

Oracle® GoldenGate
Teradata Installation and Setup Guide
11g Release 1 Patch Set 1 (11.1.1.1)
E21510-01

September 2011

ORACLE®

E21510-01

Copyright © 1995, 2011 Oracle and/or its affiliates. All rights reserved.

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

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

If this software or related documentation 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 RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

Contents



- Chapter 1 System requirements and preinstallation instructions1**
 - Overview of Oracle GoldenGate for Teradata.....1
 - What this documentation provides1
 - Supported Platforms2
 - Operating system requirements2
 - Replication server..... 2
 - Disk requirements 2
 - Relay Services Gateway (RSG) vprocs 3
 - Network 3
 - Operating system privileges..... 3
 - Other programs 3
 - Database requirements.....4
 - Database configuration..... 4
 - Database user for Oracle GoldenGate processes 4
 - Supported Teradata data types4
 - Supported objects and operations for Teradata.....8
 - DML 8
 - DDL..... 8
 - Supported and non-supported object names and case..... 10
 - Object names and owners..... 10
 - Case sensitivity..... 10
 - Supported characters..... 10
 - Non-supported characters 12
- Chapter 2 Installing Oracle GoldenGate 13**
 - Installation overview 13
 - Downloading Oracle GoldenGate..... 13
 - Setting library paths for dynamic builds on UNIX 14
 - Installing Oracle GoldenGate on Linux and UNIX..... 15
 - Installing Oracle GoldenGate into a Linux or UNIX cluster 15
 - Installing the Oracle GoldenGate files..... 15
 - Installing Oracle GoldenGate on Windows and Windows Cluster 16
 - Installing Oracle GoldenGate into a Windows Cluster 16
 - Installing the Oracle GoldenGate files..... 16
 - Specifying a custom Manager name 16



| | | |
|-------------------|---|-----------|
| | Installing Manager as a Windows service | 17 |
| | Adding Oracle GoldenGate as a Windows cluster resource | 18 |
| | Configuring Manager and other processes | 19 |
| Chapter 3 | Preparing the system for Oracle GoldenGate | 21 |
| | Preparing tables for processing | 21 |
| | Disabling triggers and cascade constraints | 21 |
| | Assigning row identifiers | 21 |
| | Configuring character sets | 22 |
| | Creating a Teradata replication group | 22 |
| | Activating DDL capture by the Teradata TAM | 23 |
| | Configuring the TAM initialization file | 24 |
| Chapter 4 | Configuring an Extract commit mode | 28 |
| | Maximum protection mode | 28 |
| | Recommended maximum protection configuration | 28 |
| | Maximum performance mode | 32 |
| | Recommended maximum performance configuration | 32 |
| | Additional Oracle GoldenGate configuration guidelines | 35 |
| | Handling massive update and delete operations | 35 |
| | Preventing multiple connections | 35 |
| | Improving Replicat performance over ODBC | 35 |
| | Performing initial synchronization | 36 |
| Chapter 5 | Modifying objects in the Oracle GoldenGate configuration | 37 |
| | Deleting an Extract group | 37 |
| | Adding a table to an existing Extract group | 37 |
| | Moving a table to a new Extract group | 38 |
| | Modifying columns of a table | 39 |
| Chapter 6 | Uninstalling Oracle GoldenGate | 41 |
| | Uninstalling Oracle GoldenGate from Linux or UNIX | 41 |
| | Uninstalling Oracle GoldenGate from Windows (non-cluster) | 41 |
| | Uninstalling Oracle GoldenGate from Windows Cluster | 42 |
| Appendix 1 | Oracle GoldenGate installed components | 44 |
| | Oracle GoldenGate Programs and Utilities | 44 |
| | Oracle GoldenGate subdirectories | 46 |
| | Other Oracle GoldenGate files | 48 |
| | Oracle GoldenGate checkpoint table | 53 |
| Index | | 54 |

CHAPTER 1

System requirements and preinstallation instructions

.....

This chapter contains the requirements for the system and database resources that support Oracle GoldenGate.

Overview of Oracle GoldenGate for Teradata

Oracle GoldenGate supports the replication of data as follows:

- between a Teradata source database (known as a *source server*) and a Teradata target database (known as a *subscriber server*).
- between Teradata databases and other supported database platforms.

In addition, Oracle GoldenGate replicates DDL operations between identical Teradata source and subscriber servers.

Oracle GoldenGate operates on a *replication server*, which is separate from the servers that contain the Teradata databases. Oracle GoldenGate receives transactional changes or table-copy operations from the Teradata Change Data Capture (CDC) facility on the source server, and then transmits it to the subscriber server using ODBC over a TCP/IP connection. Communication between the CDC and Oracle GoldenGate is managed by the Teradata Access Module (TAM).

Oracle GoldenGate for Teradata supports the filtering, mapping, and transformation of data unless noted otherwise in this documentation.

What this documentation provides

This documentation contains information that is specific to the setup of the Oracle GoldenGate solution within a Teradata environment. It assumes that the reader has a fundamental knowledge of the Teradata database and the Teradata Replication Solutions. It also assumes that the following have been configured properly:

- Relay Services Gateway (RSG)
- Change Data Capture (CDC)
- Teradata Access Module (TAM)
- Replication groups

To configure replication for the Teradata database, see the Teradata Replication Solutions documentation.

.....

Supported Platforms

To find out which Oracle GoldenGate builds are available for a specific combination of database version and operating system, log onto <http://support.oracle.com> and select the **Certifications** tab. For assistance, click **Tips for Finding Certifications**.

An e-mail and password are required to enter this site.

Operating system requirements

Replication server

- Install Oracle GoldenGate on a server that is separate from the one where the Teradata source and target databases are installed. This machine will be the *replication server*.
 - The replication server can exist in the same location as the source or target server, or it can be remote from one or both.
 - For replication between Teradata systems in remote locations, install the Oracle GoldenGate Extract process on a replication server at the source location, and install the Oracle GoldenGate Replicat process on a different replication server at the target location.
- To use Oracle GoldenGate in a bidirectional Teradata configuration, you can do either of the following:
 - Install Oracle GoldenGate on one replication server and use it to move data in both directions.
 - Install instances of Oracle GoldenGate on separate replication servers, each one handling data movement in one direction.
- If possible, install Oracle GoldenGate on a multi-node cluster server to minimize the impact of any Oracle GoldenGate outages that are caused by server failure.
- Install the Teradata Access Module (TAM) library into the root Oracle GoldenGate directory on the replication server. The TAM communicates with an Oracle GoldenGate API that is known as a *Vendor Access Module*, or *VAM*. The VAM passes transactional data changes to the Extract process. For instructions on pairing the correct TAM version with your Teradata version, and for configuring the TAM for use with the Teradata database and Oracle GoldenGate, see the *Teradata Replication Services Using Oracle GoldenGate* documentation. In general, the TAM version should match the database version.
- For additional configuration considerations, consult the *Teradata Replication Services Using Oracle GoldenGate* documentation before installing Oracle GoldenGate.

Disk requirements

The recommended hardware configuration for the Oracle GoldenGate replication server is:

- Four 300-GB disks
- 4 dual-core CPUs
- 8 GB of RAM

Assign the following free disk space:

- 50-150 MB, depending on the database and platform. This includes space for the compressed download file and space for the uncompressed files. You can delete the download file after the installation is complete.
- 40 MB for the working directories and binaries for each instance of Oracle GoldenGate that you are installing on the system. For example, to install two builds of Oracle GoldenGate into two separate directories, allocate 80 MB of space.
- To install Oracle GoldenGate into a cluster environment, install the Oracle GoldenGate binaries and files on a shared file system that is available to all cluster nodes.
- An additional 1 GB of disk space on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. You may need more or less than this amount, because the space that is consumed by the trails depends on the volume of data that will be processed. Start with 1 GB and adjust as needed.

Relay Services Gateway (RSG) vprocs

Replication tasks run on RSG vprocs on the source server for connections with the replication server. The connection implements the TCP/IP protocol. As of Teradata V12, each system node can have one RSG.

Network

- Configure the system to use TCP/IP services, including DNS.
- Configure the network with the host names or IP addresses of all systems that will be hosting Oracle GoldenGate processes and to which Oracle GoldenGate will be connecting. Host names are easier to use.
- Oracle GoldenGate requires some unreserved and unrestricted TCP/IP ports, the number of which depends on the number and types of processes in your configuration. See the Oracle GoldenGate *Windows and UNIX Administrator's Guide* for details on how to configure the Manager process to handle the required ports.
- Keep a record of the ports that you assigned to Oracle GoldenGate. You will specify them with parameters when configuring the Manager process.
- Configure your firewalls to accept connections through the Oracle GoldenGate ports.

Operating system privileges

The Manager process requires an operating system user that has privileges to control Oracle GoldenGate processes and to read, write, and purge files and subdirectories in the Oracle GoldenGate directory.

The Extract and Replicat processes require privileges to access the database.

Other programs

- Before installing Oracle GoldenGate on a Windows system, install and configure the Microsoft Visual C++ 2005 SP1 Redistributable Package. **Make certain it is the SP1 version of this package, and make certain to get the correct bit version for your server.** This package installs runtime components of Visual C++ Libraries. For more information, and to download this package, go to <http://www.microsoft.com>.

- Oracle GoldenGate fully supports virtual machine environments created with any virtualization software on any platform. When installing Oracle GoldenGate into a virtual machine environment, select a build that matches the database and the operating system of the virtual machine, not the host system.

Database requirements

Database configuration

- Install an appropriate ODBC (Open Database Connectivity) driver:
 - 12.00.00.01 or greater for the TTU 12.0 family, V12
 - 13.00.00.00 or greater for the TTU 13.0 family, V13.1
- Configure ODBC on each source and target system, including the creation of a data source name (DSN). A DSN stores information about how to connect to the database. See the *ODBC Driver for Teradata User Guide* for instructions.
- Create Teradata replication groups for the source tables. For instructions, see the *Teradata Replication Services Using Oracle GoldenGate* documentation.

Database user for Oracle GoldenGate processes

- Create a database user that is dedicated to Oracle GoldenGate. It can be the same user for all of the Oracle GoldenGate processes that must connect to a database:
 - Extract (source database)
 - Replicat (target database)
 - DEFGEN (source or target database)
- To preserve the security of your data, and to monitor Oracle GoldenGate processing accurately, do not permit other users, applications, or processes to log on as, or operate as, the Oracle GoldenGate database user.
- For each Teradata replication group, issue the following security grants to the Extract database user.

```
GRANT SELECT ON DBC.REPGROUP TO <user>;
GRANT SELECT ON DBC.TVM TO <user>;
GRANT SELECT ON DBC.DBASE TO <user>;
GRANT SELECT ON DBC.ERRORMSGs TO <user>;
GRANT SELECT ON DBC.TVFIELDS TO <user>;
GRANT SELECT ON DBC.INDEXES TO <user>;
GRANT SELECT ON DBC.INDOUBTRESLOG TO <user>;
GRANT REPLCONTROL TO <user>;
GRANT ALL ON <database> TO <user>;
GRANT ALL ON SYSUDTLIB TO <user> WITH GRANT OPTION;
```

Supported Teradata data types

The following table shows the Teradata data types that Oracle GoldenGate supports. Any limitations or conditions that apply follow this table.

Table 1 Supported data types by Oracle GoldenGate, per Teradata version

| Data type | v12 | v13 | v13.1 |
|------------------------|-----|-----|-------|
| BLOB | No | Yes | Yes |
| BYTEINT | Yes | Yes | Yes |
| VARBYTE | Yes | Yes | Yes |
| BIGINT | Yes | Yes | Yes |
| BYTEINT | Yes | Yes | Yes |
| DATE | Yes | Yes | Yes |
| DECIMAL - 18 and under | Yes | Yes | Yes |
| DECIMAL - 19 to 38 | No | Yes | Yes |
| DOUBLE PRECISION | Yes | Yes | Yes |
| FLOAT | Yes | Yes | Yes |
| INTEGER | Yes | Yes | Yes |
| NUMERIC - 18 and under | Yes | Yes | Yes |
| NUMERIC - 19 to 38 | No | Yes | Yes |
| REAL | Yes | Yes | Yes |
| SMALLINT | Yes | Yes | Yes |
| TIME | Yes | Yes | Yes |
| TIMESTAMP | Yes | Yes | Yes |
| INTERVAL | Yes | Yes | Yes |
| INTERVAL DAY | Yes | Yes | Yes |
| INTERVAL DAY TO HOUR | Yes | Yes | Yes |
| INTERVAL DAY TO MINUTE | Yes | Yes | Yes |
| INTERVAL DAY TO SECOND | Yes | Yes | Yes |
| INTERVAL HOUR | Yes | Yes | Yes |

Table 1 Supported data types by Oracle GoldenGate, per Teradata version

| Data type | v12 | v13 | v13.1 |
|---------------------------|-----|-----|-------|
| INTERVAL HOUR TO MINUTE | Yes | Yes | Yes |
| INTERVAL HOUR TO SECOND | Yes | Yes | Yes |
| INTERVAL MINUTE | Yes | Yes | Yes |
| INTERVAL MINUTE TO SECOND | Yes | Yes | Yes |
| INTERVAL MONTH | Yes | Yes | Yes |
| INTERVAL SECOND | Yes | Yes | Yes |
| INTERVAL YEAR | Yes | Yes | Yes |
| INTERVAL YEAR TO MONTH | Yes | Yes | Yes |
| CHAR | Yes | Yes | Yes |
| CLOB | No | Yes | Yes |
| CHAR VARYING | Yes | Yes | Yes |
| LONG VARCHAR | Yes | Yes | Yes |
| VARCHAR | Yes | Yes | Yes |
| GRAPHIC | Yes | Yes | Yes |
| LONG VARGRAPHIC | Yes | Yes | Yes |
| VARGRAPHIC | Yes | Yes | Yes |
| PERIOD (DATE) | No | Yes | Yes |
| PERIOD (TIME) | No | Yes | Yes |
| PERIOD (TIMESTAMP) | No | Yes | Yes |
| UDT | No | Yes | Yes |

Limitations of support for numeric data types

Numeric data types are fully supported between Teradata source and target databases. When replicating these data types from a different type of database to Teradata, truncation can occur if the source database supports a higher precision than Teradata does.

The support of range and precision for floating-point numbers depends on the host machine. In general, the precision is accurate to 16 significant digits, but you should review the database documentation to determine the expected approximations. Oracle

GoldenGate rounds or truncates values that exceed the supported precision.

Limitations of support for single-byte character data types

Single-byte character types are fully supported within a single-byte Latin character set between a Teradata source and Teradata targets, and between other databases and Teradata. A VARCHAR or CHAR column cannot have more than 32k-1 bytes. If using UTF-16, this is 16k-2 characters.

Conditions and limitations of support for multi-byte character data

- Install Oracle GoldenGate on a Windows or Linux replication server.
- Use the Teradata ODBC driver version 12.0.0.x or later.
- Do not use filtering, mapping, and transformation for multi-byte data types.
- Source Teradata tables can contain only CHAR, VARCHAR, INTEGER, SMALLINT, DATE, TIME, and TIMESTAMP columns. No other data types can be replicated while multi-byte data is being replicated.
- A CHAR or VARCHAR column cannot contain more than 32k-1 bytes. If using UTF-16, these columns cannot contain more than 16k-2 characters.
- Set the ODBC driver and the Teradata Access Module (TAM) to the UTF-16 character set in the initialization file (see page 24).
- When creating Replicat groups, use the NODBCHECKPOINT option with the ADD REPLICAT command. The Replicat database checkpointing feature does not support an ODBC driver that is set to the UTF-16 character set. Checkpoints will be maintained in the checkpoint file on disk.
- To support Unicode on Linux, Oracle GoldenGate must be configured in pass-through mode, and the ODBC Teradata Linux driver must be version 12.0.0.7 or higher, or 13.0.0.5 or higher. For more information about pass-through mode, see the PASSTHRU parameter in the Oracle GoldenGate *Windows and UNIX Reference Guide*.

Limitations of support for binary data types

No limitations. These data types are supported between a Teradata source and Teradata targets, and between other source databases and Teradata targets.

Limitations of support for large object data types

- To replicate large objects, at least TAM 13.0 and Oracle GoldenGate version 10.0 are required.
- To replicate UDTs, the target database must be Teradata Database 12.00.00.01 or greater or later.
- To replicate large objects from other databases to Teradata, use Teradata ODBC driver version 12.0 or higher on the target system. The target must support large objects that are delivered by ODBC.
- Enable the UseNativeLOBSupport flag in the ODBC configuration file. See the Teradata ODBC documentation.

Limitations of support for date data types

- Date types are fully supported between Teradata source and Teradata target databases. Additionally, INTERVAL is supported between Teradata and Oracle if the size of the target column is equal to, or greater than, that of the source.

- DATE, TIME, and TIMESTAMP are fully supported when replicated from a different type of source database to Teradata.
- TIME with TIMESZONE, TIMESTAMP with TIMEZONE, and INTERVAL are not supported from a different type of source database to Teradata.
- Oracle GoldenGate does not support negative dates.

Limitations of support for IDENTITY data types

IDENTITY must be configured as GENERATED BY DEFAULT AS IDENTITY on the target to enable the correct value to be inserted by Replicat. To include IDENTITY in a bi-directional replication configuration, the ranges of the values defined on the source and target systems must be disjoint, for example odd on one and even on the other.

Supported objects and operations for Teradata

DML

- Oracle GoldenGate supports the extraction and replication of Teradata DML (data manipulation language) for Teradata tables that contain rows of up to 512 KB in length.
- Oracle GoldenGate supports the maximum number of columns per table that is supported by the database.

DDL

A Teradata DDL statement can be replicated when it satisfies one of the following conditions:

- The DDL statement affects a table that is a member of a replication group.
- The DDL statement matches a user-defined replication rule.
- The DDL statement changes certain properties of a replication group.

Oracle GoldenGate supports the extraction and replication of the following Teradata DDL operations, up to 2 MB in statement length. At least TAM 13.0 is required, and both source and target databases must be Teradata Database 13.0 or later.

Table 2 Supported Teradata DDL

| Operations | Object |
|------------|---|
| CREATE | TABLE <table name> ¹ GLOBAL TEMPORARY TABLE <table name> ² [RECURSIVE] VIEW <view name> MACRO <macro name> HASH INDEX <index name> JOIN INDEX <index name> TRIGGER <trigger name> |

Table 2 Supported Teradata DDL (continued)

| Operations | Object |
|--|---|
| ALTER | TABLE |
| DROP | TABLE <table name> VIEW MACRO <macro name> HASH INDEX <index name> JOIN INDEX <index name> TRIGGER <trigger name> |
| RENAME | TABLE <table name> TO VIEW <view name> TO MACRO <macro name> TRIGGER <trigger name> |
| GRANT ... ON REVOKE ... ON | TABLE <table name> VIEW <view name> MACRO <macro name> |
| REPLACE | [RECURSIVE] VIEW <view name> TRIGGER <trigger name> MACRO <macro name> |
| COMMENT ON ³ | TABLE <table name> COLUMN <table name>.<column name> VIEW <view name> COLUMN <view name>.<column name> MACRO <macro name> TRIGGER <trigger name> |
| COLLECT STATISTICS ON ⁴ DROP STATISTICS ON | <table name> |

¹ DDL operations on tables that are members of a replication group are automatically captured.

² DDL statements that refer to the temporary materialized state of the table cannot be replicated.

³ Only COMMENT statements that create a user-defined description of an object in the data dictionary are captured.

⁴ Only the optimizer form that is used by the Optimizer for generating table access and join plans is captured.

NOTE The actual size limit of the DDL support is approximate, because the size will not only include the statement text but also Oracle GoldenGate maintenance overhead that depends on the length of the object name, the DDL type, and other characteristics of keeping a DDL record internally.

The following DDL statements that change the properties of replication groups will be replicated automatically.

- ALTER REPLICATION GROUP with ADD and/or DROP clauses
- CREATE REPLICATION RULESET

- REPLACE REPLICATION RULESET
- DROP REPLICATION RULESET

NOTE An ALTER REPLICATION GROUP statement that is used to generate a new security token will not be replicated.

See the Oracle GoldenGate *Windows and UNIX Administrator's Guide* for additional support limitations and configuration instructions.

Supported and non-supported object names and case

The following will help you verify whether the name of a supported object qualifies or disqualifies it for inclusion in an Oracle GoldenGate configuration.

Object names and owners

Source and target object names must be fully qualified in Oracle GoldenGate parameter files, as in `fin.emp`. Oracle GoldenGate supports character case as follows.

Case sensitivity

The following are general guidelines for the case-sensitivity of object names as it relates to Oracle GoldenGate. These guidelines may or may not apply to your databases, depending on whether the database or the underlying operating system supports case-sensitivity. Keep in mind that case-sensitivity (or lack thereof) may apply to the source database but not the target, or to the target but not the source.

- If the system or database is case-sensitive, Oracle GoldenGate supports the case sensitivity of database names, owner and schema names, object names, column names, and user names.
- If the system or database is case-insensitive (or is configured for case-insensitivity), Oracle GoldenGate converts all names to upper case.

To preserve case-sensitivity in an Oracle GoldenGate configuration

In Oracle GoldenGate parameter files, specify case-sensitive names exactly as they appear in the database. In `TABLE` and `MAP` parameters, enclose case-sensitive names in double quotes if the other database (the source or target of the case-sensitive objects) is not case-sensitive.

If replicating from a case-insensitive source to a case-sensitive target, enter the source names in upper case in the `Replicat MAP` statements, to reflect the fact that `Extract` writes them to the trail as uppercase.

For example:

```
MAP SALES.CUSTOMER, TARGET "Sales.Account";
```

Supported characters

Oracle GoldenGate supports alphanumeric characters in the names of objects, key columns, and non-key columns. Oracle GoldenGate also supports the following characters in columns that are not being used by Oracle GoldenGate as a key.

Table 3 Supported non-alphanumeric characters in object names and non-key column names¹

| Character | Description |
|-----------|---------------------------------------|
| ~ | Tilde |
| <> | Greater-than and less-than symbols |
| / | Forward slash |
| \ | Backward slash |
| ! | Exclamation point |
| @ | At symbol |
| # | Pound symbol |
| \$ | Dollar symbol |
| % | Percent symbol |
| ^ | Caret symbol |
| () | Open and close parentheses |
| _ | Underscore |
| - | Dash |
| + | Plus sign |
| = | Equal symbol |
| | Pipe |
| [] | Begin and end brackets |
| {} | Begin and end curly brackets (braces) |

¹ The type of key that is being used by Oracle GoldenGate depends on the definition of a given table and whether there are any overrides by means of a KEYCOLS clause. Oracle GoldenGate will use a primary key, if available, or a unique key/index (selection is dependent on the database). In the absence of those definitions, all columns of the table are used, but a KEYCOLS clause overrides all existing key types. For columns that are being used by Oracle GoldenGate as a key, the characters in the names must be valid for inclusion in a WHERE clause. This list is all-inclusive; a given database platform may or may not support all listed characters.

Non-supported characters

Oracle GoldenGate does not support the following characters in object or column names.

Table 4 Non-supported characters in object and column names¹

| Character | Description |
|-----------|--------------------------------|
| & | Ampersand |
| * | Asterisk |
| ? | Question mark |
| : | Colon |
| ; | Semi-colon |
| , | Comma |
| ' | Single quotes |
| “ ” | Double quotes |
| ‘ | Accent mark (Diacritical mark) |
| . | Period |
| | Space |

¹ This list is all-inclusive; a given database platform may or may not support all listed characters.

CHAPTER 2

Installing Oracle GoldenGate

.....

These instructions are for installing Oracle GoldenGate for the first time. To upgrade Oracle GoldenGate from one version to another, follow the instructions on:

<http://www.oracle.com/technology/software/products/goldengate/index.html>

Installing Oracle GoldenGate installs all of the components that are required to run and manage the processing (excluding any components required from other vendors, such as drivers or libraries) and it installs the Oracle GoldenGate utilities.

The installation process takes a short amount of time.

Installation overview

To install Oracle GoldenGate, the following steps are required:

- Downloading Oracle GoldenGate
- Setting library paths for dynamic builds
- Installing the Oracle GoldenGate software
- Configuring Manager and other processes

NOTE Before proceeding, make certain that you have reviewed the System Requirements in this guide.

Downloading Oracle GoldenGate

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

1. Navigate to <http://edelivery.oracle.com>.
2. On the **Welcome** page:
 - Select your language.
 - Click **Continue**.
3. On the **Export Validation** page:
 - Enter your identification information.
 - Accept the **Trial License Agreement** (even if you have a permanent license).
 - Accept the **Export Restrictions**.
 - Click **Continue**.
4. On the **Media Pack Search** page:
 - Select the **Oracle Fusion Middleware** Product Pack.
 - Select the platform on which you will be installing the software.
 - Click **Go**.

5. In the **Results List**:
 - Select the Media Pack that you want to download.
 - Click **Continue**.
6. On the **Download** page:
 - Click **Download** for each component that you want. Follow the automatic download process to transfer the mediapack.zip file to your system.

NOTE Before installing the software, review the release notes for any new features, new requirements, or bug fixes that affect your current configuration. Review the readme file for known issues.

Setting library paths for dynamic builds on UNIX

Oracle GoldenGate uses shared libraries. When you install Oracle GoldenGate on a UNIX system, the following must be true *before you run GGSCI or any other Oracle GoldenGate process*.

1. Make certain that the database libraries are added to the shared-library environment variables of the system. This procedure is usually performed at database installation time. Consult your Database Administrator if you have any questions.
2. If you will be running an Oracle GoldenGate program from outside the Oracle GoldenGate installation directory on a UNIX system:
 - (Optional) Add the Oracle GoldenGate installation directory to the PATH environment variable.
 - (Required) Add the Oracle GoldenGate installation directory to the shared-libraries environment variable.

For example, given an Oracle GoldenGate installation directory of /users/ogg, the second command in the following example requires these variables to be set:

| Command | Requires GG libraries in environment variable? |
|------------------------|--|
| \$ users/ogg > ./ggsci | No |
| \$ users > ./ogg/ggsci | Yes |

To set the variables in Korn shell

```
PATH=<installation directory>:$PATH
export PATH
<shared libraries variable>=<absolute path of installation directory>:<shared libraries variable>
export <shared libraries variable>
```

To set the variables in Bourne shell

```
export PATH=<installation directory>:$PATH
export <shared libraries variable>=<absolute path of installation directory>:<shared libraries variable>
```

To set the variables in C shell

```
setenv PATH <installation directory>:$PATH  
setenv <shared libraries variable> <absolute path of installation directory>:${shared libraries variable}
```

Where: <shared libraries variable> is one of the following:

UNIX/Linux library path variables per platform

| Platform ¹ | Environment variable |
|-----------------------|------------------------------|
| IBM AIX | LIBPATH |
| IBM z/OS | |
| HP-UX | SHLIB_PATH |
| Sun Solaris | LD_LIBRARY_PATH ² |
| HP Tru64 (OSF/1) | |
| LINUX | |

¹ A specific platform may or may not be supported by Oracle GoldenGate for your database.

² In 64-bit environments with 32-bit Oracle databases, Oracle GoldenGate requires the LD_LIBRARY_PATH to include the 32-bit Oracle libraries.

Example `export LD_LIBRARY_PATH=/ggs/10.0:$LD_LIBRARY_PATH`

NOTE To view the libraries that are required by an Oracle GoldenGate process, use the `ldd <process>` shell command before starting the process. This command also shows an error message for any that are missing.

Installing Oracle GoldenGate on Linux and UNIX

Installing Oracle GoldenGate into a Linux or UNIX cluster

Follow these guidelines if installing into a cluster configuration:

- Install the Oracle GoldenGate binaries and files on a file system that is available to all cluster nodes, according to the directions that follow.
- After installing Oracle GoldenGate, configure the Manager process within the cluster application, as directed by the cluster documentation, so that Oracle GoldenGate will fail over properly with the other applications.

Installing the Oracle GoldenGate files

1. Extract the Oracle GoldenGate mediapack.zip file to the system and directory where you want Oracle GoldenGate to be installed.
2. Run the command shell.
3. Change directories to the new Oracle GoldenGate directory.
4. From the Oracle GoldenGate directory, run the GGSCI program.

```
GGSCI
```

5. In GGSCI, issue the following command to create the Oracle GoldenGate working directories.

```
CREATE SUBDIRS
```

6. Issue the following command to exit GGSCI.

```
EXIT
```

Installing Oracle GoldenGate on Windows and Windows Cluster

Installing Oracle GoldenGate into a Windows Cluster

1. Log into one of the nodes in the cluster.
2. Choose a drive for the Oracle GoldenGate installation location. This drive must be a resource within the same cluster group that contains the database instance.
3. Ensure that this cluster group is owned by the cluster node that you are logging into.
4. Install Oracle GoldenGate according to the following instructions.

Installing the Oracle GoldenGate files

1. Unzip the downloaded file(s) by using WinZip or an equivalent compression product.
2. Move the files in binary mode to a folder on the drive where you want to install Oracle GoldenGate. *Do not* install Oracle GoldenGate into a folder that contains spaces in its name, even if the path is in quotes. For example:

C:\“Oracle GoldenGate” is not valid.

C:\Oracle_GoldenGate is valid.

3. From the Oracle GoldenGate folder, run the GGSCI program.
4. In GGSCI, issue the following command to create the Oracle GoldenGate working directories.

```
CREATE SUBDIRS
```

5. Issue the following command to exit GGSCI.

```
EXIT
```

Specifying a custom Manager name

You must specify a custom name for the Manager process if either of the following is true:

- You want to use a name for Manager other than the default of GGSMDR.
- There will be multiple Manager processes running as Windows services on this system, such as one for the Oracle GoldenGate replication software and one for Oracle GoldenGate Veridata. Each Manager on a system must have a unique name. Before proceeding further, note the names of any local Manager services.

To specify a custom Manager name

1. From the directory that contains the Manager program, run GGSCI.

2. Issue the following command.

```
EDIT PARAMS ./GLOBALS
```

NOTE The ./ portion of this command must be used, because the GLOBALS file must reside at the root of the Oracle GoldenGate installation file.

3. In the file, add the following line, where <name> is a one-word name for the Manager service.

```
MGRSERVNAME <name>
```

4. Save the file. The file is saved automatically with the name GLOBALS, *without a file extension*. Do not move this file. It is used during installation of the Windows service and during data processing.

Installing Manager as a Windows service

By default, Manager is not installed as a service and can be run by a local or domain account. However, when run this way, Manager will stop when the user logs out. When you install Manager as a service, you can operate it independently of user connections, and you can configure it to start manually or at system start-up. Installing Manager as a service is required on a Windows Cluster, but optional otherwise.

To install Manager as a Windows service

1. (Recommended) Log on as the system administrator.
2. Click **Start > Run**, and type **cmd** in the **Run** dialog box.
3. From the directory that contains the Manager program that you are installing as a service, run the **install** program with the following syntax:

```
install <option> [...]
```

Where: <option> is one of the following:

Table 5 INSTALL options

| Option | Description |
|-----------|--|
| ADDEVENTS | Adds Oracle GoldenGate events to the Windows Event Manager. By default, Oracle GoldenGate errors are generic. To produce more specific error content, copy the following files from the Oracle GoldenGate installation directory to the SYSTEM32 directory. category.dll ggsmg.dll |

Table 5 INSTALL options (continued)

| Option | Description |
|------------------------|--|
| ADDSERVICE | <p>Adds Manager as a service with the name that is specified with the MGRSERVNAME parameter in the GLOBALS file, if one exists, or by the default of GGSMGR. ADDSERVICE configures the service to run as the Local System account, the standard for most Windows applications because the service can be run independently of user logins and password changes. To run Manager as a specific account, use the USER and PASSWORD options.¹</p> <p>The service is installed to start at system boot time (see AUTOSTART). To start it after installation, either reboot the system or start the service manually from the Services applet of the Control Panel.</p> |
| AUTOSTART | Sets the service that is created with ADDSERVICE to start at system boot time. This is the default unless MANUALSTART is used. |
| MANUALSTART | Sets the service that is created with ADDSERVICE to start manually through GGSCI, a script, or the Services applet of the Control Panel. The default is AUTOSTART. |
| USER <name> | <p>Specifies a domain user account that executes Manager. For <name>, include the domain name, a backward slash, and the user name, for example HEADQT\GGSMGR.</p> <p>By default, the Manager service is installed to use the Local System account.</p> |
| PASSWORD <password> | Specifies the password for the user that is specified with USER. |

¹ A user account can be changed by selecting the Properties action from the Services applet of the Windows Control Panel.

4. (Windows Server 2008) If Windows User Account Control (UAC) is enabled, you are prompted to allow or deny the program access to the computer. Select **Allow** to enable the **install** program to run. This installs the Manager service with a local system account running with administrator privileges. No further UAC prompts will be encountered when running Manager if installed as a service.

NOTE If Manager is not installed as a service, Oracle GoldenGate users will receive a UAC prompt to confirm the elevation of privileges for Manager when it is started from the GGSCI command prompt. Running other Oracle GoldenGate programs also triggers a prompt.

Adding Oracle GoldenGate as a Windows cluster resource

If you installed Oracle GoldenGate into a cluster, follow these instructions to establish Oracle GoldenGate as a cluster resource and configure the Manager service correctly on all nodes.

1. In the Cluster Administrator, select **File>New>Resource**.
2. In the New Resource dialog box, provide a descriptive name for the Oracle GoldenGate Manager (need not be its actual name). For Resource Type, select Generic Service. For Group, select the group that contains the database instance to which Oracle GoldenGate will connect.

3. Click **Next**.
4. In the Possible Owners dialog box, select the nodes on which Oracle GoldenGate will run.
5. Click **Next**.
6. In the GGS Manager Service Properties dialog box, click the Dependencies tab, and add the following to the Resource dependencies list:
 - The database resource
 - The disk resource that contains the Oracle GoldenGate directory
 - The disk resource that contains the database transaction log files
 - The disk resource that contains the database transaction log backup files
7. Click **Apply**, then **OK**.
8. In the Generic Service Parameters dialog box, type either the default Manager service name of GGSMGR or the custom name, if one is specified with MGRSERVNAME in the GLOBALS file.
9. Click **Next**.
10. Click **Finish** to exit the wizard.
11. In the Cluster Administrator tree, right-click the Manager resource and then select Properties.
12. Click the Advanced tab, and then deselect Affect the Group. This is a recommendation, but you can configure it as needed for your environment.
13. Click **Apply**.
14. Bring the cluster resource online to verify that it was installed correctly.
15. Take the resource offline again.
16. Move the group to the next node in the cluster. When the group moves successfully to the second node, the Manager resource should still be offline.
17. Log onto the second node.
18. Install Oracle GoldenGate Manager as a service on this node by running the **install** program as you did on the previous node. If you created a custom name for Manager in the GLOBALS file, that name will be used.
19. Bring the resource online to verify that it is running correctly on this node.
20. Repeat from step 16 for each additional node in the cluster.

Configuring Manager and other processes

To configure Oracle GoldenGate to support your business requirements, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*. It contains instructions to:

- Configure the Manager process with a TCP/IP port and other optional parameters that control dynamic port assignments, trail file maintenance, automatic startup, and other properties.
- Configure Extract and Replicat processes to support reporting, high availability, disaster recovery, and other topologies.

- Configure security to control user access, file security, and data encryption.
- Configure integration, manipulation, and conversion features that enable you to customize Oracle GoldenGate and support the delivery of data across heterogeneous environments.
- Configure utilities and other tools that support Oracle GoldenGate.

CHAPTER 3

Preparing the system for Oracle GoldenGate

.....

Preparing tables for processing

The following table attributes must be addressed in an Oracle GoldenGate environment.

Disabling triggers and cascade constraints

Disable triggers, cascade delete constraints, and cascade update constraints on the target tables, or alter them to ignore changes made by the Oracle GoldenGate database user. Oracle GoldenGate replicates DML that results from a trigger or cascade constraint. If the same trigger or constraint gets activated on the target table, it becomes redundant because of the replicated version, and the database returns an error. Consider the following example, where the source tables are “emp_src” and “salary_src” and the target tables are “emp_targ” and “salary_targ.”

1. A delete is issued for emp_src.
2. It cascades a delete to salary_src.
3. Oracle GoldenGate sends both deletes to the target.
4. The parent delete arrives first and is applied to emp_targ.
5. The parent delete cascades a delete to salary_targ.
6. The cascaded delete from salary_src is applied to salary_targ.
7. The row cannot be located because it was already deleted in step 5.

Assigning row identifiers

Oracle GoldenGate requires some form of unique row identifier on the source and target tables to locate the correct target rows for replicated updates and deletes.

How Oracle GoldenGate determines the kind of row identifier to use

Unless a KEYCOLS clause is used in the TABLE or MAP statement, Oracle GoldenGate selects a row identifier to use in the following order of priority:

1. Primary key
2. First unique key alphanumerically that does not contain a timestamp or non-materialized computed column.
3. If none of the preceding key types exist (even though there might be other types of keys defined on the table) Oracle GoldenGate constructs a pseudo key of all columns that the database allows to be used in a unique key, excluding those that are not supported by Oracle GoldenGate in a key or those that are excluded from the Oracle GoldenGate configuration.

.....

NOTE If there are other, non-usable keys on a table or if there are no keys at all on the table, Oracle GoldenGate logs an appropriate message to the report file. Constructing a key from all of the columns impedes the performance of Oracle GoldenGate on the source system. On the target, this key causes Replicat to use a larger, less efficient WHERE clause.

Using KEYCOLS to specify a custom key

If a table does not have one of the preceding types of row identifiers, or if you prefer those identifiers not to be used, you can define a substitute key if the table has columns that always contain unique values. You define this substitute key by including a KEYCOLS clause within the Extract TABLE parameter and the Replicat MAP parameter. The specified key will override any existing primary or unique key that Oracle GoldenGate finds. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

Configuring character sets

To ensure accurate character representation from one database to another, the following must be true:

- The character set of the target database must be a superset of the character set of the source database.
- If your client applications use different character sets, the database character set must be a superset of the character sets of the client applications. In this configuration, every character is represented when converting from a client character set to the database character set.

Creating a Teradata replication group

To create a replication group, use a CreateGroupStmtFile (Create Group Statement) file. By using a CreateGroupStmtFile, the correct identifier information for the replication group is automatically written to the tam.ini file. For more information, see the *Teradata Replication Services Using Oracle GoldenGate* documentation.

All objects that have dependencies on one another must be specified in the same replication group. A transaction must be wholly contained within the same replication group.

To create a Create Group Statement file

1. Use a text editor to create a text file.
2. Add the following lines:
 - The Teradata command "create replication group"
 - The name of the Teradata replication group
 - The table list that is associated with the replication group. A table can only be associated with one replication group. Only one replication group is allowed per Oracle GoldenGate Extract group.
3. Save the file with the suffix .sql in a directory within the Oracle GoldenGate installation directory, either at the root level or in a sub-directory that you create, such as "dirtam."

4. Specify the name of this file with the CreateGroupStmtFile parameter in the TAM initialization file. See “Configuring the TAM initialization file” on page 24 for more information.

Figure 1 Sample Create Group Statement file

```
Create Replication Group HRTRG1 (HR.EMPLOYEE,  
                                HR.EMP_INFO,  
                                HR.EMP_DEPT,  
                                HR.EMP_REVIEWS) ;
```

Activating DDL capture by the Teradata TAM

To specify DDL that you want the Teradata TAM to pass to Oracle GoldenGate, you create a replication ruleset statement for a replication group. A ruleset statement creates a set of one or more DDL-capture rules and associates them with the specified replication group. The rules are applied against the names and types of the target objects of the DDL statements as those operations are executed, making them immediately available for replication. DDL operations on tables that are members of a replication group are automatically captured.

NOTE The Teradata RSG must be properly configured. For more information, see the *Teradata Replication Services Using Oracle GoldenGate* documentation.

To activate DDL capture by the Teradata TAM

1. Log in as a user with REPLCONTROL privilege.
2. Create the replication ruleset.

```
[CREATE | REPLACE] REPLICATION RULESET <rule set name>  
[, DEFAULT]  
FOR <replication group name>  
AS <rule specification> [, <rule specification>]
```

Where: <rule specification> is:

```
<object kind> LIKE <string literal> [ESCAPE <character literal>]  
[AND NOT LIKE <string literal> [ESCAPE <character literal> ]
```

Where: <object kind> is:

```
TABLE | TEMPORARY TABLE | VIEW | MACRO | TRIGGER | INDEX
```

Usage requirements

- A replication group must contain explicitly defined members with a table list, or it must be an empty group that is associated with a replication ruleset.
- If the CREATE form of the statement is used, and a rule set with the same rule set name already exists for the specified replication group, the CREATE statement fails.
- If the REPLACE form of the statement is used, and a rule set with the same rule set name already exists for the specified replication group, the existing ruleset is replaced by the new rule set.

- If the DEFAULT option is used, all of the rules in the rule set are considered to be default rules. A default rule is applied if no other rule matches the object.
 - A non-default rule must not match the same object as a non-default rule that is associated with another replication group.
 - A default rule must not match the same object as a default rule that is associated with another replication group.
- The LIKE and NOT LIKE clauses specify pattern strings to match against the fully-qualified names of the objects of the SQL statements. The pattern strings can contain wildcard characters. The pattern and the optional ESCAPE character are used together in the same way as the LIKE predicate operator.

For more information about creating replication groups and rulesets, see the *Teradata Replication Services Using Oracle GoldenGate* documentation.

Example This example creates a rule set named “Sales1” for the replication group named “MyRepGroup” to capture any table that is created in the database named “SalesDB” and also capture any DDL that affects any view in “SalesDB” where the view name does not have the suffix “_s”.

```
CREATE REPLICATION GROUP MyRepGroup
CREATE REPLICATION RULESET Sales1 FOR MyRepGroup AS
TABLE LIKE 'SalesDB.%',
VIEW LIKE 'SalesDB.%' AND NOT LIKE '%z_s' ESCAPE 'z'
```

Note the use of an escape character to override the normal treatment of the underscore (“_”) as a wildcard.

To disable replication of DDL

Log in as a user with REPLCONTROL privilege and issue either of the following commands:

To disable the DDL but keep the replication group:

```
DROP REPLICATION RULESET <rule set name> FOR <replication group name>
```

To disable the DDL and delete the replication group:

```
DROP REPLICATION GROUP <group name> [ ( <table name> [, ... ] )]
```

Example DROP REPLICATION RULESET Sales1 FOR MyRepGroup;

Example DROP REPLICATION GROUP MyRepGroup

To configure Oracle GoldenGate DDL replication, see the Oracle GoldenGate *Windows and UNIX Administrator’s Guide*.

Configuring the TAM initialization file

The Teradata Access Module (TAM) provides the Extract process with links to the Teradata environment. To configure the TAM, create an initialization file.

To create a TAM initialization file

1. Use a text editor to create a text file.

2. Add the following required parameters to the file. Optional parameters that are listed also can be used as needed. See Figure 2 on page 27 for an example.

Table 6 Required TAM initialization file parameters

| Parameter | Description |
|--|---|
| Mode={Replication} | Required Specifies that the replication mode will be change data capture. To configure Extract for change data capture, see the Oracle GoldenGate <i>Windows and UNIX Administrator's Guide</i> . |
| DictOdbcConnString= <ODBC connection string for metadata> | Required The logon string of a user with access rights to the dictionary tables. |
| MgmtOdbcConnString= <ODBC connection string for management functions> | Required The logon string of a user with rights to execute management functions, such as CREATE REPLICATION GROUP. This logon requires the REPLCONTROL privilege. |
| ReplicationGroupName= <name> | Required for versions earlier than TAM12 The name of the replication group, as specified in the CREATE REPLICATION GROUP statement. |
| CreateGroupStmtFile= <name> | Required for TAM12 and later The name of the Create Group statement file that contains the CREATE REPLICATION GROUP statement for a new group. If the replication group was not created with a Create Group statement file, omit or comment out this parameter and use the GroupID and SecurityToken parameters. |
| GroupID=<ID> | Required if SecurityToken is used The ID of the replication group that is associated with the TAM. If GroupID is used, SecurityToken must be used. You can view the ID of any replication group with this command: HELP REPLICATION GROUP <rep group name>; For example: help replication group g1; The ID is a numerical ID in the Identifier column next to the name of the group. |

Table 6 Required TAM initialization file parameters (continued)

| Parameter | Description |
|---|--|
| SecurityToken=<token> | Required if replication group was not created with Create Group Statement The security token for the replication group that is associated with the TAM. If a group was created with a Create Group Statement file and you specify that file with the CreateGroupStmtFile parameter, the SecurityToken and GroupID parameters can be omitted because they will be generated automatically at runtime. |
| AltControlRSG=<IP or name> | Optional Specifies the IP address or name of a server that can take over as the control RSG when the primary one fails. |
| CharacterSet= {ASCII UTF16} | Required The character set for this replication group. There is more overhead associated with UTF16, so use it only when required |
| ControlRSG= <address>[:<port>] DataRSG1= <address>[:<port>] [DataRSG2= <address>[:<port>]] [...] | Required The RSG node addresses. Can be either a node name or an IP address and can be appended with an optional port number. The control RSG should be the highest numbered of the RSGs in the system. |
| Encryption= {None Control Data All} | Required The types of messages to encrypt. There is little difference in overhead between encrypting just data messages and encrypting both data and control messages. Use All to specify encryption or None to specify no encryption. |
| RsgTimeoutSec=<0-60 secs> | Optional The timeout, in seconds, when polling RSG for data. The default is 1. |
| RsgTimeoutMSec= <0-60000000 milliseconds> | Optional The timeout, in milliseconds, when polling RSG for data. The default is 0. |
| Tracing= { Debug Performance All None} | Optional The level of debug tracing. The default is None. |

Table 6 Required TAM initialization file parameters (continued)

| Parameter | Description |
|--|---|
| MaxProtTransCompleteThresh= <0-24 transactions> | Optional, valid for max protection mode The number of outstanding transactions that can be held after which an Oracle GoldenGate checkpoint must be requested. This is significant when operating in maximum protection mode. If there are numerous sessions that are applying transactions at a high rate, this parameter can be set to a higher value, such as 10. However, if the number of sessions is small, or if the rate of submission is low, you can set it to a lower value, such as 1-4, to minimize latency and maximize throughput. The default is 0. |
| Bidirectional {TRUE FALSE} | Optional Specifies whether or not before images of data are sent to Oracle GoldenGate. Must be TRUE if the Extract parameter GETUPDATEBEFORES is used. The default is FALSE, which sends only the after image of data to reduce the CDC overhead and communication bandwidth that is used. |

3. Save the file as an ASCII file named tam.ini within the Oracle GoldenGate installation directory, either at the root level or in a sub-directory that you create, such as “dirtam.”
4. Specify the name of this file with the VAM parameter in the Extract parameter file.

Figure 2 Sample TAM initialization file.

```
Mode=Replication
DictOdbcConnString=DSN=myDsn;uid=myUser;pwd=myPass
MgmtOdbcConnString=DSN=myDsn;uid=myUser;pwd=myPass
CreateGroupStmtFile=c:\GGS\Teradata\dirtam\hrtrg1.sql
CharacterSet=ASCII
ControlRSG=10.10.10.49:1152
DataRSG1=10.10.10.50:1153
DataRSG2=node3
DataRSG3=node4:1155
Encryption=All
```

CHAPTER 4

Configuring an Extract commit mode

.....

Oracle GoldenGate works with the Change Data Capture (CDC) component of a source Teradata database to operate in one of two modes:

- *maximum protection mode*
- *maximum performance mode*

The mode determines the commit protocol that is used and whether or not Oracle GoldenGate has an effect on the Teradata applications.

Maximum protection mode

Maximum protection mode is the recommended Oracle GoldenGate configuration for the Teradata database. Maximum protection mode incorporates Oracle GoldenGate into the production system by using a two-phase commit protocol between CDC on the source server and a primary Extract process on the replication server (through the TAM). The two-phase commit requires a source transaction to be committed to Oracle GoldenGate as well as to the source database, ensuring that no transactions are lost in transit or duplicated if communication is interrupted or a component fails.

In this mode, a transaction is *in doubt* (not committed) until the primary Extract acknowledges that it received all of the data and saved it to an Oracle GoldenGate VAM trail on disk.

- If CDC receives the acknowledgement within a given timeout period, it releases the transaction for commit to the application, for commit to the database, and for propagation by Oracle GoldenGate.
- If CDC does not receive the acknowledgement within a given timeout period, it rolls back the transaction, and the application user receives an error message.

The VAM trail is a series of files that work like a transaction log. It stores incoming data in the order that it is received, but not necessarily in transaction order. A secondary Extract process, known as a *VAM-sort Extract*, sorts the data into transaction order and either deletes a transaction if a rollback is received (because the two-phase commit failed) or releases it to a regular trail for further processing.

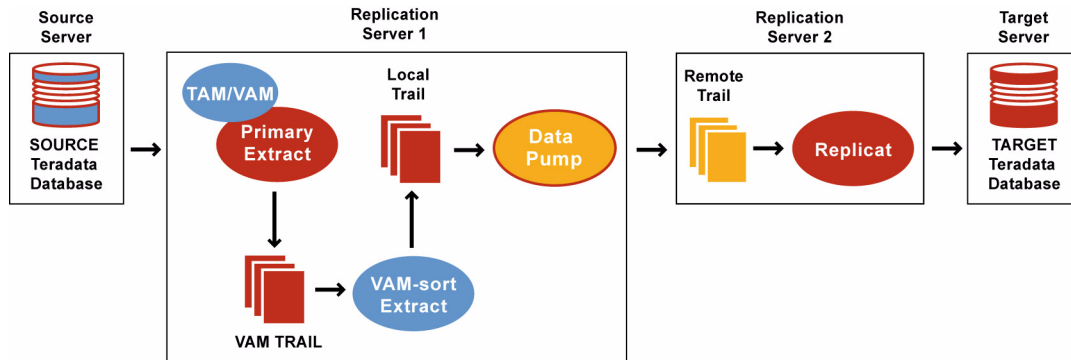
Recommended maximum protection configuration

Install the Extract and Replicat processes on separate replication servers and use a data pump with a local trail on the server where the Extract processes are installed. In this configuration, the primary Extract group captures the data, and then the VAM-sort Extract persists the sorted data to a regular Oracle GoldenGate trail on the local disk. A data-pump Extract reads this trail and sends the data across TCP/IP to a trail on the Replicat replication server, where it is read again by a Replicat process and applied to the target. If there is a failure of communication between the Extract server and the Replicat

.....

server, only the data pump is affected. The other two Extract processes can continue to do their work without running out of memory if the outage persists.

Figure 3 Recommended maximum protection configuration



To configure Extract in maximum protection mode

Perform these steps on the source replication server.

1. Configure the Manager process according to the instructions in the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
2. In the Manager parameter file, use the PURGEOLDEXTRACTS parameter to control the purging of files from the local trail.
3. Run GGSCI.
4. Create a primary Extract group. For documentation purposes, this group is called *ext*.
ADD EXTRACT <ext>, VAM
5. Create a local trail that is to be the VAM trail.
ADD EXTTRAIL <VAM_trail>, EXTRACT <ext>
 - Use the EXTRACT argument to link this trail to the primary Extract group. That Extract group creates this trail as a VAM trail.
6. Use the EDIT PARAMS command to create a parameter file for the primary Extract group. Include the following parameters plus any others that apply to your database environment.

NOTE Encryption options exist for encrypting the database password, the data in the trail, and the data sent across TCP/IP. See the Oracle GoldenGate *Windows and UNIX Administrator's Guide* for more information about security features.

```
-- Identify the Extract group:
EXTRACT <ext>
-- Specify database login information as needed for the database:
[SOURCEDB <dsn1>][USERID <user>[, PASSWORD <pw>]]
-- Specify the VAM trail:
EXTTRAIL <VAM_trail>
-- Specify that this Extract creates and writes to a VAM trail:
DSOPTIONS CREATETRANLOG
-- Specify library, TAM initialization file, and other parameters:
VAM <library>, PARAMS ("<init file>", "...")
-- Specify tables to be captured:
TABLE <owner>.<table>;
```

7. Create a VAM-sort Extract group to read the VAM trail. For documentation purposes, this group is called *extsort*.

```
ADD EXTRACT <extsort>, VAMTRAILSOURCE <VAM_trail>
```

8. Add a local trail to receive the sorted data.

```
ADD EXTTRAIL <local_trail>, EXTRACT <extsort>
```

- Use the EXTRACT argument to link this trail to the VAM-sort Extract group. A data pump group will read this trail.

9. Use the EDIT PARAMS command to create a parameter file for the VAM-sort Extract group. Include the following parameters plus any others that apply to your database environment.

```
-- Identify the Extract group:
EXTRACT <extsort>
-- Specify database login information as needed for the database:
[SOURCEDB <dsn1>][USERID <user>[, PASSWORD <pw>]]
-- Specify that this Extract reads a VAM trail and sorts the data:
DSOPTIONS SORTTRANLOG
-- Specify the local trail to receive the sorted data:
EXTTRAIL <local_trail>
-- Specify tables to be captured:
TABLE <owner>.<table>;
```

10. Create a data pump group to read the local trail and send the data to a remote trail on one of the following:

- The replication server where Replicat is running against a target Teradata database.
- A target server where Replicat is running against another database platform that is supported by Oracle GoldenGate.

```
ADD EXTRACT <pump>, EXTTRAILSOURCE <local_trail>
```

For documentation purposes, this group is called *pump*.

11. Add the remote trail.

```
ADD RMTTRAIL <remote_trail>, EXTRACT <pump>
```

- Use the EXTRACT argument to link the remote trail to the data pump group.

12. Create a parameter file for the data pump. Include the following parameters plus any others that apply to your database environment.

```
-- Identify the data pump group:  
EXTRACT <pump>  
-- Specify database login information as needed for the database:  
[SOURCEDB <dsn1>],[USERID <user>[, PASSWORD <pw>]]  
-- Specify the name or IP address of the remote system:  
RMTTHOST <target>, MGRPORT <portnumber>  
-- Specify the remote trail:  
RMTTRAIL <remote_trail>  
-- Allow mapping, filtering, conversion or pass data through as-is:  
[PASSTHRU | NOPASSTHRU]  
-- Specify tables to be captured:  
TABLE <owner>.<table>;
```

NOTE To use PASSTHRU mode, the names of the source and target objects must be identical. No column mapping, filtering, SQLEXEC functions, transformation, or other functions that require data manipulation can be specified in the parameter file. You can combine normal processing with pass-through processing by pairing PASSTHRU and NOPASSTHRU with different TABLE statements.

To configure Replicat

Perform these steps on the target replication server or target database system.

1. Configure the Manager process according to the instructions in the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
2. In the Manager parameter file, use the PURGEOLDEXTRACTS parameter to control the purging of files from the local trail.
3. Create a Replicat checkpoint table. There are multiple options for this purpose. For instructions, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
4. Create a Replicat group. For documentation purposes, this group is called *rep*.

```
ADD REPLICAT <rep>, EXTTRAIL <remote_trail>
```

- Use the EXTTRAIL argument to link the Replicat group to the remote trail that you specified for the data pump on the source server.

5. Use the EDIT PARAMS command to create a parameter file for the Replicat group. Include the following parameters plus any others that apply to your database environment.

```
-- Identify the Replicat group:
REPLICAT <rep>
-- State whether or not source and target definitions are identical:
SOURCEDEFS {<full_pathname> | ASSUMETARGETDEFS}
-- Specify database login information as needed for the database:
[TARGETDB <dsn2>,) [USERID <user id>[, PASSWORD <pw>]]
-- Specify error handling rules:
REPEROR (<error>, <response>)
-- Specify tables for delivery:
MAP <owner>.<table>, TARGET <owner>.<table>[, DEF <template name>];
```

To handle Multiload errors

Teradata Multiload does not participate in the full two-phase commit protocol of maximum protection mode. In a recovery situation, it is possible that Replicat could attempt to apply some updates twice. If a multiset table is affected, this could result in duplicate rows being created. Use the REPEROR parameter in the Replicat parameter file so that Replicat ignores duplicate rows.

Maximum performance mode

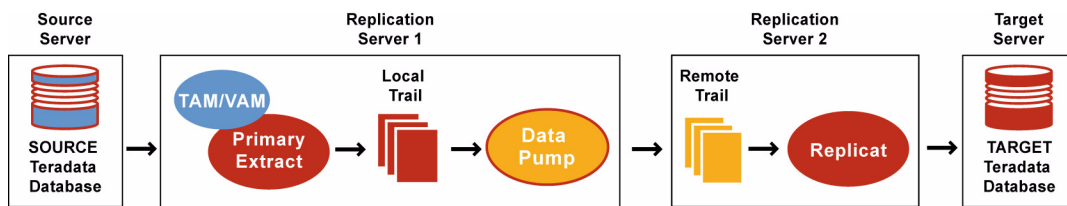
Maximum performance mode is faster and less intrusive than maximum protection mode, but it is less fault tolerant. When the source application issues a commit, CDC begins transmitting the data to the replication server, where it is buffered and sorted by Extract. When the data is finished being transmitted, CDC sends Extract a commit and releases the transaction for commit to the application and to the database.

The maximum performance configuration does not persist incoming data to disk, nor does it have an acknowledgement system between CDC and Extract that prevents data loss. If communication between the primary Extract and the Teradata source is interrupted, or if a component fails, the source and target data are no longer synchronized.

Recommended maximum performance configuration

Install the Extract and Replicat processes on separate replication servers, and use a data pump with a local trail on the Extract server. In this configuration, the primary Extract persists transactions to a local Oracle GoldenGate trail. A data pump Extract reads this trail and sends the data across TCP/IP to a trail on the Replicat server, where it is read again by Replicat and applied to the target. If there is a failure of communication between the Extract and Replicat replication servers, only the data pump is affected. The primary Extract can continue to write incoming data to disk instead of having to retain it in memory, which otherwise could be depleted in a longlasting outage.

Figure 4 Recommended maximum performance configuration



To configure Extract in maximum performance mode

Perform these steps on the source replication server.

1. Configure the Manager process according to the instructions in the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
2. In the Manager parameter file, use the PURGEOLDEXTRACTS parameter to control the purging of files from the local trail.
3. Run GGSCI.
4. Create a primary Extract group. For documentation purposes, this group is called *ext*.

```
ADD EXTRACT <ext>, VAM
```

5. Add a local trail.

```
ADD EXTTRAIL <local_trail>, EXTRACT <ext>
```

- Use the EXTRACT argument to link this trail to the primary Extract group.

6. Use the EDIT PARAMS command to create a parameter file for the primary Extract group. Include the following parameters plus any others that apply to your database environment.

NOTE Encryption options exist for encrypting the database password, the data in the trail, and the data sent across TCP/IP. See the Oracle GoldenGate *Windows and UNIX Administrator's Guide* for more information about security features.

```
-- Identify the Extract group:
EXTRACT <ext>
-- Specify database login information as needed for the database:
[SOURCEDB <dsn1>,[USERID <user>[, PASSWORD <pw>]]
-- Specify the local trail that this Extract writes to:
EXTTRAIL <local_trail>
-- Specify that this Extract is in maximum performance mode:
DSOPTIONS COMMITTEDTRANLOG, RESTARTAPPEND
-- Specify library, TAM initialization file, and other parameters:
VAM <library>, PARAMS ("<init file>", "...")
-- Specify tables to be captured:
TABLE <owner>.<table>;
```

7. Create a data pump group to read the local trail and send the data to a remote trail on one of the following:

- The replication server where Replicat is running against a target Teradata database.
- A target server where Replicat is running against another database platform that is supported by Oracle GoldenGate.

```
ADD EXTRACT <pump>, EXTTRAILSOURCE <local_trail>
```

For documentation purposes, this group is called *pump*.

8. Add the remote trail.

```
ADD RMTTRAIL <remote_trail>, EXTRACT <pump>
```

- Use the EXTRACT argument to link the remote trail to the data pump group.

9. Create a parameter file for the data pump. Include the following parameters plus any others that apply to your database environment.

```
-- Identify the data pump group:
EXTRACT <pump>
-- Specify database login information as needed for the database:
[SOURCEDB <dsn1>],[USERID <user>[, PASSWORD <pw>]]
-- Specify the name or IP address of the remote system:
RMTHOST <target>, MGRPORT <portnumber>
-- Specify the remote trail:
RMTTRAIL <remote_trail>
-- Allow mapping, filtering, conversion or pass data through as-is:
[PASSTHRU | NOPASSTHRU]
-- Specify tables to be captured:
TABLE <owner>.<table>;
```

NOTE To use PASSTHRU mode, the names of the source and target objects must be identical. No column mapping, filtering, SQLEXEC functions, transformation, or other functions that require data manipulation can be specified in the parameter file. You can combine normal processing with pass-through processing by pairing PASSTHRU and NOPASSTHRU with different TABLE statements.

To configure Replicat

Perform these steps on the target replication server or target database system.

1. Configure the Manager process according to the instructions in the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
2. In the Manager parameter file, use the PURGEOLDEXTRACTS parameter to control the purging of files from the local trail.
3. Create a Replicat checkpoint table. There are multiple options for this purpose. For instructions, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
4. Create a Replicat group. For documentation purposes, this group is called *rep*.

```
ADD REPLICAT <rep>, EXTTRAIL <remote_trail>
```

- Use the EXTTRAIL argument to link the Replicat group to the remote trail that you specified for the data pump on the source server.

5. Use the EDIT PARAMS command to create a parameter file for the Replicat group. Include the following parameters plus any others that apply to your database environment.

```
-- Identify the Replicat group:
REPLICAT <rep>
-- State whether or not source and target definitions are identical:
SOURCEDEFS <full_pathname> | ASSUMETARGETDEFS
-- Specify database login information as needed for the database:
[TARGETDB <dsn2>,) [USERID <user id>[, PASSWORD <pw>]]
-- Specify error handling rules:
REPERROR (<error>, <response>)
-- Specify tables for delivery:
MAP <owner>.<table>, TARGET <owner>.<table>[, DEF <template name>];
```

Additional Oracle GoldenGate configuration guidelines

The following are additional considerations to make once you have installed and configured your Oracle GoldenGate environment.

Handling massive update and delete operations

Operations that update or delete a large number of rows will generate discrete updates and deletes for each row on the subscriber database. This could cause a lock manager overflow on the Teradata subscriber system, and thus terminate the Replicat process.

To avoid these errors, you can do either of the following:

- Temporarily suspend replication for these operations and then perform them manually on the source and target systems. To suspend replication, use the following command, which suspends replication for that session only. The operations of other sessions on that table are replicated normally.

```
set session override replication on;
commit;
```
- Set the Replicat parameter MAXTRANSOPS to a value of less than 1000. This parameter splits large transactions into smaller ones.

Preventing multiple connections

By default, the Extract and Replicat processes create a new connection for catalog queries. You can prevent this extra connection by using the DBOPTIONS parameter with the NOCATALOGCONNECT option.

Improving Replicat performance over ODBC

To improve the throughput of the Replicat process, consider using multiple trails and parallel Replicat processes. Because each Replicat uses a single ODBC session, multiple sessions might be necessary to achieve reasonably high delivery rates. For configuration instructions, see the Oracle GoldenGate *Windows and UNIX Troubleshooting and Tuning Guide*.

Performing initial synchronization

Perform an initial synchronization of the source and target data before using Oracle GoldenGate to transmit transactional changes for the first time. The preferred methods for synchronizing two Teradata databases is to use any of the Teradata data loader utilities. The recommended utility is MultiLoad. To configure an initial load, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

CHAPTER 5

Modifying objects in the Oracle GoldenGate configuration

.....

This chapter contains instructions for performing some common maintenance tasks when using the Oracle GoldenGate replication solution.

Deleting an Extract group

To delete an Oracle GoldenGate Extract group, the Extract process must be decoupled from the Teradata replication group.

1. Start GGSCI.
2. While Extract is still running, issue this command:

```
SEND EXTRACT <group>, vammessages "control:terminate"
```
3. Stop Extract.

```
STOP EXTRACT <group>
```
4. Delete the Extract group forcefully.

```
DELETE EXTRACT <group> !
```
5. From any Teradata client, issue this command:

```
drop replication group <repgroup name>
```

Adding a table to an existing Extract group

1. Suspend activity on the source tables that are linked to Oracle GoldenGate.
2. Start GGSCI.
3. In GGSCI, issue this command:

```
INFO EXTRACT <group>
```
4. On the Checkpoint Lag line, verify whether there is any Extract lag. If needed, continue to issue INFO EXTRACT until lag is zero, which indicates that all of the transaction data so far has been processed.
5. While Extract is still running, issue this command:

```
SEND EXTRACT <group>, vammessages "control:terminate"
```
6. Stop the Extract group.

```
STOP EXTRACT <group>
```

7. From any Teradata client, issue this command to add the new table:

```
ALTER REPLICATION GROUP <group> ADD <database>.<table>
```
8. From any Teradata client, issue this command to generate a security token.

```
ALTER REPLICATION GROUP <group>
```
9. Edit the TAM initialization file and specify the security token with the SecurityToken parameter.
10. Edit the Extract parameter file to add a TABLE parameter that specifies the new table.

```
EDIT PARAMS <group>
```
11. Save and close the file.
12. In GGSCI, issue this command to start Extract:

```
START EXTRACT <group>
```
13. Allow activity on the source tables that are linked to Oracle GoldenGate.

Moving a table to a new Extract group

1. Suspend activity on the source database for all tables that are linked to Oracle GoldenGate.
2. Edit the current Teradata Create Group Statement file to remove the table from the CREATE REPLICATION GROUP statement.
3. Start GGSCI.
4. In GGSCI, issue this command for the current Extract group:

```
INFO EXTRACT <group>
```
5. On the Checkpoint Lag line, verify whether there is any Extract lag. If needed, continue to issue INFO EXTRACT until lag is zero, which indicates that all of the transaction data so far has been processed.
6. In GGSCI, issue this command:

```
SEND EXTRACT <group>, vammessages "control:terminate"
```
7. Stop the current Extract group.

```
STOP EXTRACT <group>
```
8. Edit the current Extract parameter file.

```
EDIT PARAMS <group>
```
9. Remove the TABLE parameter that contains the table.
10. From any Teradata client, issue this command to drop the replication group that contains the table that is being moved:

```
ALTER REPLICATION GROUP <group> DROP <table>
```

11. In GGSCI, issue this command to start the current Extract group, so that it can continue processing its assigned tables, minus the one that was moved:

```
START EXTRACT <group>
```

12. Add a new Extract group that contains a TABLE statement for the moved table, and then add the other processes, trails, and parameter files that are appropriate for the capture method that you are using. See “Configuring an Extract commit mode” on page 28 for instructions.
13. Create a new tam.ini file and a new Teradata Create Group Statement file that contains the table.
14. Start the new Extract group and any associated processes.

```
START EXTRACT <new_group>
```

15. Allow user activity to resume on all of the source tables that are linked to Oracle GoldenGate.

Modifying columns of a table

1. Suspend activity on the source database for all tables that are linked to Oracle GoldenGate.
2. Start GGSCI.
3. In GGSCI, issue this command for the Extract group:

```
INFO EXTRACT <group>
```

4. On the Checkpoint Lag line, verify whether there is any Extract lag. If needed, continue to issue INFO EXTRACT until lag is zero, which indicates that all of the transaction data so far has been processed.

5. While Extract is still running, issue this command:

```
SEND EXTRACT <group>, vammesssage "control:terminate"
```

6. Stop the Extract group.

```
STOP EXTRACT <group>
```

7. In GGSCI, issue this command for the Replicat group:

```
INFO REPLICAT <group>
```

8. On the Checkpoint Lag line, verify whether there is any Replicat lag. If needed, continue to issue INFO REPLICAT until lag is zero, which indicates that all of the data in the trail has been processed.

9. Stop the Replicat group.

```
STOP REPLICAT <group>
```

10. Perform the table modifications on the source and target databases.

11. Start the Extract and Replicat processes.

```
START EXTRACT <group>  
START REPLICAT <group>
```

12. Allow user activity to resume on all of the source tables that are linked to Oracle GoldenGate.

CHAPTER 6

Uninstalling Oracle GoldenGate

.....

This procedure assumes that you no longer need the data in the Oracle GoldenGate trails, and that you no longer need to preserve the current Oracle GoldenGate environment. To preserve your current environment and data, make a backup of the Oracle GoldenGate directory and all subdirectories before starting this procedure.

Uninstalling Oracle GoldenGate from Linux or UNIX

1. Run the command shell.
2. (Suggested) Log on as the system administrator, or as a user with permission to issue Oracle GoldenGate commands, and to delete files and directories from the operating system.
3. Run GGSCI.
4. While Extract is still running, issue the following command.

```
SEND EXTRACT <group>, vammesssage "control:terminate"
```
5. Stop Extract.

```
STOP EXTRACT <group>
```
6. Delete the Extract group forcefully.

```
DELETE EXTRACT <group> !
```
7. Stop the Manager process (where ! in the following command can be used to bypass the interactive prompt).

```
Stop Manager [!]
```
8. Exit GGSCI.
9. From any Teradata client, issue the following command.

```
drop replication group <repgroup name>
```
10. Remove the Oracle GoldenGate files by removing the installation directory.
11. Drop any Oracle GoldenGate-related objects from the database as needed.

Uninstalling Oracle GoldenGate from Windows (non-cluster)

1. (Suggested) Log on as the system administrator, or as a user with permission to issue Oracle GoldenGate commands, and to delete files and directories from the operating system.
 2. Run GGSCI.
-

3. While Extract is still running, issue the following command.

```
SEND EXTRACT <group>, vammessage "control:terminate"
```
4. Stop Extract.

```
STOP EXTRACT <group>
```
5. Delete the Extract group forcefully.

```
DELETE EXTRACT <group> !
```
6. Stop the Manager process (where ! in the following command can be used to bypass the interactive prompt).

```
Stop Manager [!]
```
7. Exit GGSCI.
8. From any Teradata client, issue the following command.

```
drop replication group <repgroup name>
```
9. Click **Start > Run**, and type cmd in the **Run** dialog box.
10. Change directories to the Oracle GoldenGate installation directory.
11. Run the install program using the following syntax.

```
install deleteevents deleteservice
```

This command stops Oracle GoldenGate events from being reported to the Windows Event Manager and removes the Manager service.
12. Delete the CATEGORY.DLL and GGSMMSG.DLL files from the Windows SYSTEM32 folder.
13. Delete the Oracle GoldenGate installation folder.
14. Drop any Oracle GoldenGate-related objects from the database as needed.

Uninstalling Oracle GoldenGate from Windows Cluster

1. Log into the node in the cluster that owns the cluster group that contains the Manager resource. Log on as the system administrator, or as a user with permission to issue Oracle GoldenGate commands and to delete files and directories from the operating system.
2. Run GGSCI.
3. While Extract is still running, issue the following command.

```
SEND EXTRACT <group>, vammessage "control:terminate"
```
4. Stop Extract.

```
STOP EXTRACT <group>
```
5. Delete the Extract group forcefully.

```
DELETE EXTRACT <group> !
```

6. Stop the Manager process (where ! in the following command can be used to bypass the interactive prompt).

```
Stop Manager [!]
```

7. Exit GGSCI.
8. Use the Cluster Administrator tool to take the Manager resource offline.
9. Right click the resource and select **Delete** to remove it.
10. Run the install program using the following syntax.

```
install deleteevents deleteservice
```

This command stops Oracle GoldenGate events from being reported to the Windows Event Manager and removes the Manager service.

11. Delete the CATEGORY.DLL and GGSMMSG.DLL files from the Windows SYSTEM32 folder.
12. Move the cluster group to the next node in the cluster, and repeat from step 10.
13. From any Teradata client, issue the following command.

```
drop replication group <repgroup name>
```

14. Delete the Oracle GoldenGate installation folder.
15. Drop any Oracle GoldenGate-related objects from the database as needed.

APPENDIX 1

Oracle GoldenGate installed components

.....

This appendix describes the programs, directories, and other components created or used by the Oracle GoldenGate software in the Oracle GoldenGate installation directory. Additional files not listed here might be installed on certain platforms. Files listed here might not be installed on every platform.

Oracle GoldenGate Programs and Utilities

This section describes programs installed in the root Oracle GoldenGate installation directory.

NOTE **Some programs may not exist in all installations.** For example, if only capture or delivery is supported by Oracle GoldenGate for your platform, the extract or replicat program will not be installed, respectively. Likewise, special files might be installed to support a specific database.

Table 7 Programs and utilities (continued)

| Program | Description |
|---------|--|
| cobgen | Generates source definitions based on COBOL layouts. Used for Oracle GoldenGate for Datawise on Stratus. |
| convchk | Converts checkpoint files to a newer version. |
| ddlcob | Generates target DDL table creation statements based on COBOL layouts. Used for Oracle GoldenGate for Datawise on Stratus. |
| ddlgen | Generates target database table definitions based on source database DDL. Used primarily on the NonStop platform. |
| defgen | Generates data definitions and is referenced by Oracle GoldenGate processes when source and target tables have dissimilar definitions. |
| emslnt | Sends event messages created by Collector and Replicat on Windows or UNIX systems to EMS on NonStop systems. |
| extract | Performs capture from database tables or transaction logs or receives transaction data from a vendor access module. |

Table 7 Programs and utilities (continued)

| Program | Description |
|--|---|
| ggmxinstall | Oracle GoldenGate installation script for the SQL/MX database. |
| ggsci | User interface to Oracle GoldenGate for issuing commands and managing parameter files. |
| ggsmgr.jcl ggsmgr.proc ggsmgrst.jcl ggsmgrst.proc | Start the Oracle GoldenGate Manager process from a batch job or the operator console on a z/OS system. Installed to support DB2 z/OS databases. |
| install | Installs Oracle GoldenGate as a Windows service and provides other Windows-based service options. |
| keygen | Generates data-encryption keys. |
| logdump | A utility for viewing and saving information stored in extract trails or files. |
| mgr | (Manager) Control process for resource management, control and monitoring of Oracle GoldenGate processes, reporting, and routing of requests through the GGSCI interface. |
| replicat | Applies data to target database tables. |
| reverse | A utility that reverses the order of transactional operations, so that Replicat can be used to back out changes from target tables, restoring them to a previous state. |
| server | The Collector process, an Extract TCP/IP server collector that writes data to remote trails. |
| vamserv | Started by Extract to read the TMF audit trails generated by TMF-enabled applications. Installed to support the NonStop SQL/MX database. |

Oracle GoldenGate subdirectories

This section describes the subdirectories of the Oracle GoldenGate installation directory and their contents.

NOTE Some directories may not exist in all installations.

Table 8 Subdirectories

| Directory | Description |
|-----------|--|
| br | Contains the checkpoint files for the bounded recover feature. |
| cfg | Contains the property and xml files that are used to configure Oracle GoldenGate Monitor. |
| dirchk | <p>Contains the checkpoint files created by Extract and Replicat processes, which store current read and write positions to support data accuracy and fault tolerance. Written in internal Oracle GoldenGate format.</p> <p>File name format is <group name><sequence number>.<ext> where <sequence number> is a sequential number appended to aged files and <ext> is either cpe for Extract checkpoint files or cpr for Replicat checkpoint files.</p> <p>Do not edit these files.</p> <p>Examples:</p> <p>ext1.cpe rep1.cpr</p> |
| dirdat | <p>The default location for Oracle GoldenGate trail files and extract files that are created by Extract processes to store extracted data for further processing by the Replicat process or another application or utility. Written in internal Oracle GoldenGate format.</p> <p>File name format is a user-defined two-character prefix followed by either a six-digit sequence number (trail files) or the user-defined name of the associated Extract process group (extract files).</p> <p>Do not edit these files.</p> <p>Examples:</p> <p>rt000001 finance</p> |
| dirdef | <p>The default location for data definitions files created by the DEFGEN utility to contain source or target data definitions used in a heterogeneous synchronization environment. Written in external ASCII. File name format is a user-defined name specified in the DEFGEN parameter file.</p> <p>These files may be edited to add definitions for newly created tables. If you are unsure of how to edit a definitions file, contact Oracle GoldenGate technical support.</p> <p>Example:</p> <p>defs.dat</p> |

Table 8 Subdirectories

| Directory | Description |
|-----------|--|
| dirjar | Contains the Java executable files that support Oracle GoldenGate Monitor. |
| dirout | This directory is not used any more. |
| dirpcs | <p>Default location for status files. File name format is <group>.<extension> where <group> is the name of the group and <extension> is either pce (Extract), pcr (Replicat), or pcm (Manager).</p> <p>These files are only created while a process is running. The file shows the program name, the process name, the port number, and the process ID.</p> <p>Do not edit these files.</p> <p>Examples: mgr.pcm ext.pce</p> |
| dirprm | <p>The default location for Oracle GoldenGate parameter files created by Oracle GoldenGate users to store run-time parameters for Oracle GoldenGate process groups or utilities. Written in external ASCII format. File name format is <group name/user-defined name>.prm or mgr.prm.</p> <p>These files may be edited to change Oracle GoldenGate parameter values after stopping the process. They can be edited directly from a text editor or by using the EDIT PARAMS command in GGSCI.</p> <p>Examples: defgen.prm finance.prm</p> |
| dirrec | Not used by Oracle GoldenGate. |
| dirrpt | <p>The default location for process report files created by Extract, Replicat, and Manager processes to report statistical information relating to a processing run. Written in external ASCII format.</p> <p>File name format is <group name><sequence number>.rpt where <sequence number> is a sequential number appended to aged files.</p> <p>Do not edit these files.</p> <p>Examples: fin2.rpt mgr4.rpt</p> |
| dirsql | Used by the TRIGGEN utility to store SQL scripts before TRIGGEN was deprecated. Currently used to store training scripts and any user-created SQL scripts that support Oracle GoldenGate. |

Table 8 Subdirectories

| Directory | Description |
|------------------|---|
| dirtmp | The default location for storing large transactions when the size exceeds the allocated memory size. Do not edit these files. |
| dirver | An Oracle GoldenGate Veridata directory. Not used unless this software is installed in the Oracle GoldenGate location. |
| dirwal | Contains the Oracle Wallet that supports Oracle GoldenGate Monitor. This directory is not installed until the utility that creates the wallet is run. |
| UserExitExamples | Contains sample files to help with the creation of user exits. |

Other Oracle GoldenGate files

This section describes other files, templates, and objects created or installed in the root Oracle GoldenGate installation directory.

NOTE Some files may not be installed in your environment, depending on the database and OS platform.

Table 9 Other files

| Component | Description |
|-----------------------|---|
| bcpfmt.tpl | Template for use with Replicat when creating a run file for the Microsoft BCP/DTS bulk-load utility. |
| bcrypt.txt | Blowfish encryption software license agreement. |
| cagent.dll | Contains the Windows dynamic link library for the Oracle GoldenGate Monitor C sub-agent. |
| category.dll | Windows dynamic link library used by the INSTALL program. |
| chkpt_<db>_create.sql | Script that creates a checkpoint table in the local database. A different script is installed for each database type. |
| db2cntl.tpl | Template for use with Replicat when creating a control file for the IBM LOADUTIL bulk-load utility. |
| ddl_access.tpl | Template used by the DDLGEN utility to convert source DDL to Microsoft Access DDL. |
| ddl_cleartrace.sql | Script that removes the DDL trace file. (Oracle installations) |
| ddl_db2.tpl | Template used by the DDLGEN utility to convert source DDL to DB2 DDL (Linux, UNIX, Windows). |

Table 9 Other files

| Component | Description |
|---|---|
| ddl_db2_os390.tpl | Template used by the DDLGEN utility to convert source DDL to DB2 DDL (z/OS systems). |
| ddl_ddl2file.sql | Script that saves DDL from the marker table to a file. |
| ddl_disable.sql | Script that disables the Oracle GoldenGate DDL trigger. (Oracle installations) |
| ddl_enable.sql | Script that enables the Oracle GoldenGate DDL trigger. (Oracle installations) |
| ddl_filter.sql | Script that supports filtering of DDL by Oracle GoldenGate. This script runs programmatically; do not run it manually. |
| ddl_informix.tpl | Template used by the DDLGEN utility to convert source DDL to Informix DDL. |
| ddl_mss.tpl | Template used by the DDLGEN utility to convert source DDL to SQL Server DDL. |
| ddl_mysql.tpl | Template used by the DDLGEN utility to convert source DDL to MySQL DDL. |
| ddl_nopurgeRecyclebin.sql | Empty script file for use by Oracle GoldenGate support staff. |
| ddl_nssql.tpl | Template used by the DDLGEN utility to convert source DDL to NonStop SQL DDL. |
| ddl_ora9.sql ddl_ora10.sql ddl_ora11.sql ddl_ora10upCommon.sql | Scripts that run programmatically as part of Oracle GoldenGate DDL support; do not run these scripts. |
| ddl_oracle.tpl | Template used by the DDLGEN utility to convert source DDL to Oracle DDL. |
| ddl_pin.sql | Script that pins DDL tracing, the DDL package, and the DDL trigger for performance improvements. (Oracle installations) |
| ddl_purgeRecyclebin.sql | Script that purges the Oracle recyclebin in support of the DDL replication feature. |
| ddl_remove.sql | Script that removes the DDL extraction trigger and package. (Oracle installations) |
| ddl_session.sql ddl_session1.sql | Supports the installation of the Oracle DDL objects. This script runs programmatically; do not run it manually. |

Table 9 Other files

| Component | Description |
|--|---|
| ddl_setup.sql | Script that installs the Oracle GoldenGate DDL extraction and replication objects. (Oracle installations) |
| ddl_sqlmx.tpl | Template used by the DDLGEN utility to convert Tandem Enscribe DDL to NonStop SQL/MX DDL. |
| ddl_status.sql | Script that verifies whether or not each object created by the Oracle GoldenGate DDL support feature exists and is functioning properly. (Oracle installations) |
| ddl_staymetadata_off.sql ddl_staymetadata_on.sql | Scripts that control whether the Oracle DDL trigger collects metadata. This script runs programmatically; do not run it manually. |
| ddl_sybase.tpl | Template used by the DDLGEN utility to convert source DDL to Sybase DDL. |
| ddl_tandem.tpl | Template used by the DDLGEN utility to convert source DDL to NonStop SQL DDL. |
| ddl_trace_off.sql ddl_trace_on.sql | Scripts that control whether DDL tracing is on or off. |
| ddl_tracelevel.sql | Script that sets the level of tracing for the DDL support feature. (Oracle installations) |
| debug files | Debug text files that may be present if tracing was turned on. |
| demo_<db>_create.sql demo_more_<db>_create.sql demo_<db>_insert.sql demo_more_<db>_insert.sql demo_<db>_lob_create.sql demo_<db>_misc.sql | Scripts that create and populate demonstration tables for use with tutorials and basic testing. |
| .dmp files | Dump files created by Oracle GoldenGate processes for tracing purposes. |

Table 9 Other files

| Component | Description |
|---|---|
| ENCKEYS | User-created file that stores encryption keys. Written in external ASCII format. |
| exitdemo.c | User exit example. |
| freeBSD.txt | License agreement for FreeBSD. |
| ggmessage.dat | Data file that contains error, informational, and warning messages that are returned by the Oracle GoldenGate processes. The version of this file is checked upon process startup and must be identical to that of the process in order for the process to operate. |
| ggserr.log | File that logs processing events, messages, errors, and warnings generated by Oracle GoldenGate. |
| ggsmsg.dll | Windows dynamic link library used by the INSTALL program. |
| GLOBALS | User-created file that stores parameters applying to the Oracle GoldenGate instance as a whole. |
| help.txt | Help file for the GGSCI command interface. |
| icudt38.dll icuin38.dll icuuc38.dll | Windows shared libraries for International Components for Unicode. |
| jagent.bat | Windows batch file for the Java Agent for Oracle GoldenGate Monitor. |
| jagent.log jagentjni.log | Log files for the Oracle GoldenGate Monitor Agent. |
| jagent.sh | UNIX shell script for the Java Agent for Oracle GoldenGate Monitor |
| LGPL.txt | Lesser General Public License statement. Applies to free libraries from the Free Software Foundation. |
| libxml2.dll | Windows dynamic link library containing the XML library for the Oracle GoldenGate XML procedures. |

Table 9 Other files

| Component | Description |
|---------------------------------------|---|
| libxml2.txt | License agreement for libxml2.dll. |
| marker.hist | File created by Replicat if markers were passed from a NonStop source system. |
| marker_remove.sql | Script that removes the DDL marker table. (Oracle installations) |
| marker_setup.sql | Script that installs the Oracle GoldenGate DDL marker table. (Oracle installations) |
| marker_status.sql | Script that confirms successful installation of the DDL marker table. (Oracle installations) |
| notices.txt | Third-party software license file. |
| params.sql | Script that contains configurable parameters for DDL support. (Oracle installations) |
| pthread-win32.txt | License agreement for pthread-VC.dll. |
| pthread-VC.dll | POSIX threads library for Microsoft Windows. |
| prvtclkm.plb | Supports the replication of Oracle encrypted data. |
| pw_agent_util.bat pw_agent_util.sh | Script files that support the Oracle GoldenGate Monitor Agent. |
| role_setup.sql | Script that creates the database role necessary for Oracle GoldenGate DDL support. (Oracle installations) |
| sqlldr.tpl | Template for use with Replicat when creating a control file for the Oracle SQL*Loader bulk-load utility. |
| start.prm stop.prm | z/OS paramlib members to start and stop the Manager process. |
| startmgr stopmgr | z/OS Unix System Services scripts to start the Manager process from GGSCI. |
| startmgrcom stopmgrcom | z/OS system input command for the Manager process. |
| tcperrs | File containing user-defined instructions for responding to TCP/IP errors. |
| usrdecs.h | Include file for user exit API. |

Table 9 Other files

| Component | Description |
|------------------|---|
| xerces-c_2_8.dll | Apache XML parser library. |
| zlib.txt | License agreement for zlib compression library. |

Oracle GoldenGate checkpoint table

When database checkpoints are being used, Oracle GoldenGate creates a checkpoint table with a user-defined name in the database upon execution of the ADD CHECKPOINTTABLE command, or a user can create the table by using the chkpt_<db>_create.sql script, where <db> is the type of database.

Do not change the names or attributes of the columns in this table. You can change table storage attributes as needed.

Table 10 Checkpoint table definitions

| Column | Description |
|--------------------------|---|
| GROUP_NAME (primary key) | The name of a Replicat group using this table for checkpoints. There can be multiple Replicat groups using the same table. |
| GROUP_KEY (primary key) | A unique identifier that, together with GROUPNAME, uniquely identifies a checkpoint regardless of how many Replicat groups are writing to the same table. |
| SEQNO | The sequence number of the checkpoint file. |
| RBA | The relative byte address of the checkpoint in the file. |
| AUDIT_TS | The timestamp of the checkpoint position in the checkpoint file. |
| CREATE_TS | The date and time when the checkpoint table was created. |
| LAST_UPDATE_TS | The date and time when the checkpoint table was last updated. |
| CURRENT_DIR | The current Oracle GoldenGate home directory or folder. |

Index



Symbols

- \$LD_LIBRARY_PATH variable** 15
- \$PATH variable** 14, 15

A

- ADD REPLICAT command** 7
- ADDEVENTS Windows service option** 17
- ADDSERVICE Windows service option** 18
- AUTOSTART Windows service option** 18

C

- cascade deletes, disabling** 21
- cascade updates, disabling** 21
- case, supported** 10
- category.dll** 17
- Change Data Capture (CDC)** 28
- character sets, configuring** 22
- characters**
 - supported in object names 10
- cluster**
 - installing on 15, 16, 18
- cluster, installing on** 3
- columns**
 - number and size supported 8
 - supported data types 4
- commit mode, choosing** 28
- components, Oracle GoldenGate** 44
- connections, to database** 35
- constraints, integrity** 21
- CREATE SUBDIRS command** 16

D

- data types, supported** 4

database

- character sets 22
- preparing for processing 21
- requirements 4
- versions supported 2

DBOPTIONS parameter

DDL

- activating capture of 23
- supported objects and operations 8

deletes, cascaded

disk requirements

E

- environment variables, setting** 14
- Extract, VAM-sort** 28

F

- files, installed by Oracle GoldenGate** 44
- firewall, configuring** 3

G

- ggmessage.dat file** 51
- GGSMGR default Manager name** 16
- ggsmsg.dll** 17
- GLOBALS file** 17
- grants, security** 4

I

- identity data type** 8
- initial synchronization** 36
- initialization file, configuring** 24
- installation**
 - requirements 2
- installing on**
 - Linux and UNIX 15
 - Windows 16



K

key

- assigning 21
- name, supported characters 10

KEYCOLS option, TABLE or MAP 22

L

LIBPATH variable 15

libraries, Visual C++ 3

Linux, installing on 15

M

Manager

- as Windows service 17
- multiple on same system 16
- name, customizing 16

MANUALSTART Windows service option 18

maximum performance mode 28

maximum protection mode 28

MGRSERVNAME parameter 17

Microsoft Visual C ++ 2005 SP1 Redistributable Package 3

mode, commit 28

N

name

- non-supported characters in 12
- supported characters in 10

names, supported 10

network configuration 3

NOCATALOGCONNECT option, DBOPTIONS 35

NODBCHECKPOINT option, ADD REPLICAT 7

O

objects, supported 8

operating systems supported 2

operations, supported 8

Oracle GoldenGate

- installed programs and files 44
- installing 13
- uninstalling 41

Oracle, versions supported 2

P

PASSWORD Windows service option 18

platforms, supported 2

ports, required by Oracle GoldenGate 3

pre-installation instructions 1

privileges

- operating system 3

R

Relay Services Gateway (RSG) vprocs 3

replication group

- creating 4
- security grants 4

replication server

- about 1
- requirements 2

rows, size supported 8

S

security grants 4

server, replication

- about 1
- requirements 2

SHLIB_PATH variable 15

spaces

- in folder names 16
- in object and column names 12

SQL Server requirements 4

subdirectories, creating 16

system requirements 1

T

tables

- initial synchronization 36
- preparing for processing 21
- supported kinds 8

tam.ini file 24

TCP/IP, configuring 3

Teradata Access Module (TAM) 2

trail, VAM 28

triggers, disabling on target 21

two-phase commit 28

U

uninstalling Oracle GoldenGate 41

UNIX, installing on 15

USER Windows service option 18

V

VAM trail 28

VAMSERV program 45

VAM-sort Extract 28

versions of Oracle supported 2

virtual machine, support for 4

Visual C ++ 2005 SP1 Redistributable Package 3

W

Windows, installing on 16