the need for parallel Replicat processes. A coordinated Replicat requires only one trail, so there is no need for parallel Extract processes or data pumps.

For more information about coordinated Replicat, see *Administering Oracle GoldenGate for Windows and UNIX*.

4.3.1 Procedure Overview

This procedure makes use of the EVENTACTIONS parameter with a STOP action, which enables all of the Replicat processes to stop at the same point in the trail. The EVENTACTIONS action is triggered by a transaction that contains an INSERT to a dummy table. The INSERT causes each process to finish processing everything up to, and including, the event transaction and then stop cleanly. An additional event action of IGNORE is specified for Replicat to prevent the parallel Replicat processes from attempting to insert the same record to the target. The result of this procedure is that all processes stop at the same point in the data stream: after completing the INSERT transaction to the dummy table.

After the processes stop, you move all of the TABLE statements to one primary Extract group. You move the same TABLE statements to the data pump that reads the trail of the Extract group that you retained. You move all of the MAP statements to a new coordinated Replicat group that reads the remote trail that is associated with the retained data pump. Once all of the MAP statements are in one parameter file, you edit them to add the thread specifications to support a coordinated Replicat. (This can be done ahead of time.) Then you drop the Replicat group and add it back in coordinated mode with the same name.

4.3.2 Performing the Switch to Coordinated Replicat

To perform the switch to coordinated Replicat:

1. Review the configuration requirements for using a coordinated Replicat in *Administering Oracle GoldenGate for Windows and UNIX*. Do not create the Replicat group until prompted by these instructions.
2. Back up the current parameter files of all of the Extract groups, data pumps, and Replicat groups. You will be editing them.
3. Create a working directory outside the Oracle GoldenGate directory. You will use this directory to create and stage new versions of the parameter files. If needed, you can create a working directory on the source and target systems.
4. In the working directory, create a parameter file for a coordinated Replicat. Copy the `MAP` parameters from the active parameter files of all of the Replicat groups to this parameter file, and then add the thread specifications and any other parameters that support your required coordinated Replicat configuration
5. If using parallel primary Extract groups, select one to keep, and then save a copy of its current parameter file to the working directory.
6. Copy all of the `TABLE` statements from the other Extract groups to the new parameter file of the primary Extract that you are keeping.
7. In the working directory, save a copy of the parameter file of the data pump that is linked to the primary Extract that you are keeping.
8. Copy all of the `TABLE` statements from the other data pumps to the copied parameter file of the kept data pump.
9. In the source database, create a simple dummy table on which a simple `INSERT` statement can be performed. For this procedure, the name `schema.event` is used.
10. Edit the active parameter files (not the copies) of all primary and data-pump Extract groups to add the following `EVENTACTIONS` parameter to each one.

    ```
    TABLE schema.event, EVENTACTIONS(STOP);
    ```

11. Edit the active parameter files (not the copies) of all of the Replicat groups to add the following `EVENTACTIONS` parameter to each one.

    ```
    MAP schema.event, TARGET schema.event, EVENTACTIONS(IGNORE, STOP);
    ```

12. Stop the Oracle GoldenGate processes gracefully in the following order:
    - Stop all Replicat processes.
    - Stop all data pumps.
    - Stop all Extract processes.
13. Restart the Oracle GoldenGate processes in the following order so that the `EVENTACTIONS` parameters take effect:
    - Start all Extract processes.
    - Start all data pumps.
    - Start all Replicat processes.
14. On the source system, issue a transaction on the `schema.event` table that contains one `INSERT` statement. Make certain to commit the transaction.
15. In GGSCI, issue the `STATUS` command for all of the primary Extract and data pump processes on the source system, and issue the same command for all of the Replicat processes on the target system, until the commands show that all of the processes are `STOPPED`.

```
STATUS EXTRACT *
STATUS REPLICAT *
```

16. Replace the active parameter files of the primary Extract and data pump that you kept with the new parameter files from the working directory.
17. Delete the unneeded Extract and data pump groups and their parameter files.
18. Log into the target database by using the `DBLOGIN` command.

```
GGSCI> DBLOGIN {USERID {/ | userid}[, PASSWORD password]
   [algorithm ENCRYPTKEY {keyname | DEFAULT}] |
USERIDALIAS alias [DOMAIN domain]}
```

19. Delete all of the Replicat groups and their active parameter files.
20. Copy or move the new coordinated Replicat parameter file from the working directory to the Oracle GoldenGate directory.
21. In GGSCI, issue the `INFO EXTRACT` command for the data pump and make note of its write checkpoint position in the output (remote) trail.

```
INFO EXTRACT pump, DETAIL
```

22. Add a new coordinated Replicat group with the following parameters.

```
ADD REPLICAT group, EXTTRAIL trail, EXTSEQNO sequence_number, EXTRBA rba,
COORDINATED MAXTHREADS number [, CHECKPOINTTABLE owner.table]
```

where:

- *group* is the name of the coordinated Replicat group. The name must match that of the new parameter file created for this group.
- `EXTTRAIL` *trail* is the name of the trail that the data pump writes to.
- `EXTSEQNO` *sequence_number* is the sequence number of the trail as shown in the write checkpoint returned by the `INFO EXTRACT` that you issued for the data pump.
- `EXTRBA` *rba* is the relative byte address in the trail as shown in the write checkpoint returned by `INFO EXTRACT`. Together, these position Replicat to resume processing at the correct point in the trial.
- `MAXTHREADS` number specifies the maximum number of threads allowed for this group. This value should be appropriate for the number of threads that are specified in the parameter file.
- `CHECKPOINTTABLE` *owner.table* provide the option to specify a checkpoint table specifically for this Replicat group, rather than use the checkpoint table specified with the `GLOBALS` parameter `CHECKPOINTTABLE`.

23. Start the primary Extract group.
24. Start the data pump group.
25. Start the coordinated Replicat group.

4.4 Deploying Feature Upgrades that Require Only Parameter Changes

This procedure guides you through changes that are necessary to update the Oracle GoldenGate parameter files to add parameters that support new features and remove parameters that are no longer required.

Note: Some new features replace existing parameters with new ones. Make certain to remove the parameters that are no longer needed.

Use this procedure to implement the following new features:

- Enhanced encryption support for user passwords and trail files: You need to make parameter changes and add the required security structures to implement this new functionality. See *Release Notes for Oracle GoldenGate for Windows and UNIX* and *Administering Oracle GoldenGate for Windows and UNIX*.
- Conflict Detection and Resolution (CDR) enhancements: See *Release Notes for Oracle GoldenGate for Windows and UNIX* for changes to CDR resolution behavior. See *Administering Oracle GoldenGate for Windows and UNIX* for more information about conflict detection and resolution.
- Capture from and apply to an Oracle container database: You need to specify three-part object names and register Extract with the appropriate containers. See *Installing and Configuring Oracle GoldenGate for Oracle Database*.
- Other parameter changes listed in *Release Notes for Oracle GoldenGate for Windows and UNIX*.

To implement the new features:

1. Back up the current Oracle GoldenGate working directories on the source and target systems.
2. While the Oracle GoldenGate processes continue to run in their current configuration, so that they keep up with current change activity, copy the parameter files to new names.
3. Edit the new parameter files to add the new parameters. Do not change group names or the trail name.
4. Stop user activity on the source tables.
5. In GGSCI on the source system, issue the `SEND EXTRACT` command with the `LOGEND` option until it confirms that Extract finished processing all of the data in the transaction log.

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

6. In GGSCI, issue `SEND EXTRACT` with the `ROLLOVER` option to cause Extract to roll over the trail to a new file the next time it starts.

```
GGSCI> SEND EXTRACT group ROLLOVER
```

7. In GGSCI, stop Extract.
8. Allow user activity on the source tables.
9. In GGSCI on the target system, issue the `SEND REPLICAT` command with the `STATUS` option until it shows a status of "`At EOF`" to indicate that Replicat finished processing all of the data in the trail.

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

10. In GGSCI, stop the data pump on the source system and stop Replicat on the target.

```
GGSCI> STOP REPLICAT group
```

11. Copy the edited new parameter file(s) back to the original name(s).

12. In GGSCI, on the target system, start Replicat.

    ```
    GGSCI> START REPLICAT group
    ```

13. In GGSCI, on the source system, start Extract.

    ```
    GGSCI> START EXTRACT group
    ```

14. In GGSCI, start the data pump on the source system.

    ```
    GGSCI> START REPLICAT group
    ```