Oracle® GoldenGate
Sybase Installation and Setup Guide
11*g* Release 2 Patch Set 1 (11.2.1.0.1)
E27294-01

April 2012

ORACLE®

Oracle GoldenGate SybaseInstallation and Setup Guide 11g Release 2 Patch Set 1 (11.2.1.0.1)

E27294-01

Contents

. . . .

. . . .

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 Sybase

With Oracle GoldenGate for Sybase database, you can replicate data to and from supported Sybase versions, or between a Sybase database and a database of another type. Oracle GoldenGate for Sybase supports data filtering, mapping, and transformation unless noted otherwise in this 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

Memory requirements

The amount of memory that is required for Oracle GoldenGate depends on the number of concurrent processes that will be running. At minimum on the source system, there is a primary Extract process that captures source data and a secondary Extract data-pump process that transfers data across the network. At minimum on the target system is at least one Replicat process that applies the replicated data to the target database. In some cases, these processes might all operate on the same system, depending on the required configuration.

It is possible that you will need to use additional, parallel processes to improve throughput if your environment generates a large volume of transactional data that must be replicated. Oracle GoldenGate supports up to 5,000 concurrent Extract and Replicat processes per instance of Oracle GoldenGate. Each Extract and Replicat process needs approximately 25-55 MB of memory, or more depending on the size of the transactions and the number of concurrent transactions.

The actual amount of physical memory that is used by any Oracle GoldenGate process is controlled by the operating system, not the Oracle GoldenGate program. The Oracle GoldenGate cache manager takes advantage of the memory management functions of the operating system to ensure that Oracle GoldenGate processes work in a sustained and

. . . .

efficient manner. For more information about evaluating Oracle GoldenGate memory requirements, see the CACHEMGR parameter in the Oracle GoldenGate *Windows and UNIX Reference Guide*.

Disk requirements

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 as the Oracle user on a shared file system that is available to all cluster nodes. See "Preparing to install Oracle GoldenGate within a cluster" on page 28 for more information.
- 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. See the guidelines for sizing trails in the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

Temporary disk requirements

By default, Oracle GoldenGate maintains data that it swaps to disk in the dirtmp sub-directory of the Oracle GoldenGate installation directory. The cache manager assumes that all of the free space on the file system is available. This directory can fill up quickly if there is a large transaction volume with large transaction sizes. To prevent I/O contention and possible disk-related Extract failures, dedicate a disk to this directory. You can assign a name to this directory with the CACHEDIRECTORY option of the CACHEMGR parameter.

Network

- Configure the system to use TCP/IP services, including DNS. Oracle GoldenGate supports IPv4 and IPv6 and can operate in a system that supports one or both of these protocols.
- 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

- To install on Windows, the person who installs Oracle GoldenGate must log in as Administrator.
- To install on UNIX, the person who installs Oracle GoldenGate must have read and write privileges on the Oracle GoldenGate installation directory.
- The Oracle GoldenGate Extract, Replicat, and Manager processes must operate as an operating system user that has privileges to read, write, and delete files and subdirectories in the Oracle GoldenGate directory. In addition, the Manager process requires privileges to control the other Oracle GoldenGate processes.
- The Extract process must operate as an operating system user that has read access to the transaction log files, both online and archived. If you install the Manager process as a Windows service during the installation steps in this documentation, you must install as Administrator for the correct permissions to be assigned. If you cannot install Manager as a service, assign read access to the Extract process manually, and then always run Manager and Extract as Administrator.
- Dedicate the Extract, Replicat, and Manager operating system users to Oracle GoldenGate. Sensitive information might be available to anyone who runs an Oracle GoldenGate process.

Console

The operating system and the command console must have the same character sets. Mismatches occur on Microsoft Windows systems, where the operating system is set to one character set, but the DOS command prompt uses a different, older DOS character set. Oracle GoldenGate uses the character set of the operating system to send information to GGSCI command output; therefore a non-matching console character set causes characters not to display correctly. You can set the character set of the console before opening a GGSCI session by using the following DOS command:

```
chcp <OS character set>
```

If the characters do not display correctly after setting the code page, try changing the console font to Lucida Console, which has an extended character set.

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

- Set the DSQUERY variable to the server that contains the database that Oracle GoldenGate will be using.
- The Extract process uses an interface that makes calls directly to the Sybase Replication API.
- Extract uses the Sybase LTM to read the Sybase transaction log. Extract cannot run if Sybase RepServer is running. Only one process at a time can reserve a context that allows it to read the transaction log on the same database.
- The Extract process must be permitted to manage the secondary log truncation point.
- The source replication server must be an active database. Oracle GoldenGate cannot capture from a database that is in warm standby mode.
- To correctly process UPDATE operations that affect a decimal column,

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.
- The Extract process requires permission to access the source database. Do one of the following:
  - ○ Grant System Administrator privileges.
  - ○ Assign a user name with the sa_role and replication_role. The command to grant replication role is:

    ```
    sp_role 'grant', replication_role, <Extract user>
    ```

- The Replicat process requires connect and DML privileges on the target database.

Supported Sybase data types

Integers

- BIT
- DECIMAL
- INT (signed)
- TINYINT (signed)
- NUMERIC
- SMALLINT (signed)

Limitations of support

- NUMERIC and DECIMAL (fixed-point) are supported with no integrity loss when moving data to a target column of the same data type without involving calculations or transformation. When calculations or transformation must be performed, Oracle GoldenGate supports a maximum value of a signed long integer (32-bits).

- BIT is supported for automatic mapping between Sybase databases. To move BIT data between Sybase and another database type, Oracle GoldenGate treats BIT data as binary. In this case, the following are required:
  - ○ The BIT column must be mapped to the corresponding source or target column with a COLMAP clause in a TABLE or MAP statement.
  - ○ The BIT data must be converted by means of the NUMBIN Oracle GoldenGate column-conversion function.

- See also Non-Supported Sybase Data Types.

Floating-point numbers

- DOUBLE
- FLOAT
- REAL

Limitations of support

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.

Character data

- CHAR
- NCHAR
- NVARCHAR
- VARCHAR
- UNICHAR
- UNIVARCHAR

Limitations of support

These data types are supported to the maximum length supported by the database, this being the maximum page size.

Dates and timestamps

- BIGDATETIME
- BIGTIME
- DATE
- DATETIME
- SMALLDATETIME
- TIME

Limitations of support

Oracle GoldenGate does not support negative dates.

Large objects

- BINARY
- IMAGE
- TEXT
- VARBINARY

Limitations of support

- TEXT and IMAGE are supported up to 2 GB in length.

- Large objects that are replicated from other databases (such as Oracle BLOB and CLOB) can be mapped to Sybase CHAR, VARCHAR, BINARY, and VARBINARY columns. To prevent Replicat from abending if the replicated large object is bigger than the size of the target column, use the DBOPTIONS parameter with the ALLOWLOBDATATRUNCATE option in the Replicat parameter file. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

- To move data to a Sybase target from a source database that permits empty LOB columns, use the DBOPTIONS parameter with the EMPTYLOBSTRING option in the Replicat parameter file. This parameter accepts a string value and prevents Replicat from setting the target column to NULL, which is not permitted by Sybase. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

- When a source table contains multiple identical rows, it can cause LOB inconsistencies in the target table. This occurs when the source table lacks a primary key or other unique row identifier. The rows are inserted by Replicat on the target, but if the LOB data is updated in a subsequent source operation, it will only be replicated to the first row that was inserted on the target.

- When the size of a large object exceeds 4K, Oracle GoldenGate stores the data in segments within the Oracle GoldenGate trail. The first 4K is stored in the base segment, and the rest is stored in a series of 2K segments. Oracle GoldenGate does not support the filtering, column mapping, or manipulation of large objects of this size. Full Oracle GoldenGate functionality can be used for objects that are 4K or smaller.

Money types

- MONEY
- SMALLMONEY

Limitations of support

Money data types are supported with no integrity loss when moving data to a target column of the same data type without involving calculations or transformation. When calculations or transformation must be performed, Oracle GoldenGate supports a maximum value of a signed long integer (32-bits).

IDENTITY type

The IDENTITY data type is supported for replication in one direction only, but not for a bi-directional configuration.

User-defined types

User-defined types are fully supported.

Non-supported Sybase datatypes

TIMESTAMP

Timestamp columns must be excluded from Oracle GoldenGate because they are populated automatically by the database, which generates errors on the target if Replicat attempts to apply a replicated timestamp value. To exclude timestamp columns from being captured by Oracle GoldenGate, use the COLSEXCEPT option of the TABLE parameter. Because the system generates the timestamps, the source and target values will be different.

Other non-supported Sybase data types

- BIGINT
- UNITEXT
- Unsigned INT
- Unsigned TINYINT
- Unsigned SMALLINT

Supported operations and objects for Sybase

- Oracle GoldenGate supports the extraction and replication of insert, update, and delete operations on Sybase tables that contain rows of up to 512 KB in length.
- Oracle GoldenGate supports the maximum number of columns and the maximum column size per table that is supported by the database.
- Oracle GoldenGate supports deferred inserts, deferred indirect inserts, deferred updates, and deferred deletes. It is possible that the use of deferred updates could cause primary key constraint violations for the affected SQL on the target. If these errors occur, use the Replicat parameter HANDLECOLLISIONS.
- Oracle GoldenGate supports TRUNCATE TABLE if the names of the affected tables are unique across all schemas. If the table names are not unique across all schemas, use the IGNORETRUNCATES parameter for those tables to prevent Replicat from abending.
- Oracle GoldenGate supports data that is encrypted with a system-encrypted password.
- Oracle GoldenGate supports array fetching during initial loads, as controlled by the FETCHBATCHSIZE parameter.

Limitations on computed columns

- Oracle GoldenGate fully supports persisted computed columns. The change values are present in the transaction log and can be captured to the trail.
- Oracle GoldenGate supports tables with non-persisted computed columns, but does not capture change data for these columns, because the database does not write it to the transaction log. To replicate data for non-persisted computed columns, you can use the FETCHCOLS or FETCHMODCOLS option of the TABLE parameter to fetch the column data from

the table. Keep in mind that there can be discrepancies caused by differences in data values between when the column was changed in the database and when Extract fetches the data for the transaction record that is being processed.

● Replicat does not apply DML to any computed column, even if the data for that column is in the trail, because the database does not permit DML on that type of column. Data from a source *persisted* computed column, or from a fetched non-persisted column, can be applied to a target column that is not a computed column.

● In an initial load, all of the data is selected directly from the source tables, not the transaction log. Therefore, in an initial load, data values for all columns, including non-persisted computed columns, gets written to the trail or sent to the target, depending on the method that is being used. As when applying change data, however, Replicat does not apply initial load data to computed columns, because the database does not permit DML on that type of column.

● Oracle GoldenGate will not use a persisted computed column that is defined as a key column, an index column, or that is part of a KEYCOLS clause in a TABLE or MAP statement. If a unique key or index includes a computed column and Oracle GoldenGate must use that key, the computed column will be ignored. Additionally, if a unique key or index contains a computed column and is the only unique identifier on the table, Oracle GoldenGate will use all of the columns except the computed column as an identifier to find the target row. Thus, the presence of a computed column in a key or index affects data integrity if the remaining columns do not enforce uniqueness. Note that Sybase does not support non-persisted computed columns as part of a key, and neither does Oracle GoldenGate.

Non-supported operations and objects for Sybase

● Data that is encrypted with a user-defined password.

● Extraction or replication of DDL (data definition language) operations.

● The BATCHSQL feature of Oracle GoldenGate.

● Multi-Extract configuration. Only one Extract can reserve a context to read the Sybase transaction logs.

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 on UNIX

Preparing to install Oracle GoldenGate within a cluster

Installing Oracle GoldenGate on Linux and UNIX

Installing Oracle GoldenGate on Windows

Integrating Oracle GoldenGate into a cluster

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 done *before you run GGSCI or any other Oracle GoldenGate process*. 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.

Preparing to install Oracle GoldenGate within a cluster

This topic covers the installation requirements that apply when Oracle GoldenGate will be installed in a cluster environment. Oracle GoldenGate can be used with any cluster-management solution that is Oracle-certified. You will need to install at least some Oracle GoldenGate objects on shared storage. Select cluster-aware shared storage that is independent of, but available to, all nodes of the cluster.

Deciding where to install Oracle GoldenGate binaries and files in the cluster

The best practice is the install Oracle GoldenGate entirely on shared storage. This allows you to start the Oracle GoldenGate processes from any of the nodes without having to make changes to the parameter files. If the active node fails, the processes can be started quickly on another node, using the processing checkpoints that are preserved in the installation directory.

If you decide to install the Oracle GoldenGate binaries and files on each node, rather than on shared storage, the following must be true:

● The Oracle GoldenGate installation must have the same location path on every node

● At minimum, install the following directories on the shared storage to support Oracle GoldenGate recovery requirements. On UNIX or Linux, you can create symbolic links to them from the installation directory on each node.

  ❍ br
  ❍ dirchk
  ❍ dirdat
  ❍ dirtmp

These directories are among those created when you issue CREATE SUBDIRS during installation.

● The parameter files in the dirprm directory, if not placed on the shared drive, must be identical on all nodes. To resolve environment settings that must be different from one node to the other, you can set environment settings so they are inherited from the local Manager process or reference a node-specific Oracle GoldenGate macro file. Because this scenario can be difficult to enforce, the inherent concerns can be avoided by storing the parameter files on the shared drive.

See also "Integrating Oracle GoldenGate into a cluster" on page 15 after you install Oracle GoldenGate.

Installing Oracle GoldenGate on Linux and UNIX

Follow these steps to install Oracle GoldenGate for Oracle on a Linux or UNIX system or in the appropriate location in a cluster. See "Preparing to install Oracle GoldenGate within a cluster" on page 11 for more information.

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

Follow these steps to install Oracle GoldenGate for Oracle on a Windows system or in the appropriate location in a cluster. See "Preparing to install Oracle GoldenGate within a cluster" on page 11 for more information.

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 GGSMGR.

● There will be multiple Manager processes running as Windows services on this system. 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 1   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 ggsmsg.dll
ADDSERVICE	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] 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.
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>	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. By default, the Manager service is installed to use the Local System account.
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.

Integrating Oracle GoldenGate into a cluster

If you installed Oracle GoldenGate in a cluster, take the following steps to integrate Oracle GoldenGate within the cluster solution.

General requirements in a cluster

1. Register the Oracle GoldenGate Manager process (and only Manager) as a cluster-managed resource as you would any other application. Manager must be the only Oracle GoldenGate process that the cluster-management software starts and stops, because it is the parent process that manages all other processes.

2. If the cluster uses a virtual IP address, you may need to obtain an available fixed IP address for the Manager process. The VIP must be an available IP address on the public subnet and cannot be determined through DHCP. In the parameter files of the Extract data pumps, specify the VIP of the remote Manager as the input value of the RMTHOST parameter. Other Oracle GoldenGate products that access Manager also should use the VIP.

3. Make certain that all nodes in the cluster have synchronized system clocks. The clocks must be synchronized with the clock on the system where Extract is executed. Oracle GoldenGate compares the time of the local system to the commit timestamps to make critical decisions. For information about synchronizing system clocks, consult www.ntp.org or your systems administrator. See also the IOLATENCY option of the THREADOPTIONS parameter in the ***Oracle GoldenGate Windows and UNIX Reference Guide***.

4. When you configure Manager, add the AUTOSTART and AUTORESTART parameters so that Manager starts the replication processes automatically (see "Creating the Oracle GoldenGate instance" on page 46). You can, when needed, control Extract, Replicat, and other Oracle GoldenGate processes from within the Oracle GoldenGate user interfaces.

5. Mount the shared drive on one node only. This prevents processes from being started on another node. Use the same mount point on all nodes.

6. Configure Oracle GoldenGate as directed in this documentation.

Adding Oracle GoldenGate as a Windows cluster resource

When installing Oracle GoldenGate in a Windows cluster, follow these instructions to establish Oracle GoldenGate as a cluster resource and configure the Manager service correctly on all nodes.

● In the cluster administrator, add the Manager process to the group that contains the database to which Oracle GoldenGate will connect.

● Make sure all nodes on which Oracle GoldenGate will run are selected as possible owners of the resource.

● Make certain the Manager Windows service has the following dependencies (configurable from the Services control panel):
  ❍ 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

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 target Sybase 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.

To configure Replicat to disable target triggers at the start of its database session:

1. Assign the Replicat user the replication role.

2. Add the following parameter statement to the root level of the Replicat parameter file.

   ```
   SQLEXEC "set triggers off"
   ```

Assigning row identifiers

Oracle GoldenGate requires some form of unique row identifier on 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 with no computed columns, no function-based columns, and no nullable columns.

3. First unique key alphanumerically with no computed or function-based columns, but can include nullable columns.

4. If none of the preceding key types exist (even though there might be other types of keys defined on the table) GoldenGate constructs a pseudo key of all columns that the database allows to be used in a unique key, excluding computed columns, function-based columns, and any columns that are explicitly excluded from the 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.

Oracle GoldenGate does not support a primary or unique key in which a column name contains an embedded comma, such as the following:

```
CREATE TABLE Tab ... PRIMARY KEY ("col1, Id", col2)
```

In this case "col1, ID" is the first primary key column and col2 is the second primary key column.

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. Oracle GoldenGate uses the specified key in place of any existing primary or unique key that it finds. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

Limiting row changes in tables that do not have a key

If a target table has no primary key or unique key, duplicate rows can exist. It is possible for Oracle GoldenGate to update or delete too many rows in the target table, causing the source and target data to go out of synchronization without error messages to alert you. To limit the number of rows that are updated, use the DBOPTIONS parameter with the LIMITROWS option in the Replicat parameter file. LIMITROWS can increase the performance of Oracle GoldenGate on the target system because only one row is processed.

Replicating encrypted data

Oracle GoldenGate supports columns that are encrypted with a system-encrypted password, but not columns that are encrypted with a user-defined password. Check the tables from which you want to capture data against the following Oracle GoldenGate limitations:

● The table that contains the encrypted columns must have a primary or unique key.

● Columns that use encryption cannot be part of the primary key.

To enable processing of encrypted data

Encrypted columns are encrypted in the data files and in the log, so Extract must be configured to fetch the clear-text values from the database. To trigger this fetch, use the FETCHCOLS and FETCHMODCOLS[EXCEPT] options of the Extract TABLE parameter. FETCHCOLS

forces a fetch of values that are not in the log, and FETCHMODCOLS or FETCHMODCOLS[EXCEPT] forces a fetch of values that are in the logs. Used together, these parameters ensure that the encrypted columns are always fetched from the database.

The following is an example of how to configure Extract to support the encryption. In this example, the encrypted column is cardnum.

```
TABLE ab.payments, FETCHCOLS (cardnum), FETCHMODCOLS (cardnum);
```

Preparing the transaction logs

To capture DML operations, Oracle GoldenGate reads the online logs. To ensure the continuity and integrity of Oracle GoldenGate processing, configure the logs as follows.

Enabling transaction logging

Use the ADD TRANDATA command to mark each source table for replication. This command uses the Sybase sp_setreptable and sp_setrepcol system procedures. ADD TRANDATA is the recommended way to mark the tables, instead of using those procedures through the database interface, but the owner or the system administrator can use them if needed. For more information, see the Sybase documentation.

To mark tables for replication with ADD TRANDATA

1. On the source system, run GGSCI from the Oracle GoldenGate directory.

2. Log into the database from GGSCI.

```
DBLOGIN SOURCEDB <database> USERID <user>
```

Where:

❍ <database> is the name of the database.

❍ <user> is the database owner or the system administrator. You will be prompted for the password. This command has encryption options for the password. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

3. Issue ADD TRANDATA for each table to be marked.

```
ADD TRANDATA <table> [, LOBSNEVER | LOBSALWAYS | LOBSIFCHANGED]
```

Where:

❍ <table> is the fully qualified name of the table. You can use a wildcard character to specify multiple table names but not the owner.

❍ LOBSNEVER | LOBSALWAYS | LOBSALWAYSNOINDEX | LOBSIFCHANGED control whether LOB data is never propagated, only propagated if changed (the default), or always propagated. The ADD TRANDATA command will overwrite the LOB replication setting that is currently set for the table.

> **NOTE** Some ADD TRANDATA options enable the ALWAYS_REPLICATE option of sp_setrepcol. If a LOB column contains a NULL value, and then another column in the table gets updated (but not the LOB), that LOB will not be captured even though ALWAYS_REPLICATE is enabled.

. . . .

Oracle GoldenGate *Sybase Installation and Setup Guide*   19

Sizing and retaining the logs

Retain enough log data on the source system so that Extract can start again from its checkpoints after you stop it or there is an unplanned outage. Extract must have access to the log that contains the start of the oldest uncommitted unit of work, and all logs thereafter.

If data that Extract needs during processing was not retained, either in online or backup logs, one of the following corrective actions might be required:

● Alter Extract to capture from a later point in time for which log data is available (and accept possible data loss on the target).

● Resynchronize the source and target tables, and then start the Oracle GoldenGate environment over again.

To determine where the Extract checkpoints are, use the INFO EXTRACT command. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

Purging log archives

Make certain not to use backup or archive options that cause old archive files to be overwritten by new backups on the source system. New backups should be separate files with different names from older ones. This ensures that if Extract looks for a particular log, it will still exist, and it also ensures that the data is available in case it is needed for a support case.

Initializing the secondary truncation point

Establish a secondary log truncation point on the source system prior to running the Oracle GoldenGate Extract process. Extract uses the secondary truncation point to identify data that remains to be processed.

To initialize the secondary truncation point

1. Log on to the database as a user with sa_role privileges.

2. Issue the following Sybase command:

```
dbcc settrunc( 'ltm', valid )
```

By default, Extract will manage the secondary truncation point once it is established. Do not permit Extract to be stopped any longer than necessary; otherwise the log could eventually fill up and the database will halt. The only way to resolve this problem is to disable the secondary truncation point and manage it outside of Oracle GoldenGate, and then purge the transaction log. Data not yet processed by Extract will be lost, and you will have to resynchronize the source and target data.

To control how the secondary truncation point is managed, use the TRANLOGOPTIONS parameter. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*

CHAPTER 4

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

On all systems:

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 delete files and directories from the operating system.

3. Change directories to the Oracle GoldenGate installation directory.

4. Run GGSCI.

5. Stop all Oracle GoldenGate processes.

6. Stop the Manager process.

On a source system:

7. Log into the database with the DBLOGIN command, and then run the DELETE TRANDATA command to disable replication.

8. If Extract was managing the secondary truncation point, enable the database to assume management of it.

   ```
   dbcc settrunc( 'ltm', ignore )
   ```

On any system where a Replicat checkpoint table is being used:

9. Log into the database with the DBLOGIN command, and then remove the Replicat checkpoint table by running the DELETE CHECKPOINTTABLE command.

On all systems:

10. Make certain all processes are stopped (including GGSCI).

11. Remove the Oracle GoldenGate files by removing the installation directory.

Removing Oracle GoldenGate from Windows Cluster

1. Working from the node in the cluster that owns the cluster group that contains the Manager resource, run GGSCI and then stop any Extract and Replicat processes that are still running.

2. Use the Cluster Administrator tool to take the Manager resource offline.

3. Right click the resource and select **Delete** to remove it.

4. Click **Start** > **Run**, and type cmd in the **Run** dialog box to open the command console.

5. Change directories to the Oracle GoldenGate installation directory.

6. 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.

7. Delete the CATEGORY.DLL and GGSMSG.DLL files from the Windows SYSTEM32 folder.

8. Move the cluster group to the next node in the cluster, and repeat from step 4.

9. Follow the instructions in "Uninstalling Oracle GoldenGate from Windows (non-cluster)".

Uninstalling Oracle GoldenGate from Windows (non-cluster)

On all systems:

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. From the Oracle GoldenGate installation folder, run GGSCI.

3. Stop all Oracle GoldenGate processes.

4. Stop the Manager program or service.

    > **NOTE**   Skip step 5 through step 8 if you already performed them when removing Oracle GoldenGate from a Windows cluster.

5. Click **Start** > **Run**, and type cmd in the **Run** dialog box to open the command console.

6. Change directories to the Oracle GoldenGate installation directory.

7. 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.

8. Delete the CATEGORY.DLL and GGSMSG.DLL files from the Windows SYSTEM32 folder.

On a source system:

9. Log into the database with the DBLOGIN command, and then run the DELETE TRANDATA command to disable replication.

10. If Extract was managing the secondary truncation point, enable the database to assume management of it.

    ```
    dbcc settrunc( 'ltm', ignore )
    ```

On any system where a Replicat checkpoint table is being used:

11. Log into the database with the DBLOGIN command, and then remove the Replicat checkpoint table by running the DELETE CHECKPOINTTABLE command.

On all systems:

12. Make certain all processes are stopped (including GGSCI) and then remove the Oracle GoldenGate files by removing the installation directory.

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 2   Programs and utilities

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.
emsclnt	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.
ggmxinstall	Oracle GoldenGate installation script for the SQL/MX database.
ggsci	User interface to Oracle GoldenGate for issuing commands and managing parameter files.

· ·

Table 2   Programs and utilities (continued)

Program	Description
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 3   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.
dirdb	Contains the datastore that is used to persist information that is gathered from an Oracle GoldenGate instance for use by the Oracle GoldenGate Monitor application or within Oracle Enterprise Manager.

. . . .

Table 3   Subdirectories (continued)

Directory	Description
dirchk	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. 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. Do not edit these files. Examples: ext1.cpe rep1.cpr
dirdat	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. 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). Do not edit these files. Examples: rt000001 finance
dirdef	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. 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. Example: defs.dat
dirjar	Contains the Java executable files that support Oracle GoldenGate Monitor.
dirout	This directory is not used any more.

Table 3   Subdirectories (continued)

Directory	Description
dirpcs	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).
	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.
	Do not edit these files.
	Examples:
	mgr.pcm
	ext.pce
dirprm	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.
	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.
	Examples:
	defgen.prm
	finance.prm
dirrec	Not used by Oracle GoldenGate.
dirrpt	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.
	File name format is \<group name\>\<sequence number\>.rpt where \<sequence number\> is a sequential number appended to aged files.
	Do not edit these files.
	Examples:
	fin2.rpt
	mgr4.rpt
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.
dirtmp	The default location for storing transaction data when the size exceeds the memory size that is allocated for the cache manager. Do not edit these files.
dirwlt	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 4   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).
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.

Table 4   Other files (continued)

Component	Description
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.
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.

Table 4   Other files (continued)

Component	Description
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.
ENCKEYS	User-created file that stores encryption keys. Written in external ASCII format.
exitdemo.c	User exit example.
exitdemo_utf16.c	User exit example that demonstrates how to use UTF16 encoded data in the callback structures for information exchanged between the user exit and the process.
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.

Table 4   Other files (continued)

Component	Description
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.
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)

Table 4   Other files (continued)

Component	Description
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.
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 5   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.

Table 5   Checkpoint table definitions

Column	Description
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

● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

● ●