Oracle® GoldenGate
Upgrade Instructions
11g Release 2 (11.2.1.0.2)

E36359-01

August 2012

Oracle® GoldenGate Upgrade Instructions 11g Release 2 (11.2.1.0.2) E36359-01

Who should perform the upgrade

Important: 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.

Determining if you need to make parameter or environment changes

Before doing the upgrade procedure, please read the *New Features* section of the Oracle GoldenGate version 11.2.1.0.2 Release Notes 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

● new data type support that might require changes to TABLE or MAP statements

● interaction with native database components that might require some database changes

Preventing lag during upgrades

If you will be stopping processes during the upgrade, 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.

Upgrading to version 11.2.1.0.2 - Oracle database instructions

These instructions contain steps for upgrading Oracle GoldenGate in an Oracle database environment.

Important: These instructions assume that you are upgrading only to deploy the updated core functionality, *without* implementing any of the major new 11.2.1 features at this time, such as integrated capture, globalization support, case-sensitivity support, and enhanced encryption.

● It is best practice to perform a minimal upgrade first, so that you can trouble shoot more easily in the event any problems arise.

● Once you know your environment is upgraded successfully, you can implement the new functionality.

Select from one of the following upgrade paths:

. . . .

- Upgrading from Oracle GoldenGate v10.4.x.x.x to 11.2.1.0.2 for Oracle database (retain classic capture mode)
- Upgrading from Oracle GoldenGate v11.1.x.x.x to 11.2.1.0.2 for Oracle database (retain classic capture mode)

Upgrading from Oracle GoldenGate v10.4.x.x.x to 11.2.1.0.2 for Oracle database (retain classic capture mode)

1. 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. (Enterprise Edition v10.2 and higher only) For Oracle GoldenGate on Enterprise Edition v10.2 and higher, Extract can optionally be configured to work with Oracle Recovery Manager (RMAN) to retain the logs that Extract needs for recovery. With this new feature, you should be aware of the following:

   ❍ An Oracle Streams Capture process will be created for each Extract process and will have the same name as that process. These Capture processes are non-functional and can exist concurrently with other local Oracle Streams installations.

   ❍ To use the log retention feature, assign the Extract user the following additional privileges.

Oracle 10.2	Run package to grant Streams admin privilege.
	`exec dbms_streams_auth.grant_admin_privilege('<user>')`
	Grant INSERT into logmnr_restart_ckpt$.
	`grant insert on system.logmnr_restart_ckpt$ to <user>;`
	Grant UPDATE on streams$_capture_process.
	`grant update on sys.streams$_capture_process to <user>;`
	Grant the 'become user' privilege.
	`grant become user to <user>;`

11.1 and 11.2.0.1	Run package to grant Streams admin privilege.
	`exec dbms_streams_auth.grant_admin_privilege('<user>')`
	Grant the 'become user' privilege.
	`grant become user to <user>;`

11.2.0.2 and later	Run package to grant Streams admin privilege. `exec dbms_goldengate_auth.grant_admin_privilege('<user>')`

3. Install v11.2.1.0.2 of Oracle GoldenGate into a new directory. If you are upgrading the source and target systems, install on both. Do not create the sub-directories; just complete the steps to the point where the installation files are expanded.

4. This version of Oracle GoldenGate contains improved sequence support over version 10.4.x.x.x. To enable this support, you must install some database procedures on the **source** and **target** systems. If using sequences, continue with these steps. Otherwise, skip all of the steps marked "(Sequence upgrade, continued)".

••

5. (Sequence upgrade, continued) On the source and target Oracle systems, connect to SQL*Plus as sysdba.

6. (Sequence upgrade, continued) If you already assigned a database user to support the Oracle GoldenGate DDL replication feature, you can skip this step. Otherwise, on both systems create a database user that can also be the DDL user.

```
SQL> CREATE USER DDLuser IDENTIFIED BY password;
SQL> GRANT CONNECT,RESOURCE,DBA TO DDLuser;
```

7. (Sequence upgrade, continued) In the GLOBALS file on both systems (create one if needed), add the following parameter to designate the DDL schema.

```
GGSCHEMA DDLuser
```

8. (Sequence upgrade, continued) On both systems, run the sequence.sql script from the root of the Oracle GoldenGate installation directory. This script creates some procedures for use by Oracle GoldenGate processes. (Do not execute them yourself.) You are prompted for the schema that you created.

9. (Sequence upgrade, continued) On the source system, grant EXECUTE privilege on the updateSequence procedure to a database user that can be used to issue the DBLOGIN command. Remember or record this user.

```
SQL> GRANT EXECUTE on DDLuser.updateSequence TO DBLOGINuser;
```

10. (Sequence upgrade, continued) On the target system, grant EXECUTE privilege on the replicateSequence procedure to the Replicat database user.

```
SQL> GRANT EXECUTE on DDLuser.replicateSequence TO Replicatuser;
```

11. Stop user activity that generates both DDL and DML on objects in the Oracle GoldenGate configuration.

12. 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
```

13. In GGSCI, stop Extract and data pumps.

```
GGSCI> STOP EXTRACT <group>
```

14. (*If not upgrading the target system*) 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 that version of Replicat.

```
{EXTTRAIL | RMTTRAIL} 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. Make certain to include the dot.

15. (*If using DDL support*) Disable the DDL trigger by running the ddl_disable script from the Oracle GoldenGate directory on the source system.

16. 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 Replicat finished processing all of the data in the trail.

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

17. In GGSCI, stop Replicat.

```
GGSCI> STOP REPLICAT <group>
```

18. In GGSCI, stop Manager on the source and target systems.

```
GGSCI> STOP MANAGER
```

19. Move the expanded Oracle GoldenGate files from the new directory to your existing Oracle GoldenGate directory on source and target.

20. (If checkpoint table is in use by Replicat) In GGSCI on the target system, issue the following commands to upgrade the checkpoint table. This step updates the table definition to add columns.

```
GGSCI> DBLOGIN USERID {/ | <user id>} PASSWORD <password>
   [<encryption options>]

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.

21. *(If using DDL support)* On the source system, follow the instructions in "Rebuilding an Oracle DDL environment to a clean state" on page 7, and then return to these steps.

22. On the source system, run the ulg.sql script as sysdba. This script converts the existing supplemental log groups to the new format that is required for version 11.2.1. The script should return "Upgrade completed successfully." If it does not return this message, contact Oracle Support.

23. To use the log-retention feature, issue the following GGSCI commands on the source system.

```
GGSCI> DBLOGIN USERID {/ | <user id>} PASSWORD <password>
   [<encryption options>]
GGSCI> REGISTER EXTRACT <group> LOGRETENTION
```

24. An update of the trail version and an internal change to the checkpoint storage format requires that you perform the following steps:

❍ In GGSCI, alter the primary Extract process (that reads the transaction logs) 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 name> ETROLLOVER
```

❍ 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 name>, DETAIL
```

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

```
GGSCI> ALTER EXTRACT <pump name>, EXTSEQNO <seqno>, EXTRBA <RBA>
GGSCI> ALTER REPLICAT <group name>, EXTSEQNO <seqno>, EXTRBA <RBA>
```

25. In GGSCI, start the Oracle GoldenGate processes.

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

26. (If replicating sequences) Issue the following GGSCI commands on the source system immediately after starting Extract. They generate redo that Replicat uses to synchronize the source and target sequences. You can use an asterisk wildcard for any or all characters in the name of a sequence (but not the owner).

```
GGSCI> DBLOGIN USERID {/ | <user id>} PASSWORD <password>
   [<encryption options>]
GGSCI> FLUSH SEQUENCE <owner.sequence>
```

27. Allow user activity to resume.

Upgrading from Oracle GoldenGate v11.1.x.x.x to 11.2.1.0.2 for Oracle database (retain classic capture mode)

1. 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. Install v11.2.1.0.2 of Oracle GoldenGate into a new directory. If you are upgrading the source and target systems, install on both. Do not create the sub-directories; just complete the steps to the point where the installation files are expanded.

3. (*Oracle source database*) This step applies to any Extract that is configured with the Bounded Recovery feature (the default as of v.11.1.1.0.0). 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 can do either of the following:

❍ Restore all of the archives back to, and including, the one shown in the recovery checkpoint

❍ Use the SEND EXTRACT command with the SHOWTRANS option to view long-running transactions that apply to the Extract that you are upgrading, and then clean up any old ones to avoid restoring old archives. Use the following command to view old transactions, and then record the transaction ID of any transaction that you want to clean up.

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

• •

To clean up old transactions, take these steps:

❍ Use SEND EXTRACT to skip the transaction (SKIPTRANS) or force it to the trail as-is (FORCETRANS).

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

❍ 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 (see enhancement request 13529316).

4. Stop user activity that generates both DDL and DML on objects in the Oracle GoldenGate configuration.

5. 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
```

6. In GGSCI, stop Extract and data pumps.

```
GGSCI> STOP EXTRACT <group>
```

7. (*If not upgrading the target system*) 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 that version of Replicat.

```
{EXTTRAIL | RMTTRAIL} 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. Make certain to include the dot.

8. (*If using DDL support*) Disable the DDL trigger by running the ddl_disable script from the Oracle GoldenGate directory on the source system.

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 it finished processing all of the data in the trail.

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

10. In GGSCI, stop Replicat.

```
GGSCI> STOP REPLICAT <group>
```

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

```
GGSCI> STOP MANAGER
```

12. Move the expanded Oracle GoldenGate files from the new directory to your existing Oracle GoldenGate directory on the source and target systems.

. . . .

13. (If checkpoint table is in use by Replicat) In GGSCI on the target system, issue the following commands to upgrade the checkpoint table. This step updates the table definition to add columns.

```
GGSCI> DBLOGIN USERID {/ | <user id>} PASSWORD <password>
    [<encryption options>]

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. *(If using DDL support)* On the source system, follow the instructions "Rebuilding an Oracle DDL environment to a clean state" on page 7, and then return to these steps.

15. On the source system, run the ulg.sql script as sysdba. This script converts the existing supplemental log groups to the new format that is required for version 11.2.1. The script should return "Upgrade completed successfully." If it does not return this message, contact Oracle Support.

16. An update of the trail version and an internal change to the checkpoint storage format requires that you perform the following steps:

   ❍ In GGSCI, alter the primary Extract process (that reads the transaction logs) 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 name> ETROLLOVER
   ```

   ❍ 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 name>, DETAIL
   ```

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

   ```
   GGSCI> ALTER EXTRACT <pump name>, EXTSEQNO <seqno>, EXTRBA <RBA>
   GGSCI> ALTER REPLICAT <group name>, EXTSEQNO <seqno>, EXTRBA <RBA>
   ```

17. In GGSCI, start the Oracle GoldenGate processes on the source and target systems.

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

18. 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.

Rebuilding an Oracle DDL environment to a clean state

These steps completely remove, and then reinstall, the Oracle GoldenGate DDL objects in an Oracle source database in preparation for the upgrade.

Perform these steps on the source system. The scripts are located in the Oracle GoldenGate directory.

1. After stopping the source Oracle GoldenGate processes on the source system, change directories to the Oracle GoldenGate installation directory.

. . . .

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

3. Disconnect all sessions that ever issued DDL. Otherwise the database might generate an ORA-04021 error.

4. Run the ddl_disable script to disable the DDL trigger.

5. Run the ddl_remove script to remove the Oracle GoldenGate DDL trigger, the DDL history and marker tables, and other associated objects. This script produces a ddl_remove_spool.txt file that logs the script output and a ddl_remove_set.txt file that logs current user environment settings in case they are needed for debugging.

6. Run the marker_remove script to remove the Oracle GoldenGate marker support system. This script produces a marker_remove_spool.txt file that logs the script output and a marker_remove_set.txt file in case they are needed for debugging.

7. Run the marker_setup script to reinstall the Oracle GoldenGate marker support system. You will be prompted for the name of the Oracle GoldenGate schema.

8. Run the ddl_setup script. You will be prompted for the name of the Oracle GoldenGate DDL schema.

9. Run the role_setup script to recreate the Oracle GoldenGate DDL role.

10. 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 might need to make multiple grants if the processes have different user names.

11. Run the ddl_enable.sql script to enable the DDL trigger.

12. Return to the upgrade procedure that you are following:

    ❍ step 21 on page 4 if upgrading from v10.4.x
    ❍ step 14 on page 7 if upgrading from v11.1.x

Upgrading to v11.2.1.0.2 - non-Oracle database instructions

These instructions contain steps for upgrading Oracle GoldenGate in a non-Oracle database environment.

Important: These instructions assume that you are upgrading only to deploy the updated core functionality, *without* implementing any of the major new 11.2.1 features at this time, such as globalization support, case-sensitivity support, conflict detection and resolution (CDR), and enhanced encryption.

● It is best practice to perform a minimal upgrade first, so that you can trouble shoot more easily in the event any problems arise.

● Once you know your environment is upgraded successfully, you can implement the new functionality.

Select from one of the following upgrade paths:

"Upgrading from v10.4.x to v11.2.1.0.2 (source, target, or both)" on page 9

"Upgrading from v11.1.1.x.x to v11.2.1.0.2 (source, target, or both)" on page 10

• •

Oracle GoldenGate for Windows and UNIX Upgrade Guide   8

Upgrading from v10.4.x to v11.2.1.0.2 (source, target, or both)

1. 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. Install v11.2.1.0.2 of Oracle GoldenGate into a new directory. If you are upgrading the source and target systems, install on both. Do not create the sub-directories; just complete the steps to the point where the installation files are expanded.

3. In GGSCI, stop all of the Oracle GoldenGate processes, starting with Extract and Replicat processes, and then the Manager process.

   ```
   GGSCI> STOP EXTRACT <group>
   GGSCI> STOP REPLICAT <group>
   GGSCI> STOP MANAGER
   ```

4. Move the files and programs that you expanded from your new directory into your existing Oracle GoldenGate directory on the source and target systems.

5. *Oracle GoldenGate upgrades on SQL/MX and Teradata*: An update of the trail version and an internal change to the checkpoint storage format requires that you perform the following steps:

   ○ In GGSCI, alter the primary Extract process (that reads the transaction logs) 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 name> ETROLLOVER
      ```

   ○ 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 name>, DETAIL
      ```

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

      ```
      GGSCI> ALTER EXTRACT <pump name>, EXTSEQNO <seqno>, EXTRBA <RBA>
      GGSCI> ALTER REPLICAT <group name>, EXTSEQNO <seqno>, EXTRBA <RBA>
      ```

   ○ (SQL M/X only) 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 <Extract group name>
      ```

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

6. (*If not upgrading the target system*) 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 that version of Replicat.

   ```
   {EXTTRAIL | RMTTRAIL} 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. Make certain to include the dot.

7. *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 the Oracle GoldenGate Installation and Setup Guide for DB2 z/OS.

8. In GGSCI, start all of the Oracle GoldenGate processes, starting with Manager, and then Extract and Replicat.

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

Upgrading from v11.1.1.x.x to v11.2.1.0.2 (source, target, or both)

1. 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. Install v11.2.1.0.2 of Oracle GoldenGate into a new directory. If you are upgrading the source and target systems, install on both. Do not create the sub-directories; just complete the steps to the point where the installation files are expanded.

3. In GGSCI, stop all of the Oracle GoldenGate processes, starting with Extract and Replicat processes, and then the Manager process.

```
GGSCI> STOP EXTRACT <group>
GGSCI> STOP REPLICAT <group>
GGSCI> STOP MANAGER
```

4. Move the files and programs that you expanded from your new directory into your existing Oracle GoldenGate directory on the source and target systems.

5. *Oracle GoldenGate upgrades on SQL/MX and Teradata*: An update of the trail version requires that you perform the following steps:

   ❍ In GGSCI, alter the primary Extract process (that reads the transaction logs) 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 name> ETROLLOVER
   ```

   ❍ 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 name>, DETAIL
   ```

   ❍ In GGSCI, reposition the data pumps and Replicat processes to start at the new trail sequence number.
   ```
   GGSCI> ALTER EXTRACT <pump name>, EXTSEQNO <seqno>, EXTRBA <RBA>
   GGSCI> ALTER REPLICAT <group name>, EXTSEQNO <seqno>, EXTRBA <RBA>
   ```

• •

6. (*If not upgrading the target system*) 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 that version of Replicat.

```
{EXTTRAIL | RMTTRAIL} 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. Make certain to include the dot.

7. *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 the Oracle GoldenGate Installation and Setup Guide for DB2 z/OS.

8. In GGSCI, start all of the Oracle GoldenGate processes, starting with Manager, and then Extract and Replicat.

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

Deploying advanced new features of v11.2.1.0.2

Once you know that the basic upgrade to v11.2.1.0.2 is successful, you can make configuration changes that support the new features that you want to use. These instructions assume that you tested 11.2.1.0.2 and that all processes are running properly post-upgrade.

Steps follow to deploy these new features:

Deploying features that only require parameter changes

Use this procedure to implement the following new features:

● Enhanced encryption support for passwords, TCP/IP, and trail files. First, see "Configuring Oracle GoldenGate security" in the *Oracle GoldenGate for Windows and UNIX Administrator's Guide* for the encryption options.

● Conflict Detection and Resolution (CDR). First, see "Configuring Oracle GoldenGate for active-active high availability" in the *Oracle GoldenGate for Windows and UNIX Administrator's Guide* for how to configure the source database and Oracle GoldenGate processes to support this feature.

● Other new parameters listed in the Release Notes. First see their documentation in the *Oracle GoldenGate for Windows and UNIX Reference Guide*.

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 Replicat.

```
GGSCI> STOP REPLICAT <group>
```

11. Copy the edited new parameter files back to the original names.

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>
```

Deploying integrated capture mode for an Oracle source database

These instructions assume:

● You read the information about integrated capture in the *Release Notes* and in the *Oracle GoldenGate for Oracle Installation and Setup Guide*.

● You know whether you want the logmining server, from which Extract receives logical change records, to be installed at the local source database or at a downstream database, and that you understand the RDBMS versions that each of those configurations requires.

> **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 on the source system.

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 file to a new name.

3. Add integrated capture parameters: These parameters are set with the TRANLOGOPTIONS INTEGRATEDPARAMS <params> [,...] parameter and are as follows:

❍ max_sga_size: Specifies the amount of SGA memory that is used by the database logmining server. Can be a positive integer in megabytes. The default is 1 GB if streams_pool_size is greater than 1 GB; otherwise, it is 75% of streams_pool_size. As best practice, keep 25 percent of the Streams pool available to the Extracts running in integrated capture mode. For example, if there are three Extracts in integrated capture mode, set stream_pool_size to the following: 3 GB + (3 GB * 0.25) = 3.75 GB

❍ parallelism: Specifies the number of processes supporting the database logmining server. Can be a positive integer. The default is 0.

❍ downstream_real_time_mine: Set this parameter to Y if you are using a downstream mining database AND you want integrated capture to be performed in real-time. Otherwise, omit it because the default is N.

4. (Downstream mining deployment only) To use a downstream mining deployment, prepare that database according to the instructions in the appendix entitled "Configuring a downstream mining database for integrated capture" in the *Oracle GoldenGate for Oracle Installation and Setup Guide v.11.2.1.0.2*, including creating the downstream mining user.

5. Prepare the redo logs: To support the upgrade to integrated capture, the transaction log that contains the start of the oldest open transaction must be available on the source or downstream mining system. 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. Review these options in the *Oracle GoldenGate for Windows and UNIX Reference Guide* before using them.

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

6. The Extract user in the source database must have at least the following privileges in the source database. Grant any that do not exist.

```
grant become user to gguser;
grant select any table to gguser;
grant select any dictionary to gguser;
grant execute on utl_file to gguser;
grant create session, alter session to gguser;
grant flashback any table to gguser;
grant execute on dbms_flashback to gguser;
grant alter any table to gguser;
grant select any transaction to gguser;
grant connect, resource to gguser;
```

7. Log in to the mining database as sysdba and execute the following PL/SQL to grant privileges to the mining user.

```
SQL> execute DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE(
grantee=>'gguser');
```

8. In GGSCI, use the following commands to log in to the source database and to the mining database, if using a downstream deployment.

Command for source database:

```
GGSCI> DBLOGIN USERID <user>, PASSWORD <password>, [<encryption
options>]
```

Command for downstream mining database, if applicable:

```
GGSCI> MININGDBLOGIN USERID <user>, PASSWORD <password>, [<encryption
options>]
```

9. In GGSCI, issue the following command until it confirms that the Extract is ready to be upgraded. If this command does not show that Extract can be upgraded, contact Oracle Support.

```
GGSCI> INFO EXTRACT <group> UPGRADE
```

10. In GGSCI, stop Extract.

```
GGSCI> STOP EXTRACT <group>
```

11. In GGSCI, issue the following command to upgrade the Extract to integrated capture mode.

```
GGSCI> ALTER EXTRACT <group> UPGRADE INTEGRATED TRANLOG
```

12. In GGSCI, issue the following command to register Extract with the database. This command creates the database logmining server in the mining database and registers Extract with the components that support integrated capture.

```
GGSCI> REGISTER EXTRACT <group> DATABASE
```

13. In GGSCI, start Extract.

```
GGSCI> START EXTRACT <group
```

Deploying the new globalization support

The new globalization support is documented in the *Oracle GoldenGate for Windows and UNIX Administrator's Guide* in the following chapters. It is recommended that you review these chapters to understand the changes required for this feature, and then follow the deployment steps in this section.

Chapter 2 "Oracle GoldenGate globalization support"

Overview of the globalization support.

Chapter 3 "Getting started with the Oracle GoldenGate process interfaces"

● Character set support for the command interface
● Character set support for parameter files

. . . .

- Supported characters in object names
- How to specify object names, including the new case-sensitivity support, new question-mark wildcard support, new fallback name matching behavior, and support for wildcard mapping from pre-11.2.1 trails
- Applying SQL-92 rules for names and literals in parameter files by using the USEANSISQLQUOTES parameter in the GLOBALS file

Chapter 11 "Mapping and manipulating data"

- Converting between character sets
- Preserving locale
- Supporting case and special characters in column mapping

Chapter 13 "Associating replicated data with metadata"

- How differences in character sets affect where data conversion can be performed (source, target, intermediary system)
- The affect of the character set of the definitions file when transferring it to a remote system that has a different operating system character set from the local system
- How to specify a character set for a new definitions file other than the default of the operating system
- How to use the UPDATECS input parameter to convert the character set of an existing definitions file to one that is compatible with the remote system
- How to use the NOEXTATTR input parameter to generate a definitions file that is backward compatible with an earlier version of Oracle GoldenGate

Chapter 16 "Customizing Oracle GoldenGate processing"

- See this chapter for an overview of globalization support for user exits.
- See the release notes for a list of these new features for user exits.
- See the reference guide for details about the new functions that support globalization.

Other information source

See the Oracle GoldenGate v11.2.1.0.2 Release Notes for parameter changes that apply specifically to your database.

To deploy the globalization features

These steps apply to all processes for which you want to deploy globalization support.

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, and to add quote marks to object names, as appropriate for the way that you want to implement globalization support. Do not change group names or the trail name.

4. Run the Defgen utility on the source system to create a new definitions file with a different name from the original one.

5. Transfer the new definitions file to the target system.

6. Specify the new definitions file with the SOURCEDEFS parameter in the new Replicat parameter file.

7. Stop user activity on the source tables.

8. In GGSCI on the source system, use 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
   ```

9. In GGSCI, issue the SEND EXTRACT command 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
   ```

10. In GGSCI, stop Extract.

    ```
    GGSCI> STOP EXTRACT <group>
    ```

11. 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
    ```

12. In GGSCI, stop Replicat.

    ```
    GGSCI> STOP REPLICAT <group>
    ```

13. Copy the edited parameter files back to the original names.

14. Copy the new definitions file to the name of the old definitions file.

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

    ```
    GGSCI> START REPLICAT <group>
    ```

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

    ```
    GGSCI> START EXTRACT <group>
    ```

17. Allow user activity on the source tables.

Obtaining Customer Support

Oracle GoldenGate is fully committed to providing timely, high-quality service to all of our customers. Our support analysts can be contacted via e-mail, and live assistance is available over the phone or through interactive online support sessions. For support contact information please go to http://support.oracle.com.