

**Oracle® GoldenGate**  
Oracle Installation and Setup Guide  
11g Release 1 Patch Set 1 (11.1.1.1)  
**E21406-02**

May 2011

**ORACLE®**

**E21406-02**

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



<b>Chapter 1</b>	<b>System requirements and preinstallation instructions</b>	1
	Overview of Oracle GoldenGate for Oracle	1
	Supported Platforms	1
	Operating system requirements	1
	Memory requirements	1
	Disk requirements	2
	Storage for Oracle GoldenGate trails	2
	Temporary disk requirements	3
	Oracle RAC requirements	3
	Network	4
	Operating system privileges	4
	Itanium requirements	4
	Other programs	5
	Database configuration	5
	Database client	5
	Database user for Oracle GoldenGate processes	5
	Database privileges	6
	Supported Oracle data types	8
	Numeric data types	8
	Character data types	8
	Multi-byte character types	9
	Binary data types	9
	Date and timestamp data types	9
	Large object data types	10
	XML data types	11
	User defined types	12
	Other supported data types	13
	Non-supported Oracle data types	13
	Supported objects and operations for Oracle DML	13
	Tables, views, and materialized views	13
	Sequences	16
	Non-supported objects and operations for Oracle DML	17
	Supported objects and operations for Oracle DDL	17
	Non-supported objects and operations for Oracle DDL	18



	Oracle-reserved schemas .....	18
	Oracle recycle bin .....	18
	Other non-supported DDL .....	18
	Supported and non-supported object names and case.....	19
	Object names and owners.....	19
	Case sensitivity.....	19
	Supported characters.....	19
	Non-supported characters .....	21
<b>Chapter 2</b>	<b>Installing Oracle GoldenGate .....</b>	<b>22</b>
	Installation overview .....	22
	Downloading Oracle GoldenGate.....	22
	Setting ORACLE_HOME and ORACLE_SID .....	23
	Setting library paths for dynamic builds on UNIX .....	24
	Installing Oracle GoldenGate on Linux and UNIX.....	26
	Installing Oracle GoldenGate into a Linux or UNIX cluster .....	26
	Installing the Oracle GoldenGate files.....	26
	Installing Oracle GoldenGate on Windows and Windows Cluster .....	27
	Installing Oracle GoldenGate into a Windows Cluster.....	27
	Installing the Oracle GoldenGate files.....	27
	Specifying a custom Manager name .....	27
	Installing Manager as a Windows service .....	28
	Adding Oracle GoldenGate as a Windows cluster resource .....	29
	Installing support for Oracle sequences .....	30
	Configuring Manager and other processes.....	31
<b>Chapter 3</b>	<b>Installing Oracle GoldenGate DDL support for an Oracle database.....</b>	<b>32</b>
	Overview of the DDL objects.....	32
	Installing the DDL objects.....	33
<b>Chapter 4</b>	<b>Preparing the database for Oracle GoldenGate.....</b>	<b>36</b>
	Preparing constraints for Oracle GoldenGate.....	36
	Disabling triggers and cascade constraints .....	36
	Deferring constraint checking.....	36
	Assigning row identifiers .....	37
	Configuring the database to log key values .....	38
	Limiting row changes in tables that do not have a key.....	40
	Configuring character sets .....	40
	Adjusting cursors .....	41
	Setting fetch options .....	41
	Replicating TDE-encrypted data .....	43

Required database patches.....	43
Overview of TDE support.....	43
Requirements for replicating TDE.....	43
Recommendations for replicating TDE.....	44
Configuring TDE support.....	44
Changing the Oracle shared secret.....	45
Ensuring correct handling of Oracle Spatial objects.....	46
Mapping the georaster tables.....	46
Sizing the XML memory buffer.....	46
Handling triggers on the georaster tables.....	46
Replicating TIMESTAMP with TIME_ZONE.....	48
Controlling Replicat COMMIT options on an Oracle target.....	49
Supporting delivery to Oracle Exadata with EHCC compressed data.....	49
Managing LOB caching on a target Oracle database.....	50
Additional requirements for Oracle RAC.....	50
General requirements.....	50
Special procedures on RAC.....	51
Additional requirements for ASM.....	52
Ensuring ASM connectivity.....	52
Optimizing the ASM connection.....	52
<b>Chapter 5 Preparing DBFS for active-active propagation with Oracle GoldenGate.....</b>	<b>53</b>
Supported operations and prerequisites.....	53
Applying the required patch.....	53
Examples used in these procedures.....	53
Partitioning the DBFS sequence numbers.....	54
Configuring the DBFS filesystem.....	55
Mapping local and remote peers correctly.....	56
<b>Chapter 6 Configuring the Oracle redo logs.....</b>	<b>58</b>
Ensuring data availability.....	58
Log retention requirements per Extract recovery mode.....	58
Log retention options.....	58
Determining how much data to retain.....	59
Purging log archives.....	60
Specifying the archive location.....	60
Mounting logs that are stored on other platforms.....	60
Configuring Oracle GoldenGate to read only the archived logs.....	60
Limitations and requirements of ALO mode.....	60
Configuring Extract for ALO mode.....	61

	Setting redo parallelism for Oracle 9i sources .....	62
	Avoiding log-read bottlenecks .....	62
<b>Chapter 7</b>	<b>Managing the Oracle DDL replication environment .....</b>	<b>63</b>
	Enabling and disabling the DDL trigger .....	63
	Maintaining the DDL marker table .....	63
	Deleting the DDL marker table .....	63
	Maintaining the DDL history table .....	64
	Deleting the DDL history table .....	64
	Purging the DDL trace file .....	65
	Applying database patches and upgrades when DDL support is enabled .....	65
	Applying Oracle GoldenGate patches and upgrades when DDL support is enabled ..	65
	Restoring an existing DDL environment to a clean state .....	66
	Removing the DDL objects from the system .....	67
<b>Chapter 8</b>	<b>Uninstalling Oracle GoldenGate .....</b>	<b>69</b>
	Uninstalling Oracle GoldenGate from Linux or UNIX .....	69
	Uninstalling Oracle GoldenGate from Windows (non-cluster) .....	69
	Uninstalling Oracle GoldenGate from Windows Cluster .....	70
<b>Appendix 1</b>	<b>Oracle GoldenGate installed components .....</b>	<b>71</b>
	Oracle GoldenGate Programs and Utilities .....	71
	Oracle GoldenGate subdirectories .....	72
	Other Oracle GoldenGate files .....	75
	Oracle GoldenGate checkpoint table .....	79

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 Oracle

With Oracle GoldenGate, you can replicate Oracle DML and DDL operations. You can move, filter, and transform data between similar or dissimilar supported Oracle versions, or you can move, filter, and transform data between an Oracle database and a database of another type.

## 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.

- The Oracle GoldenGate GGSCI command interface fully supports up to 300 concurrent Extract and Replicat processes per instance of Oracle GoldenGate. An instance of Oracle GoldenGate equates to one Manager process, which is the main controller process.
- 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. Within its cache, it makes use of modern virtual memory techniques by:

- Allocating and managing active buffers efficiently.
- Recycling old buffers instead of paging to disk, when possible.
- Paging less-used information to disk, when necessary.

The cache manager keeps an Oracle GoldenGate process working within the soft limit of its global cache size, only allocating virtual memory (not physical memory) on demand. System calls to increase the cache size are made only as a last resort and, when used, are always followed by the release of virtual memory back to the system.

The system must have sufficient swap space for each Oracle GoldenGate Extract and Replicat process that will be running. To determine the required swap space:

1. Start up one Extract or Replicat.
2. Run GGSCI.
3. View the report file and find the line `PROCESS VM AVAIL FROM OS (min)`.
4. Round up the value to the next full gigabyte if needed. For example, round up 1.76GB to 2 GB.
5. Multiply that value by the number of Extract and Replicat processes that will be running. The result is the maximum amount of swap space that could be required. To determine the number of processes you will need, consult the configuration chapters in the *Oracle GoldenGate Windows and UNIX Administrator's 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 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. See also the following guidelines.

## Storage for Oracle GoldenGate trails

To prevent trail activity from interfering with business applications, assign a separate disk or file system to contain the trail files. These files are created during processing to store all of the data that is captured by Oracle GoldenGate. The default size is 10 megabytes, but



can be changed during the configuration process. Trail files accumulate but can be purged according to rules set with the PURGEOLDEXTRACTS parameter.

Trail files can reside on drives that are local to the Oracle GoldenGate installation, or they can reside on NAS or SAN devices. You will specify the location of the trails when you configure Oracle GoldenGate.

For trails that are stored at the source location, there should be enough space to handle data accumulation should the network connection fail. In a typical configuration, a secondary Extract process (known as a data pump) sends data from a local trail over the network, and it will fail when the network does. However, the primary Extract that reads the transaction logs and writes to the local trail will continue to do so. This Extract should not be stopped during a failure; otherwise, transaction data might be missed if the transaction logs recycle or get removed from the system before the data is completely captured. There must be enough disk space to hold the data accumulation.

For trails at the target location, provide enough disk space to handle data accumulation according to the purge rules set with the PURGEOLDEXTRACTS parameter. Even with PURGEOLDEXTRACTS in use, data will always accumulate on the target because it is transferred across the network faster than it can be applied to the target database.

#### To estimate required trail space

1. Estimate the longest time that the network could be unavailable. Plan to store enough data to withstand the longest possible outage, because otherwise you will need to resynchronize the source and target data if the outage outlasts disk capacity.
2. Estimate how much transaction log volume your business applications generate in one hour.
3. Use the following formula to calculate the required disk space.

$$[\text{log volume in one hour}] \times [\text{number of hours downtime}] \times .4 = \text{trail disk space}$$

This equation uses a multiplier of 40 percent because only about 40 percent of the data in a transaction log is needed by Oracle GoldenGate.

**NOTE** This formula is a conservative estimate, and you should run tests once you have configured Oracle GoldenGate to determine exactly how much space you need.

#### 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. You can assign a directory by using the CACHEDIRECTORY option of the CACHEMGR parameter.

#### Oracle RAC requirements

To install Oracle GoldenGate in an Oracle Real Application Cluster (RAC) environment, install Oracle GoldenGate on the shared drive(s) that are accessed by the RAC nodes. This allows you to start the Oracle GoldenGate processes from any of the nodes. If the node where the processes are running fails, you can start them on another node without modifying parameter files, because the processing checkpoints are preserved in the installation directory.

## 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 the following unreserved and unrestricted TCP/IP ports:
  - One port for communication between the Manager process and other Oracle GoldenGate processes.
  - A range of ports for local Oracle GoldenGate communications: can be the default range starting at port 7840 or a customized range of up to 256 other 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. On UNIX systems, that user must be a member of the group that owns the Oracle instance. 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, depending on how database authentication is configured.

## Itanium requirements

To install Oracle GoldenGate on a Microsoft Itanium system, the `vcredist_IA64.exe` runtime library package must be installed. You can download this package from the Microsoft website. This package includes VisualStudio DLLs necessary for Oracle GoldenGate to operate on the Itanium platform. If these libraries are not installed, Oracle GoldenGate generates the following error.

```
"The application failed to initialize properly (0xc0150002). Click on Ok to terminate the application.
```

## 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. For example, on a Windows system with a RHAS 4.0 virtual machine running Oracle11g, you would install the RHAS 4.0 build for Oracle 11g, just as you would on an actual Linux machine.

## Database configuration

- To run Oracle GoldenGate for multiple Oracle instances on a Windows system, you must install an instance of Oracle GoldenGate for each one
- On 64-bit Sun Solaris, HP Tru64 (OSF/1), and LINUX machines with 32-bit Oracle databases, Oracle GoldenGate requires LD\_LIBRARY\_PATH to include the 32-bit Oracle libraries. You will be instructed to set LD\_LIBRARY\_PATH in the installation instructions in this manual.
- If the database is Oracle 10g or later and configured to use a Bequeath connection, the sqlnet.ora file must contain the bequeath\_detach=true setting.
- To support the default RMAN log retention feature on Oracle RAC, you must download and install the database patch that is provided in BUGFIX 11879974 before you add the Extract groups. For more information, see page 58.
- Additional database configuration requirements are explained elsewhere in this manual.

## Database client

The full Oracle client must be used with Oracle GoldenGate so that the Oracle GoldenGate programs have access to the Oracle XDK libraries. Do not use Oracle Instant Client, which lacks those libraries. You can download the full client from the Oracle website.

## 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)
  - Manager (source database, if using DDL support)
  - 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.

- Keep a record of the database users. They must be specified in the Oracle GoldenGate parameter files with the USERID parameter. Use the USERID parameter for the database user, and use the TRANLOGOPTIONS parameter with the ASMUSER and ASMPASSWORD options for the ASM user.

## Database privileges

Certain privileges are required for a regular Oracle instance, plus additional privileges for an ASM instance and additional privileges for Oracle Enterprise Edition 10.2 or later.

### Regular Oracle database instance

To assign the correct privileges to the Oracle GoldenGate database user, see Table 1.

**Table 1 Database privileges by Oracle GoldenGate process**

User privilege	Extract	Replicat	Manager
CREATE SESSION, ALTER SESSION	X	X	
RESOURCE	X	X <sup>1</sup>	
CONNECT	X	X <sup>2</sup>	
SELECT ANY DICTIONARY	X	X	
FLASHBACK ANY TABLE or FLASHBACK ON <owner.table>	X		
SELECT ANY TABLE or SELECT ON <owner.table>	X	X	
SELECT on dba_clusters (Oracle 10gR2 and later)	X		
INSERT, UPDATE, DELETE ON <target tables>		X	
CREATE TABLE <sup>3</sup>		X	
Privileges required to issue DDL operations to target tables (DDL support only).		X	
EXECUTE on DBMS_FLASHBACK package <sup>4</sup>	X		
GGG_GGSUSER_ROLE <sup>5</sup>	X		
DELETE ON Oracle GoldenGate DDL objects <sup>6</sup>			X
Oracle 10 g ASM privileges	X		
LOCK ANY TABLE <sup>7</sup>		X	

**Table 1 Database privileges by Oracle GoldenGate process**

User privilege	Extract	Replicat	Manager
sys.dbms_internal_clkm <sup>8</sup>	X		
SELECT ANY TRANSACTION <sup>9</sup>	X		

- <sup>1</sup> If RESOURCE cannot be granted to Replicat, use ALTER USER <user> QUOTA {<size> | UNLIMITED} ON <tablespace>, where <tablespace> represents all tablespaces that contain target objects.
- <sup>2</sup> Required only if Replicat owns target objects or any PL/SQL procedures. If CONNECT cannot be granted, grant CREATE <object> for any object that Replicat will need to create.
- <sup>3</sup> Required if using ADD CHECKPOINTTABLE in GGSCI to enable the database checkpoint feature.
- <sup>4</sup> Oracle GoldenGate must make a call to DBMS\_FLASHBACK.GET\_SYSTEM\_CHANGE\_NUMBER.
- <sup>5</sup> Role for DML privileges on Oracle GoldenGate-owned DDL objects, if DDL support is used. Role is created during installation of those objects. User that installs this role must have SYSDBA privileges.
- <sup>6</sup> Required only if using parameters that maintain the Oracle GoldenGate DDL database objects.
- <sup>7</sup> Required if using the Oracle GoldenGate initial load method that inserts data by means of a direct bulk load to SQL\*Loader.
- <sup>8</sup> Required to replicate Oracle Transparent Data Encryption (TDE).
- <sup>9</sup> Required for Extract to use the newer ASM API that is available in certain Oracle versions. For more information, see the TRANLOGOPTIONS DBLOGREADER option in the Oracle GoldenGate reference documentation.

**Oracle ASM instance**

If Oracle 10g Automatic Storage Management (ASM) is in use, Oracle GoldenGate requires a user for the Extract process to access the ASM instance. Oracle GoldenGate does not support using operating-system authentication for the ASM user. You can use SYS user or any user with SYSDBA privileges in the ASM instance. See Table 2.

**Table 2 Extract database privileges — ASM instance**

ASM password configuration <sup>1</sup>	Permitted user
ASM instance and the database share a password file	You can use the Oracle GoldenGate database user if you grant that user SYSDBA, or you can use any other database user that has SYSDBA privileges.
ASM instance and the database have separate password files	You can overwrite the ASM password file with the database password file, understanding that this procedure changes the SYS password in the ASM instance to the value that is contained in the database password file, and it also grants ASM access to the other users in the database password file. Save a copy of the ASM file before overwriting it.

- <sup>1</sup> To view how the current ASM password file is configured, log on to the ASM instance and issue the following command in SQL\*Plus:
- ```
SQL> SELECT name, value FROM v$parameter
WHERE name = 'remote_login_passwordfile';
```

**Oracle Enterprise Edition 10.2 or later instance**

In Oracle Enterprise Edition 10.2 or later, the additional privileges in Table 3 are required for the Extract database user. In these Enterprise Edition versions, Oracle Recovery

Manager (RMAN) works with Extract to retain the archive logs that Extract needs for recovery. The special privileges are required for interaction with an underlying Oracle Streams Capture and with RMAN. (For more information, see “Ensuring data availability” on page 58.

**Table 3 Extract database privileges —Oracle EE 10.2 and later**

| Oracle EE version  | Privileges                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10.2               | <ol style="list-style-type: none"> <li>1. Run package to grant Oracle Streams admin privilege.<br/><code>exec dbms_streams_auth.grant_admin_privilege('&lt;user&gt;')</code></li> <li>2. Grant INSERT into logmnr_restart_ckpt\$.<br/><code>grant insert on system.logmnr_restart_ckpt\$ to &lt;user&gt;;</code></li> <li>3. Grant UPDATE on streams\$_capture_process.<br/><code>grant update on sys.streams\$_capture_process to &lt;user&gt;;</code></li> <li>4. Grant the 'become user' privilege.<br/><code>grant become user to &lt;user&gt;;</code></li> </ol> |
| 11.1 and 11.2.0.1  | <ol style="list-style-type: none"> <li>1. Run package to grant Oracle Streams admin privilege.<br/><code>exec dbms_streams_auth.grant_admin_privilege('&lt;user&gt;')</code></li> <li>2. Grant the 'become user' privilege.<br/><code>grant become user to &lt;user&gt;;</code></li> </ol>                                                                                                                                                                                                                                                                            |
| 11.2.0.2 and later | <p>Run package to grant Oracle Streams admin privilege.<br/><code>exec dbms_goldengate_auth.grant_admin_privilege('&lt;user&gt;')</code></p>                                                                                                                                                                                                                                                                                                                                                                                                                          |

## Supported Oracle data types

### Numeric data types

- NUMBER up to the maximum size permitted by Oracle
- BINARY FLOAT
- BINARY DOUBLE

### 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 types

- CHAR
- VARCHAR2
- LONG

- NCHAR
- NVARCHAR2

### Multi-byte character types

- NCHAR and NVARCHAR2 multi-byte character data types
- Multi-byte data stored in CHAR and VARCHAR2 columns

#### Limitations of support

- For Oracle GoldenGate to support multi-byte character data, the source and target databases must be identical. Transformation, filtering, and other manipulation are not supported.
- Multi-byte characters can be used with limitations in MAP and TABLE parameter statements such as string-based conversion functions and WHERE clauses. The code point for a multi-byte character must be represented within an escape sequence, for example “\u20ac.” For more information, see the Oracle GoldenGate *Windows and UNIX Administrator’s Guide* and the Oracle GoldenGate *Windows and UNIX Reference Guide*.
- Multi-byte data is supported whether the length semantics are in bytes or characters. If the semantics setting of the source database is BYTE and the setting of the target is CHAR, use the Replicat parameter SOURCEDEFS in your configuration, and place a DEFGEN-generated definitions file on the target. These steps are required to support the difference in semantics, whether or not the source and target data definitions are identical. Replicat refers to the definitions file to determine the upper size limit for fixed-size character columns. For information about SOURCEDEFS and DEFGEN, see the Oracle GoldenGate *Windows and UNIX Administrator’s Guide*.
- If the database has an NLS\_NCHAR\_CHARACTERSET value other than AL16UTF16, use the VARWIDTHNCHAR parameter to force NCHAR data to be written to the trail with 2-byte length information. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

### Binary data types

- RAW
- LONG RAW

### Date and timestamp data types

- DATE
- TIMESTAMP (see Limitations of support)

#### Limitations of support

- By default, only TIMESTAMP WITH TIME ZONE specified as a UTC offset is supported. To support TIMESTAMP WITH TIME ZONE specified as TZR (Region ID), you must use the Extract parameter TRANLOGOPTIONS with either the INCLUDEREGIONID or INCLUDEREGIONIDWITHOFFSET option. Without TRANLOGOPTIONS, Extract abends on this data type. See also Non-supported data types.

- **TIMESTAMP WITH TIME ZONE as TZR** is not supported by Oracle GoldenGate for initial loads, for use with the `SQLEXEC` feature, or for operations where the column must be fetched from the database. In these cases, the region ID is converted to a time offset by the Oracle database engine when the column is selected. Replicat will replicate the column data as date and time data with a time offset value.
- Because of the way that the Oracle database normalizes **TIMESTAMP WITH LOCAL TIME ZONE** to the local time zone of the database, the timestamps do not transfer correctly between databases that are in different time zones. Timestamps will transfer correctly between databases in the same time zone if you set the time zone of the Replicat session to the timezone of the database. Include the following parameter statement in the Replicat parameter file, placing it after the `USERID` parameter, but before the first `MAP` statement:  
`SQLEXEC "ALTER SESSION SET TIME_ZONE = <value of dbtimezone>"`
- Oracle GoldenGate does not support negative dates.

### Large object data types

- CLOB
- NCLOB
- BLOB

### Limitations of support

- `SECUREFILE` and `BASICFILE` are both supported.
- Store large objects out of row if possible.
- LOB capture differs between earlier and later versions of Oracle:
  - **Oracle versions earlier than 10g:** In-row LOBs are captured from the redo log, but out-of-row LOBs are fetched from the database. If a value gets deleted before a fetch occurs, Extract writes a null to the trail. If a value gets updated before the fetch occurs, Extract writes the updated value. To prevent these inaccuracies, try to keep Extract latency low. The Oracle GoldenGate documentation provides guidelines for tuning process performance.
  - **Oracle 10g and later, BASICFILE LOBs:**

All `BASICFILE` LOBs are captured from the redo log, whether stored in-row or out-of-row. However, Extract will continue to fetch LOBs under the following conditions, for which it will issue a warning:

    - ▶ Extract determines that a LOB instance is invalid.
    - ▶ The LOB data is missing from the redo log. This can occur if the `BASICFILE` LOB is created with the `no_logging` option.
    - ▶ The LOB is created with the `CACHE` attribute.
    - ▶ A LOB is only partially updated. Oracle GoldenGate does not support partial column data. Extract assumes LOB data to be incomplete if the LOB data does not start with a LOB reset record or does not start at the first byte and does not end at the last byte, according to the new LOB length. Partial updates can be generated by the following OCI calls: `OCILOBWrite()`, `OCILobAppend()`, `OCILobCopy()`, `OCILobLoadFromFile()`, `OCILobTrim()`, and by updates made through procedures in the `dbms_lob` package.





## User defined types

Oracle GoldenGate supports user defined types (UDT) when the source and target objects have the same structure. The schema names can be different.

### General limitations of support

- Extract must fetch UDTs (except for object tables) from the database, so you should configure and use a snapshot for data consistency.
- Because a UDT must be fetched, a table that contains one must have one of the following: a primary key, column(s) with a unique constraint, or a unique index.
- Oracle GoldenGate does not support UDTs with the following embedded scalar types: CLOB, CFILE, BFILE, or INTERVAL\_YM, INTERVAL\_DS, and OPAQUE (with the exception of XMLType, which is supported).
- Object or relational tables where the key contains a UDT, or where a UDT is the only column, are not supported.
- The RMTTASK parameter does not support user-defined types (UDT).
- CHAR and VARCHAR attributes that contain binary or unprintable characters are not supported.
- UDTs, including values inside object columns or rows, cannot be used within filtering criteria in TABLE or MAP statements, or as input or output for the Oracle GoldenGate column-conversion functions, SQLEXEC, or other built-in data-manipulation tools. Support is only provided for like-to-like Oracle source and targets.
- Oracle GoldenGate does not support REF types.

### Limitations for collection types

- When data in a nested table is updated, the row that contains the nested table must be updated at the same time.
- When VARRAYS and nested tables are fetched, the entire contents of the column are fetched each time, not just the changes.

### Limitations for object tables

- Oracle GoldenGate supports object tables in uni-directional and active-active configurations for Oracle 10g and later. Object tables are captured from the redo log, but certain data types that are fetched from the database when in regular relational tables, such as LOBs and collection types, will also be fetched when in object tables. Similarly, current limitations that apply to collection types when in regular tables also apply to these types when in object tables.
- An Oracle object table can be mapped to a non-Oracle object table in a supported target database.
- A primary key must be defined on the root-level object attributes of the object table, and cannot include leaf-level attributes. If no key is defined, Oracle GoldenGate will use all viable columns as a pseudo-key.
- Oracle GoldenGate does not support the replication of DDL operations for an object table. This limitation includes the database object versioning that is associated with ALTERs of object tables.
- Synonyms are not supported for object tables or relational tables that contain object tables.

### **Limitations for spatial types**

- Oracle GoldenGate supports SDO\_GEOMETRY, SDO\_TOPO\_GEOMETRY, and SDO\_GEORASTER (raster tables) for Oracle 10g and later.
- See additional configuration information for spatial types in “Ensuring correct handling of Oracle Spatial objects” on page 46.

### **Other supported data types**

- ROWID
- VARRAY
- INTERVAL DAY and INTERVAL YEAR if the size of the target column is equal to, or greater than, that of the source.

## **Non-supported Oracle data types**

- ANYDATA
- ANYDATASET
- ANYTYPE
- BFILE
- BINARY\_INTEGER
- MLSLABEL
- ORDDICOM
- PLS\_INTEGER
- TIMEZONE\_ABBR
- URITYPE
- UROWID

## **Supported objects and operations for Oracle DML**

### **Tables, views, and materialized views**

Oracle GoldenGate supports the following DML operations made to regular tables, index-organized tables (created with the ORGANIZATION INDEX clause of CREATE TABLE), clustered tables, and materialized views.

- INSERT
- UPDATE
- DELETE
- Associated transaction control operations

### **Limitations of support for regular tables**

- Oracle GoldenGate supports tables that contain any number of rows up to 2 MB in length. Each character LOB/LONG column contributes up to 4 KB to this limit, and each binary LOB column contributes up to 8 KB. This row-size limitation mostly affects update operations on columns that are being used as a row identifier. This identifier can be a primary or unique key, a key defined within the Oracle GoldenGate parameter

file, or all of the columns if no key is defined. If a row identifier is updated, the 2 MB length must include not only the after image, but also the full before image, which is required to find the correct key on the target for the update.

- LOB columns are supported in their full size.
- Oracle GoldenGate supports the maximum number of columns per table that is supported by the database.
- Oracle GoldenGate supports the maximum column size that is supported by the database.
- Oracle GoldenGate supports tables that contain only one column, except when the column contains one of the following data types:
  - LOB
  - LONG
  - Nested table
  - User defined data type
  - VARRAY
  - XML
- Oracle GoldenGate supports tables with unused columns, but the support is disabled by default, and Extract abends on them. You can use the `DBOPTIONS` parameter with the `ALLOWUNUSEDCOLUMN` option to force Extract to generate a warning and continue processing. When using `ALLOWUNUSEDCOLUMN`, either the same unused column must exist in the target table, or a source definitions file must be created for Replicat with the `DEFGEN` utility. You can include the appropriate `ALTER TABLE...SET UNUSED` statements in a DDL replication configuration.
- Oracle GoldenGate supports tables with interval partitioning. Make certain that the `WILDCARDRESOLVE` parameter remains at its default of `DYNAMIC`.
- Oracle GoldenGate supports tables with virtual columns, but does not capture change data for these columns, because the database does not write it to the transaction log. You can use the `FETCHCOLS` option of the `TABLE` parameter to fetch the value of a virtual column. Replicat does not apply DML to a virtual 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 virtual column when fetched can be applied to a target column that is not a virtual 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 virtual 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 virtual columns, because the database does not permit DML on that type of column.
- Oracle GoldenGate does not permit a virtual column to be used in a `KEYCOLS` clause in a `TABLE` or `MAP` statement.
- If a unique key includes a virtual column, and Oracle GoldenGate must use that key, the virtual column will be ignored. This might affect data integrity if the remaining columns do not enforce uniqueness. Fetching only provides an after value, and Oracle GoldenGate requires before and after values of keys.
- If a unique index is defined on any virtual columns, it will not be used.

- If a unique key or index contains a virtual column and is the only unique identifier on a table, Oracle GoldenGate must use all of the columns as an identifier to find target rows. Because a virtual column cannot be used in this identifier, it is possible that Replicat could apply operations containing this identifier to the wrong target rows.
- Tables created as EXTERNAL are not supported.
- A key cannot contain a column that is part of an invisible index.
- Tables created with table compression or OLTP table compression are not supported.
- Oracle GoldenGate supports delivery to Oracle Exadata with EHCC compression enabled for insert operations. In order for Replicat to apply data so that it is compressed with EHCC, you must use the INSERTAPPEND parameter for Replicat, which causes Replicat to use an APPEND hint for inserts. See the Oracle GoldenGate *Windows and UNIX Reference Guide*.
- Oracle GoldenGate supports Transparent Data Encryption (TDE) applied at the column and tablespace level. Column-level encryption is supported for all versions of Oracle 10.2.0.5, 11.1, and 11.2. Tablespace-level encryption is supported for all versions of Oracle 10.2.0.5 and 11.1.0.2. For more information, see “Replicating TDE-encrypted data” on page 43.
- Oracle GoldenGate supports the synchronization of TRUNCATE statements as part of the full DDL synchronization feature or as standalone functionality that is independent of full DDL synchronization. The standalone TRUNCATE feature supports the replication of TRUNCATE TABLE, but no other TRUNCATE options. The full DDL feature supports TRUNCATE TABLE, ALTER TABLE TRUNCATE PARTITION, and other DDL. To avoid errors from duplicate operations, only one of these features can be active at the same time. The GETTRUNCATES parameter controls the standalone TRUNCATE feature. See the Oracle GoldenGate *Windows and UNIX Reference Guide* for additional guidelines for TRUNCATE support.
- Oracle GoldenGate supports the capture of direct-load INSERTs for Oracle versions 9iR2, 10gR1, and 10gR2 and later. Supplemental logging must be enabled, and the database must be in archive log mode. The affected tables cannot contain LOBs if the database is 9iR2 or 10gR1, but LOBs are supported for later versions. The following direct-load methods are supported.
  - /\*+ APPEND \*/ hint
  - /\*+ PARALLEL \*/ hint (Non-RAC only)
  - SQLLDR with DIRECT=TRUE

#### **Limitations of support for views**

- Oracle GoldenGate can replicate to a view as long as it is inherently updatable.
- Oracle GoldenGate supports capture from a table in the source database to an inherently updatable view in the target database.
- The structures of the table and the view must be identical.
- A key must be defined on the unique columns in the view. This is done by means of a KEYCOLS clause in the MAP statements.

#### **Limitations of support for materialized views**

- Materialized views created WITH ROWID are not supported.
- The materialized view log can be created WITH ROWID.
- The source table must have a primary key.

- Truncates of materialized views are not supported. You can use a DELETE FROM statement.
- Some Oracle GoldenGate initial-load methods do not support LOBs in a materialized view.
- For Replicat, the materialized view must be updateable.
- DML (but not DDL) from a full refresh of a materialized view is supported for Oracle 10g and later. If DDL support for this feature is required, open an Oracle GoldenGate support case.

#### **Limitations of support for index-organized tables**

- IOTs are supported for Oracle versions 10.2 and later.
- Oracle GoldenGate supports IOTs that are created with the MAPPING TABLE option, but it only captures changes made to the base IOT, not changes made to the mapping table. However, Oracle will maintain the mapping table on the target, if one is being used.
- IOTs that are stored in a compressed format are not supported (for example, in a compressed tablespace).

**NOTE** A compressed IOT is different from an IOT that has key compression defined with the COMPRESS option. IOTs with key compression are supported.

- Because an IOT does not have a rowid, Oracle GoldenGate must fetch certain data types in an IOT from the database. The fetch uses the key value as the row identifier, which increases the potential for “row not found” errors. Oracle GoldenGate provides the FETCHOPTIONS parameter to handle these errors. Data types that are fetched are:
  - BLOB
  - CLOB
  - NCLOB
  - XMLType
  - UDT
  - Nested table
  - VARRAY
- (Oracle 10g and later) TRUNCATES of an IOT where one partition is empty will not be captured.

#### **Limitations of support for clustered tables**

- Indexed and hash clusters are both supported.
- Encrypted and compressed clustered tables are not supported.

### **Sequences**

Oracle GoldenGate supports the replication of sequence values by means of the SEQUENCE parameter. Oracle GoldenGate ensures that the target sequence values will always be higher than those of the source (or equal to them, if the cache is 0).

**NOTE** DDL support for sequences (CREATE, ALTER, DROP, RENAME) is compatible with, but not required for, replicating sequence values. To replicate just sequence values, you *do not* need to install the Oracle GoldenGate DDL support environment. You can just use the SEQUENCE parameter.

**Limitations of support for sequences**

- The cache size and the increment interval of the source and target sequences must be identical.
- The cache can be any size, including 0 (NOCACHE).
- The sequence can be set to cycle or not cycle, but the source and target databases must be set the same way.
- To add SEQUENCE to a configuration in which DDL support is enabled, you must re-install the Oracle GoldenGate DDL objects in INITIALSETUP mode.

**Non-supported objects and operations for Oracle DML**

- REF
- Tablespaces and tables created or altered with COMPRESS
- Synonyms
- Database Replay
- Distributed transactions
- XA distributed transactions
- Transparent Application Failover

**Supported objects and operations for Oracle DDL**

All Oracle GoldenGate topology configurations are supported for Oracle DDL replication. Active-active (bi-directional) replication of Oracle DDL is supported between two (and only two) databases that contain identical metadata.

Oracle GoldenGate supports DDL operations of up to 2 MB in size on the following objects:

|           |             |                    |
|-----------|-------------|--------------------|
| clusters  | tables      | triggers           |
| functions | tablespaces | types              |
| indexes   | roles       | views              |
| packages  | sequences   | materialized views |
| procedure | synonyms    | users              |

The 2 MB size limitation includes packages, procedures, and functions.

**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.

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

## Non-supported objects and operations for Oracle DDL

### Oracle-reserved schemas

The following schema names are considered Oracle-reserved and must be excluded from the Oracle GoldenGate DDL configuration. Oracle GoldenGate will ignore these schemas.

| Schema Name       | Schema Name      | Schema Name |
|-------------------|------------------|-------------|
| ANONYMOUS         | EXFSYS           | PUBLIC      |
| AURORA            | MDSYS            | REPADMIN    |
| \$JIS             | ODM              | SYS         |
| \$UTILITY         | ODM_MTR          | SYSMAN      |
| \$AURORA          | OLAPSYS          | SYSTEM      |
| \$ORB             | ORDPLUGINS       | TRACESVR    |
| \$UNAUTHENTICATED | ORDSYS           | WKPROXY     |
| CTXSYS            | OSE\$HTTP\$ADMIN | WKSYS       |
| DBSNMP            | OUTLN            | WMSYS       |
| DMSYS             | PERFSTAT         | XDB         |
| DSSYS             |                  |             |

### Oracle recycle bin

Because of a known issue in Oracle 10g, the Oracle recycle bin must be turned off to support Oracle GoldenGate DDL replication. If the recycle bin is enabled, the Oracle GoldenGate DDL trigger session receives implicit recycle bin DDL operations that cause the trigger to fail.

#### To turn off the recycle bin:

- Oracle 10g Release 2 and later: Set the RECYCLEBIN initialization parameter to OFF.
- Oracle 10g Release 1: Set the \_RECYCLEBIN initialization parameter to FALSE.

Consult the Oracle 10g database documentation for the correct syntax.

### Other non-supported DDL

Oracle GoldenGate does not support the following:

- ALTER TABLE ... MOVE TABLESPACE
- DDL on nested tables.
- DDL that involves password-based column encryption, such as:
  - CREATE TABLE t1 ( a number, b varchar2(32) ENCRYPT IDENTIFIED BY my\_password );
  - ALTER TABLE t1 ADD COLUMN c varchar2(64) ENCRYPT IDENTIFIED BY my\_password;
- ALTER DATABASE and ALTER SYSTEM (these are not considered to be DDL)
- DDL on standby databases.



## 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. The exception is Oracle 11g, where case-sensitive passwords are supported in Oracle GoldenGate input that requires passwords.

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

**Table 4 Supported non-alphanumeric characters in object names and non-key column names<sup>1</sup>**

| Character | Description |
|-----------|-------------|
| ~         | Tilde       |

**Table 4 Supported non-alphanumeric characters in object names and non-key column names<sup>1</sup>**

| Character | Description                           |
|-----------|---------------------------------------|
| < >       | 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) |

<sup>1</sup> 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 5 Non-supported characters in object and column names<sup>1</sup>**

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

---

<sup>1</sup> 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 ORACLE\_HOME and ORACLE\_SID
- Setting library paths for dynamic builds
- Installing the Oracle GoldenGate software
- Configuring Manager and other processes
- Installing support for Oracle sequences
- (Optional) Installing Oracle GoldenGate DDL support (see Chapter 3)

**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 Oracle GoldenGate Media Pack that you want.
  - 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 ORACLE\_HOME and ORACLE\_SID

Make certain that the ORACLE\_HOME and ORACLE\_SID system environment variables are set to the correct Oracle instance. The Oracle GoldenGate processes refer to them when connecting to the database.

### To specify Oracle variables on UNIX-based systems

- If there is one instance of Oracle on the system, set ORACLE\_HOME and ORACLE\_SID at the system level. If you cannot set them that way, use the following SETENV statements in the parameter file of every Extract and Replicat group that will be connecting to the instance.

```
SETENV (ORACLE_HOME = "<path to Oracle home location>")  
SETENV (ORACLE_SID = "<SID>")
```

The SETENV parameters override the system settings and allow the Oracle GoldenGate process to set the variables at the session level when it connects to the database.

- If there are multiple Oracle instances on the system with Extract and Replicat processes connecting to them, you will need to use a SETENV statement in the parameter file of each process group and point it to the correct instance. For example, the following shows parameter files for two Extract groups, each capturing from a different Oracle instance.

Group 1:

```
EXTRACT ora9a  
SETENV (ORACLE_HOME = "/home/oracle/ora9/product")  
SETENV (ORACLE_SID = "ora9a")  
USERID gggsa, PASSWORD gggsa  
RMTHOST sysb  
RMTTRAIL /home/ggs/dirdat/rt  
TABLE hr.emp;  
TABLE hr.salary;
```

Group 2:

```
EXTRACT ora9b
SETENV (ORACLE_HOME = "/home/oracle/ora9/product")
SETENV (ORACLE_SID = "ora9b")
USERID ggsb, PASSWORD ggsb
RMTHOST sysb
RMTTRAIL /home/ggs/dirdat/st
TABLE fin.sales;
TABLE fin.cust;
```

**To specify Oracle variables on Windows systems**

- If there is one instance of Oracle on the system, the Registry settings for ORACLE\_HOME and ORACLE\_SID should be sufficient for Oracle GoldenGate. If those settings are incorrect in the Registry and cannot be changed, you can set an override as follows.
  - On the desktop or Start menu (depending on the Windows version), right-click **My Computer**, and then select **Properties**.
  - In **Properties**, click the **Advanced** tab.
  - Click **Environment Variables**.
  - Under **System Variables**, click **New**.
  - For **Variable Name**, type ORACLE\_HOME.
  - For **Variable Value**, type the path to the Oracle binaries.
  - Click **OK**.
  - Click **New** again.
  - For **Variable Name**, type ORACLE\_SID.
  - For **Variable Value**, type the instance name.
  - Click **OK**.
- If there are multiple Oracle instances on the system with Extract and Replicat processes connecting to them, do the following.
  - Use the preceding procedure (single Oracle instance on system) to set the ORACLE\_HOME and ORACLE\_SID system variables to the first Oracle instance.
  - Start all of the Oracle GoldenGate processes that will connect to that instance.
  - Repeat the procedure for the next Oracle instance, but first **Edit** the existing ORACLE\_HOME and ORACLE\_SID variables to specify the new information.
  - Start the Oracle GoldenGate processes that will connect to that instance.
  - Repeat the **Edit** and startup procedure for the rest of the Oracle instances.

## 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.

When Oracle GoldenGate is running on the same server as the database, all of the following must have the same bit type, either all 32-bit, all 64-bit, or all IA64:

- Oracle library versions
- Oracle GoldenGate version
- Database versions

When Oracle GoldenGate connects remotely to the database server through SQL\*Net, the following are required:

- Replicat: The Oracle client library and the Oracle GoldenGate build must have the same Oracle version, bit type (32-bit, 64-bit, IA64), and operating system version.
- Extract: The Oracle client library and the Oracle GoldenGate build must have the same Oracle version, bit type (32-bit, 64-bit, IA64), and operating system version. In addition, both operating systems must be the same endian.

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:

| <b>Command</b>         | <b>Requires GG libraries in environment variable?</b> |
|------------------------|-------------------------------------------------------|
| \$ 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 <sup>1</sup> | Environment variable         |
|-----------------------|------------------------------|
| IBM AIX               | LIBPATH                      |
| IBM z/OS              |                              |
| HP-UX                 | SHLIB_PATH                   |
| Sun Solaris           | LD_LIBRARY_PATH <sup>2</sup> |
| HP Tru64 (OSF/1)      |                              |
| LINUX                 |                              |

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

<sup>2</sup> 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 GGSMGR.
- 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 6** 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.<br><br>category.dll<br>ggsmg.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. <sup>1</sup><br><br>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. |

**Table 6** INSTALL options (continued)

| Option                 | Description                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.<br><br>By default, the Manager service is installed to use the Local System account. |
| PASSWORD<br><password> | Specifies the password for the user that is specified with USER.                                                                                                                                                                                 |

<sup>1</sup> 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.

## Installing support for Oracle sequences

To support Oracle sequences, you must install some database procedures. These procedures support the Oracle GoldenGate FLUSH SEQUENCE command, which you issue immediately after you start the Oracle GoldenGate processes for the first time (typically when you perform the initial data synchronization procedure).

### To install Oracle sequence objects

Follow these instructions on the **source** and **target** systems.

1. On the source and target Oracle systems, connect to SQL\*Plus as SYSDBA.
2. If you already assigned a database user to support the Oracle GoldenGate DDL replication feature, you can skip this step. Otherwise, on both systems create a database user that can also be the DDL user.

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

3. From the Oracle GoldenGate installation directory, run GGSCI.

4. In GGSCI, issue the following command.

```
EDIT PARAMS ./GLOBALS
```

5. In the file, enter the GGSHEMA parameter and specify the schema of the DDL user that you created earlier in this procedure.

```
GGSHEMA <schema>
```

6. Save and close the file.

7. On both systems, run the `sequence.sql` script from the root of the Oracle GoldenGate installation directory. This script creates some procedures for use by Oracle GoldenGate processes. (Do not run them yourself.) You are prompted for the user information that you created in the first step.

```
SQL> @sequence.sql
```

8. On the source system, grant EXECUTE privilege on the `updateSequence` procedure to a database user that can be used to issue the `DBLOGIN` command. Remember or record this user. You use `DBLOGIN` to log into the database prior to issuing the `FLUSH SEQUENCE` command, which calls the procedure.

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

9. On the target system, grant EXECUTE privilege on the `replicateSequence` procedure to the Replicat database user.

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

## 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

# Installing Oracle GoldenGate DDL support for an Oracle database

.....

This chapter contains instructions for installing the objects that support DDL replication. To configure Oracle GoldenGate to capture and replicate DDL, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

**NOTE** DDL support for sequences (CREATE, ALTER, DROP, RENAME) is compatible with, but not required for, replicating the sequence values themselves. To replicate just sequence values, you *do not* need to install the Oracle GoldenGate DDL support environment. You can just use the SEQUENCE parameter.

## Overview of the DDL objects

To install the Oracle GoldenGate DDL environment, you will be installing the database objects shown in Table 7.

**Table 7 DDL synchronization objects**

| Object                   | Purpose                                                                                                                                         | Default name                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| DDL marker table         | Stores DDL information. This table only receives inserts.                                                                                       | GG\$_MARKER                                                          |
| Sequence on marker table | Used for a column in the marker table.                                                                                                          | GG\$_DDL_SEQ                                                         |
| DDL history table        | Stores object metadata history. This table receives inserts, updates, deletes.                                                                  | GG\$_DDL_HIST                                                        |
| Object ID history table  | Contains object IDs of configured objects.                                                                                                      | GG\$_DDL_HIST_ALT                                                    |
| DDL trigger              | Fires on DDL operations. Writes information about the operation to the marker and history tables. Installed with the trigger are some packages. | GG\$_DDL_TRIGGER_BEFORE                                              |
| DDL schema               | Contains the DDL synchronization objects.                                                                                                       | None; must be specified during installation and in the GLOBALS file. |

**Table 7 DDL synchronization objects (continued)**

| Object               | Purpose                                                                              | Default name       |
|----------------------|--------------------------------------------------------------------------------------|--------------------|
| User role            | Establishes the role needed to execute DDL operations.                               | GGG_GGSUSER_ROLE   |
| Internal setup table | Database table for internal use only.                                                | GGG_SETUP          |
| ddl_pin              | Pins DDL tracing, the DDL package, and the DDL trigger for performance improvements. | ddl_pin            |
| ddl_cleartrace.sql   | Removes the DDL trace file.                                                          | ddl_cleartrace.sql |
| ddl_status.sql       | Verifies that the Oracle GoldenGate DDL objects are installed                        | ddl_status.sql     |
| marker_status.sql    | Verifies that the marker table is installed.                                         | marker_status.sql  |
| ddl_tracelevel.sql   | Sets the level for DDL tracing.                                                      | ddl_tracelevel.sql |

## Installing the DDL objects

1. Choose an Oracle GoldenGate schema or another schema for the DDL objects.
2. Grant the following permission on the Oracle GoldenGate schema.  

```
GRANT EXECUTE ON utl_file TO <schema>;
```
3. Choose a tablespace for the DDL objects that can accommodate the growth of the GGS\_DDL\_HIST and GGS\_MARKER tables. The GGS\_DDL\_HIST table, in particular, will grow in proportion to overall DDL activity. If the tablespace that contains these objects fills up, no DDL operations can be issued on the database, and the business applications will pause.
4. Open the GLOBALS file in the home directory of this instance of Oracle GoldenGate. If a GLOBALS file does not exist, create one. For information on creating or editing a GLOBALS file, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
5. Specify the name of the DDL schema by adding the following parameter to the GLOBALS file.  

```
GGSCHEMA <schema_name>
```
6. (Optional) To change the names of other objects listed in Table 7, *the changes must be made now, before proceeding with the rest of the installation. Otherwise, you will need to stop Oracle GoldenGate DDL processing and reinstall the DDL objects.* It is recommended that you accept the default names of the database objects. To change any name in Table 7 (except the schema), do one or both of the following:

- Record all name changes in the params.sql script. Edit this script and change the appropriate parameters. Do not run this script.
- List the names shown in Table 8 in the GLOBALS file. The correct parameters to use are listed in the Parameter column of the table.

**Table 8 GLOBALS parameters for changing DDL object names**

| Object        | Parameter                                 |
|---------------|-------------------------------------------|
| Marker table  | MARKERTABLE <new_table_name> <sup>1</sup> |
| History table | DDLTABLE <new_table_name>                 |

<sup>1</sup> Do not qualify the name of any of these tables. The schema name for these table must be either the one that is specified with GGSCHEMA or the schema of the current user, if GGSCHEMA is not specified in GLOBALS.

7. Change directories to the Oracle GoldenGate installation directory.
8. Exit all Oracle sessions, including those of SQL\*Plus, those of business applications, those of the Oracle GoldenGate processes, and those of any other software that uses Oracle. Prevent the start of any new sessions.
9. Run SQL\*Plus and log in as a user that has SYSDBA privilege. This privilege is required to install the DDL trigger in the SYS schema, which is required by Oracle. All other DDL objects are installed in the schema that you created in step 1.

10. Run the marker\_setup.sql script.

```
@marker_setup.sql
```

- Supply the name of the Oracle GoldenGate schema when prompted, and then press **Enter** to execute the script.
- This script installs support for the Oracle GoldenGate DDL marker system.

11. Run the ddl\_setup.sql script.

```
@ddl_setup.sql
```

You are prompted to:

- Disable the Oracle recycle bin if the database is Oracle 10g.
- Stop current Oracle GoldenGate DDL replication.
- Specify the name of the DDL schema from step 1.
- Specify the installation mode: To install DDL objects the first time, use the INITIALSETUP mode.

**NOTE** INITIALSETUP mode assumes that no Oracle GoldenGate DDL objects exist and drops them if they do. If the DDL objects exist and you want to reinstall them, but preserve any DDL history, use the procedure in “Restoring an existing DDL environment to a clean state” on page 66.

12. Run the role\_setup.sql script.

```
@role_setup.sql
```

- At the prompt, supply the DDL schema name.



- The script drops and creates the role that is needed for DDL synchronization, and it grants DML permissions on the Oracle GoldenGate DDL objects.

**13.** Grant the role that was created (default name is GGS\_GGSUSER\_ROLE to all Oracle GoldenGate Extract users. You may need to make multiple grants if the processes have different user names.

```
GRANT <role> TO <user>;
```

**14.** Run the ddl\_enable.sql script to enable the DDL trigger.

```
@ddl_enable.sql
```

**To install and use the optional performance tool**

To improve the performance of the DDL trigger, make the ddl\_pin script part of the database startup. It must be invoked with the Oracle GoldenGate DDL user name, as in:

```
@ddl_pin <DDL_user>
```

This script pins the PL/SQL package that is used by the trigger into memory. If executing this script from SQL\*Plus, connect as SYSDBA from the Oracle GoldenGate installation directory. This script relies on the Oracle dmbs\_shared\_pool system package, so install that package before using ddl\_pin.

## CHAPTER 4

# Preparing the database for Oracle GoldenGate

.....

This chapter contains steps to take so that the database with which Oracle GoldenGate interacts is configured properly to support Oracle GoldenGate capture and replication. Some steps apply to just a source system, some just to a target, and some to both.

## Preparing constraints for Oracle GoldenGate

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

### Disabling triggers and cascade constraints

You will need to make some changes to the triggers, cascade update, and cascade delete constraints on the target tables. Oracle GoldenGate replicates DML that results from a trigger or a 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.

Oracle GoldenGate provides some options to handle triggers or cascade constraints automatically, depending on the Oracle version:

- For Oracle 10.2.0.5 and later patches to 10.2.0.5, and for Oracle 11.2.0.2 and later 11gR2 versions, you can use the Replicat parameter DBOPTIONS with the SUPPRESSTRIGGERS option to cause Replicat to disable the triggers during its session.
- For Oracle 9.2.0.7 and later, you can use the Replicat parameter DBOPTIONS with the DEFERRREFCONST option to delay the checking and enforcement of cascade update and cascade delete constraints until the Replicat transaction commits.
- For earlier Oracle versions, you must disable triggers and integrity constraints or alter them manually to ignore the Replicat database user.

### Deferring constraint checking

If constraints are DEFERRABLE on the source, the constraints on the target must also be DEFERRABLE. You can use one of the following parameter statements to defer constraint

.....

checking until a Replicat transaction commits:

- Use SQLEXEC at the root level of the Replicat parameter file to defer the constraints for an entire Replicat session.  
SQLEXEC ("alter session set constraint deferred")
- For Oracle 9.2.0.7 and later, you can use the Replicat parameter DBOPTIONS with the DEFERREFCONST option to delay constraint checking for each Replicat transaction.

Replicat might need to set constraints to DEFERRED if it is possible that an update transaction could affect the primary keys of multiple rows. Called a *transient primary key update* in Oracle GoldenGate terminology, this kind of operation typically uses an  $x+n$  formula or other form of manipulation that shifts the values and causes a new value to be the same as an old one.

The following illustrates a sequence of value changes that can cause this condition if constraints are not deferred. The example assumes the primary key column is "CODE" and the current key values (before the updates) are 1, 2, and 3.

```
update item set code = 2 where code = 1;  
update item set code = 3 where code = 2;  
update item set code = 4 where code = 3;
```

In this example, when Replicat applies the first update to the target, there is an error because the key value of 2 already exists in the table. The Replicat transaction returns constraint violation errors. By default, Replicat does not handle these violations and abends.

To enable Replicat to manage these updates:

- Use the Replicat parameter HANDLEPKUPDATE to enable Replicat to handle the transient primary key updates.
- Create the constraints as DEFERRABLE INITIALLY IMMEDIATE on the target tables. The constraints are checked when Replicat commits the transaction. You can:
  - Use SQLEXEC at the root level of the Replicat parameter file to defer the constraints for an entire Replicat session.  
SQLEXEC ("alter session set constraint deferred")
  - For Oracle 9.2.0.7 and later, you can use the Replicat parameter DBOPTIONS with the DEFERREFCONST option to delay constraint checking for each Replicat transaction.

If constraints are not DEFERRABLE, Replicat handles the errors according to rules that are specified with the HANDLECOLLISIONS and REPERROR parameters, if they exist, or else it abends.

## 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 automatically selects a row identifier to use in the following order of priority:

1. Primary key
2. First unique key alphanumerically with no virtual columns, no UDTs, no function-based columns, and no nullable columns
3. First unique key alphanumerically with no virtual columns, no UDTs, and no 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) Oracle GoldenGate constructs a pseudo key of all columns that the database allows to be used in a unique key, excluding virtual columns, UDTs, function-based columns, and any columns that are explicitly excluded from the Oracle GoldenGate configuration.

Depending on whether schema-level or table-level logging was activated, there might be just one key or multiple keys logged to the redo log. See “Configuring the database to log key values” on page 38.

**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 an appropriate key, or if you prefer the existing key(s) 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 the database to log key values**

GGSCI provides commands to configure the source database to log the appropriate key values whenever it logs a row change, so that they are available to Oracle GoldenGate in the redo record. By default, the database only logs column values that are changed. The appropriate command must be issued before you start Oracle GoldenGate processing.

#### **ADD TRANDATA**

The ADD TRANDATA command enables table-level logging and is generally appropriate if you will not be using the Oracle GoldenGate DDL replication feature, or if you want to use that feature and your data meets certain requirements documented for this command in the *Windows and UNIX Reference Guide*.

#### **ADD SCHEMATRANDATA**

The ADD SCHEMATRANDATA command enables schema-level logging. It logs more key values to the redo log than ADD TRANDATA does, and it affects all of the current and future tables of a given schema. Because ADD SCHEMATRANDATA logs key values atomically when each DDL operation occurs, it is the preferred logging method to use if you will be using the Oracle GoldenGate DDL replication feature. (If the database system can tolerate the extra redo data, you can also use ADD SCHEMATRANDATA without using the DDL replication feature.)

**NOTE** See the ADD SCHEMATRANDATA and ADD TRANDATA commands in the Oracle GoldenGate *Windows and UNIX Reference Guide* for additional usage considerations.

**To initiate the logging of key values**

1. (ADD SCHEMATRANDATA only) Apply Oracle Patch 10423000 to the source Oracle database.
2. On the source system, run GGSCI from the Oracle GoldenGate directory.
3. In GGSCI, issue the following command to log on to the database.

```
DBLOGIN USERID <user>, PASSWORD <password>
```

**Where:** <user> is a database user who has the privilege to enable table-level or schema-level supplemental logging, depending on the logging command that you will be choosing from Table 9, and <password> is that user's password.

4. Issue the ADD TRANDATA or ADD SCHEMATRANDATA command.

**Table 9 Trandata commands**

| Command                                                 | Syntax                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADD TRANDATA <table><br>[, COLS <columns>]<br>[, NOKEY] | Where:<br><ul style="list-style-type: none"> <li>◆ &lt;table&gt; is the owner and name of the table. You can use a wildcard for the table name, but not the owner name.</li> <li>◆ COLS &lt;columns&gt; logs non-key columns that are required for a KEYCOLS clause or for filtering and manipulation.</li> <li>◆ NOKEY prevents the logging of the primary key or unique key. Requires a KEYCOLS clause in the TABLE and MAP parameters and a COLS clause in the ADD TRANDATA command to log the KEYCOLS columns.</li> </ul> |
| ADD SCHEMATRANDATA<br><schema>                          | Where:<br><schema> is the schema for which you want to enable supplemental logging.                                                                                                                                                                                                                                                                                                                                                                                                                                           |

5. Log in to SQL\*Plus as a user with ALTER SYSTEM privilege, and then issue the following command to enable minimal supplemental logging at the database level. This logging is required to process updates to primary keys and chained rows.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

6. To start the supplemental logging, switch the log files.

```
ALTER SYSTEM SWITCH LOGFILE;
```

7. Verify that supplemental logging is enabled at the database level with this command:

```
SELECT SUPPLEMENTAL_LOG_DATA_MIN FROM V$DATABASE;
```

- For Oracle 9i, the output must be YES.
- For Oracle 10g, the output must be YES or IMPLICIT.

8. If using ADD TRANDATA with the COLS option, create a unique index for those columns on the target to optimize row retrieval. If you are logging those columns as a substitute key for a KEYCOLS clause, make a note to add the KEYCOLS clause to the TABLE and MAP statements when you configure the Oracle GoldenGate processes.

### Limiting row changes in tables that do not have a key

If a target table does not have a primary key or a unique key, duplicate rows can exist. In this case, Oracle GoldenGate could update or delete too many target rows, 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.

## 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.
- For more information, refer to Oracle's *Database Globalization Support Guide*.

### To view globalization settings

To determine the globalization settings of the database and whether it is using byte or character semantics, use the following commands in SQL\*Plus:

```
SHOW PARAMETER NLS_LANGUAGE
SHOW PARAMETER NLS_TERRITORY
SELECT name, value$ from SYS.PROPS$ WHERE name = 'NLS_CHARACTERSET';
SHOW PARAMETER NLS_LENGTH_SEMANTICS
```

### To view globalization settings from GGSCI

The VIEW REPORT <group> command in GGSCI shows the current database language and character settings and indicates whether or not NLS\_LANG is set.

### To set NLS\_LANG

1. Set the NLS\_LANG parameter according to the documentation for your database version and operating system. On UNIX systems, you can set NLS\_LANG through the operating system or by using a SETENV parameter in the Extract and Replicat parameter files. For best results, set NLS\_LANG from the parameter file, where it is less likely to be changed than at the system level.

NLS\_LANG must be set in the format of:

```
<NLS_LANGUAGE>_<NLS_TERRITORY>.<NLS_CHARACTERSET>
```

This is an example in UNIX, using the SETENV parameter in the Oracle GoldenGate parameter file:

```
SETENV (NLS_LANG = "AMERICAN_AMERICA.AL32UTF8")
```

2. Stop and then start the Oracle GoldenGate Manager process so that the processes recognize the new variable.

**NOTE** Oracle GoldenGate reports Oracle error messages in U.S. English (AMERICAN\_AMERICA), regardless of the actual character set of the reporting database. Oracle GoldenGate performs any necessary language conversion internally without changing the language configuration of the database.

## Adjusting cursors

The Extract process maintains cursors for queries that fetch data and also for SQLEXEC operations. Without enough cursors, Extract must age more statements. Extract maintains as many cursors as permitted by the Extract MAXFETCHSTATEMENTS parameter. You can increase the value of this parameter as needed. Make an appropriate adjustment to the maximum number of open cursors that are permitted by the database.

## Setting fetch options

To process certain update records from the redo log, Oracle GoldenGate fetches additional row data from the source database. Oracle GoldenGate fetches data for the following:

- Operations that contain LOBs. (Fetching of LOBs does not apply to Oracle 10g and later databases, because LOBs are captured from the redo log of those versions.)
- User-defined types
- Nested tables
- XMLType objects

By default, Oracle GoldenGate uses Flashback Query to fetch the values from the undo (rollback) tablespaces. That way, Oracle GoldenGate can reconstruct a read-consistent row image as of a specific time or SCN to match the redo record.

### To configure the database for best fetch results

For best fetch results, configure the source database as follows:

1. Set a sufficient amount of redo retention by setting the Oracle initialization parameters UNDO\_MANAGEMENT and UNDO\_RETENTION as follows (in seconds).

```
UNDO_MANAGEMENT=AUTO  
UNDO_RETENTION=86400
```

UNDO\_RETENTION can be adjusted upward in high-volume environments.

2. Calculate the space that is required in the undo tablespace by using the following formula.

$$\langle \text{undo space} \rangle = \langle \text{UNDO\_RETENTION} \rangle * \langle \text{UPS} \rangle + \langle \text{overhead} \rangle$$

**Where:**

- <undo space> is the number of undo blocks.
- <UNDO\_RETENTION> is the value of the UNDO\_RETENTION parameter (in seconds).
- <UPS> is the number of undo blocks for each second.
- <overhead> is the minimal overhead for metadata (transaction tables, etc.).

Use the system view V\$UNDOSTAT to estimate <UPS> and <overhead>.

**3.** For tables that contain LOBs, do one of the following:

- Set the LOB storage clause to RETENTION. This is the default for tables that are created when UNDO\_MANAGEMENT is set to AUTO.
- If using PCTVERSION instead of RETENTION, set PCTVERSION to an initial value of 25. You can adjust it based on the fetch statistics that are reported with the STATS EXTRACT command (see Table 10). If the value of the STAT\_OPER\_ROWFETCH\_CURRENTBYROWID or STAT\_OPER\_ROWFETCH\_CURRENTBYKEY field in these statistics is high, increase PCTVERSION in increments of 10 until the statistics show low values.

**4.** Grant the following privileges to the Oracle GoldenGate Extract user:

```
GRANT FLASHBACK ANY TABLE TO <db_user>
```

Or ...

```
GRANT FLASHBACK ON <owner.table> TO <db_user>
```

Oracle GoldenGate provides the following parameters to manage fetching.

**Table 10 Oracle GoldenGate parameters and commands to manage fetching**

| Parameter or Command                                                          | Description                                                                                                                                                                                             |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATS EXTRACT command with REPORTFETCH option                                 | Shows Extract fetch statistics on demand.                                                                                                                                                               |
| STATOPTIONS parameter with REPORTFETCH option                                 | Sets the STATS EXTRACT command so that it always shows fetch statistics.                                                                                                                                |
| MAXFETCHSTATEMENTS parameter                                                  | Controls the number of open cursors for prepared queries that Extract maintains in the source database.                                                                                                 |
| FETCHOPTIONS parameter with USESNAPSHOT or NOUSESNAPSHOT option               | Controls the default fetch behavior of Extract for an Oracle 9i database: whether Extract performs a flashback query or fetches the current image from the table.                                       |
| FETCHOPTIONS parameter with the USELATESTVERSION or NOUSELATESTVERSION option | Handles the failure of an Extract flashback query, such as if the undo retention expired or the structure of a table changed. Extract can fetch the current image from the table or ignore the failure. |



**Table 10 Oracle GoldenGate parameters and commands to manage fetching**

| Parameter or Command           | Description                                                                                                               |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| REFFETCHEDCOLOPTIONS parameter | Controls the response by Replicat when it processes trail records that include fetched data or column-missing conditions. |

## Replicating TDE-encrypted data

Oracle GoldenGate supports the Transparent Data Encryption (TDE) at the column and tablespace level.

- Column-level encryption is supported for all versions of 10.2.0.5, 11.1, and 11.2.
- Tablespace-level encryption is supported for all versions of 11.1.0.2.

### Required database patches

To support TDE, Oracle Patch 10395645 must be applied to the database. If you cannot find this patch on the My Oracle Support website (<https://support.oracle.com>), submit a service request (SR) and request a backport.

### Overview of TDE support

TDE support involves two kinds of keys:

- The *encrypted key* can be a table key (column-level encryption), an encrypted redo log key (tablespace-level encryption), or both. A key is shared between the Oracle server and Extract.
- The *decryption key* is a password known as the *shared secret* that is stored securely in the Oracle and Oracle GoldenGate domains. Only a party that has possession of the shared secret can decrypt the table and redo log keys.

The encrypted keys are delivered to the Extract process by means of built-in PL/SQL code. Extract uses the shared secret to decrypt the data. Extract never handles the wallet Master Key itself, nor is it aware of the Master Key password. Those remain within the Oracle server security framework.

Extract never writes the decrypted data to any file other than a trail file, not even a discard file (specified with the DISCARDFILE parameter). The word “ENCRYPTED” will be written to any discard file that is in use.

The impact of this feature on Oracle GoldenGate performance should mirror that of the impact of decryption on database performance. Other than a slight increase in Extract startup time, there should be a minimal affect on performance from replicating TDE data.

### Requirements for replicating TDE

- If DDL will ever be performed on a table that has column-level encryption, or if table keys will ever be re-keyed, you must either quiesce the table while the DDL is performed or enable Oracle GoldenGate DDL support. It is more practical to have the DDL environment active so that it is ready, because a re-key usually is a response to a security violation and must be performed immediately. To install the Oracle

GoldenGate DDL environment, see the instructions in this guide. To configure Oracle GoldenGate DDL support, see the *GoldenGate for Windows and UNIX Administrator Guide*. For tablespace-level encryption, the Oracle GoldenGate DDL support is not required.

- To maintain high security standards, the Oracle GoldenGate Extract process should run as part of the Oracle User (the user that runs the Oracle Server). That way, the keys are protected in memory by the same privileges as the Oracle User.
- The Extract process must run on the same machine as the database installation.

## Recommendations for replicating TDE

Extract decrypts the TDE data and writes it to the trail as clear text. To maintain data security throughout the path to the target tables, it is recommended that you also deploy Oracle GoldenGate security features to:

- encrypt the data in the trails
- encrypt the data in transit across TCP/IP

For more information, see the security chapter in the *GoldenGate for Windows and UNIX Administrator Guide*.

## Configuring TDE support

The following outlines the steps that the Oracle Security Officer and the Oracle GoldenGate Administrator take to establish communication between the Oracle server and the Extract process.

### **Oracle Security Officer and Oracle GoldenGate Administrator**

Agree on a shared secret (password) that meets or exceeds Oracle password standards. This password must not be known by anyone else. For more information, see the *Oracle Database Security Guide*.

### **Oracle Security Officer**

1. Issue the following MKSTORE command to create an "ORACLEGG" entry in the Oracle wallet. ORACLEGG must be the name of the key. Do not supply the shared secret on the command line; instead, supply it when prompted.

```
mkstore -wrl ./ -createEntry ORACLE.SECURITY.CL.ENCRYPTION.ORACLEGG
Enter <secret>
```

2. (Oracle RAC 10gR2 and 11gR1) Copy the wallet that contains the shared secret to each node, and then reopen the wallets.

**NOTE** With Oracle 11gR2, there is one wallet in a shared location, with synchronized access among all nodes.

3. Compile the dbms\_internal\_ckim.plb package that is installed in the Oracle GoldenGate installation directory.

### Oracle GoldenGate Administrator

1. Grant EXEC privilege on the `dbms_internal_ckm.get_key` procedure to the Extract database user. This procedure facilitates sharing of the encrypted keys between the Oracle Server and Extract.
2. Run GGSCI and issue the `ENCRYPT PASSWORD` command to encrypt the shared secret so that it is obfuscated within the Extract parameter file. *This is a security requirement.* `ENCRYPT PASSWORD` takes the clear-text string as input and provides options for encrypting it with an Oracle GoldenGate-generated default key or a user-defined key that is stored in a secure local `ENCKEYS` file. For instructions, see the security chapter of the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
3. In the Extract parameter file, use the `DBOPTIONS` parameter with the `DECRYPTPASSWORD` option. As input, supply the encrypted shared secret and the Oracle GoldenGate-generated or user-defined decryption key. For syntax options, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.
4. Close, and then open, the Oracle wallet before you start Extract. This step works around issues with caching that can cause an ORA-28360 (security module) error.

**NOTE** Close and then open the wallet whenever a shared secret is created or changed.

### Changing the Oracle shared secret

Use this procedure to change the shared secret that supports Oracle Transparent Data Encryption. You can change the shared secret when needed, but "ORACLEGG" must remain the name of the key.

1. Stop the Extract process.

```
STOP EXTRACT <group>
```

2. Issue the following `MKSTORE` command to modify the "ORACLEGG" entry in the Oracle wallet. Do not supply the new shared secret on the command line; instead, supply it when prompted.

```
mkstore -wrl ./ -modifyEntry ORACLE.SECURITY.CL.ENCRYPTION.ORACLEGG  
Enter <secret>
```

**NOTE** To change the shared secret, the parameter 'modifyEntry' is used instead of "createEntry," because you are modifying an existing wallet entry.

3. Use the `ENCRYPT PASSWORD` command in GGSCI to encrypt the new shared secret. For instructions, see the security chapter of the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
4. Replace the old encrypted shared secret and decryption key with the new ones in the Extract parameter file by modifying the `DECRYPTPASSWORD` option of `DBOPTIONS`. For syntax options, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.
5. Close, and then open, the Oracle wallet before you start Extract. This process works around issues with caching that can cause an ORA-28360 error.
6. Start Extract.

```
START EXTRACT <group>
```

## Ensuring correct handling of Oracle Spatial objects

To replicate georaster tables (tables that contain one or more columns of SDO\_GEOASTER object type), follow these instructions to configure Oracle GoldenGate to process them correctly.

### Mapping the georaster tables

You must create a TABLE statement and a MAP statement for the georaster tables and also for the related raster data tables.

### Sizing the XML memory buffer

Evaluate your spatial data before starting Oracle GoldenGate processes. If the METADATA attribute of the SDO\_GEOASTER data type in any of the values exceeds 1 MB, you must increase the size of the memory buffer that stores the embedded SYS.XMLTYPE attribute of the SDO\_GEOASTER data type. If the data exceeds the buffer size, Extract abends. The size of the XML buffer is controlled by the DBOPTIONS parameter with the XMLBUFSIZE option. For more information about DBOPTIONS, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

### Handling triggers on the georaster tables

Every georaster table has a trigger that affects the raster data table. To ensure the integrity of the target georaster tables, do the following:

- Keep the trigger enabled on both source and target to ensure consistency of the spatial data.
- Use the REPERROR option of the MAP parameter to handle “ORA-01403 No data found” errors.

The error is caused by redundant deletes on the target. When a row in the source georaster table is deleted, the trigger cascades the delete to the raster data table. Both deletes are replicated. The replicated parent delete triggers the cascaded (child) delete on the target. When the replicated child delete arrives, it is redundant and generates the error.

#### To handle redundant deletes with REPERROR

1. Use a REPERROR statement in each MAP statement that contains a raster data table.
2. Use Oracle error 1403 as the SQL error.
3. Use any of the response options as the error handling.

See the following examples for ways you can configure the error handling.

**Example** A sufficient way to handle the errors is simply to use REPERROR with DISCARD to discard the cascaded delete that triggers them. The trigger on the target georaster table performs the delete to the raster data table, so the replicated one is not needed.

```
MAP geo.st_rdt, TARGET geo.st_rdt, REPERROR (-1403, DISCARD);
```

**Example** If you need to keep an audit trail of the error handling, use REPERROR with EXCEPTION to invoke exceptions handling. For this, you create an exceptions table and map the source raster data table twice:

- once to the actual target raster data table (with REPERROR handling the 1403 errors).
- again to the exceptions table, which captures the 1403 error and other relevant information by means of a COLMAP clause.

When using exceptions handling like this, you must use the ALLOWDUPTARGETMAP parameter to keep Replicat from abending on the dual source mapping.

This example provides a Replicat parameter file that contains the required parameters, and it provides a sample script that creates an exceptions table. Note that a macro is used in the parameter file to populate the TARGET and COLMAP portions of the exceptions MAP statements. The required INSERTALLRECORDS and EXCEPTIONSONLY parameters are also included in the macro. The macro eliminates the need to type the same information over again for each of the MAP statements.

### Replicat parameter file

```

REPLICAT rgeoras
SETENV (ORACLE_SID=tgt111)
USERID geo, PASSWORD xxxxx, ENCRYPTKEY DEFAULT

ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rgeoras.dsc, purge
ALLOWDUPTARGETMAP

-- This starts the macro
MACRO #exception_handling
BEGIN
, TARGET geo.exceptions
, COLMAP ( rep_id = "1"
, table_name = @GETENV ("GGHEADER", "TABLENAME")
, errno = @GETENV ("LASTERR", "DBERRNUM")
, dberrmsg = @GETENV ("LASTERR", "DBERRMSG")
, optype = @GETENV ("LASTERR", "OPTYPE")
, errtype = @GETENV ("LASTERR", "ERRTYPE")
, logrba = @GETENV ("GGHEADER", "LOGRBA")
, logposition = @GETENV ("GGHEADER", "LOGPOSITION")
, committimestamp = @GETENV ("GGHEADER", "COMMITTIMESTAMP")
)
, INSERTALLRECORDS
, EXCEPTIONSONLY ;
END;
-- This ends the macro

EXTTRAIL ./dirdat/eg

-- Mapping of regular and georaster tables. Requires no exception handling.
-- Replicat abends on errors, which is its default error handling.
MAP geo.blob_table, TARGET geo.blob_table ;
MAP geo.georaster_table, TARGET geo.georaster_table ;
MAP geo.georaster_table2, TARGET geo.georaster_table2 ;
MAP geo.georaster_tab1, TARGET geo.georaster_tab1 ;
MAP geo.georaster_tab2, TARGET geo.georaster_tab2 ;
    
```

```
MAP geo.mv_georaster_table1, TARGET geo.mv_georaster_table1 ;

-- Mapping of raster data tables. Requires exception handling for 1403 errors.
MAP geo.st_rdt_3_table, TARGET geo.st_rdt_3_table, REPERROR (-1403, EXCEPTION)
;
MAP geo.st_rdt_3_table #exception_handling()
MAP geo.rdt_1_table, TARGET geo.rdt_1_table, REPERROR (-1403, EXCEPTION) ;
MAP geo.rdt_1_table #exception_handling()
MAP geo.rdt_2_table, TARGET geo.rdt_2_table, REPERROR (-1403, EXCEPTION) ;
MAP geo.rdt_2_table #exception_handling()
MAP geo.mv_rdt_1_table, TARGET geo.mv_rdt_1_table, REPERROR (-1403, EXCEPTION)
;
MAP geo.mv_rdt_1_table #exception_handling()
```

**Sample script that creates an exceptions table**

```
drop table exceptions
/

create table exceptions
( rep_id number
, table_name varchar2(61)
, errno number
, dberrmsg varchar2(4000)
, optype varchar2(20)
, errtype varchar2(20)
, logrba number
, logposition number
, committimestamp timestamp
)
/
```

**NOTE** When using an exceptions table for numerous tables, someone should monitor its growth.

For more information about using an exceptions table and using macros, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

For more information about MAP options, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

## Replicating TIMESTAMP with TIME ZONE

Oracle GoldenGate supports the capture and replication of TIMESTAMP WITH TIME ZONE as a UTC offset (TIMESTAMP '2011-01-01 8:00:00 -8:00') but abends on TIMESTAMP WITH TIME ZONE as TZR (TIMESTAMP '2011-01-01 8:00:00 US/Pacific') by default.

To support TIMESTAMP WITH TIME ZONE as TZR, use the Extract parameter TRANLOGOPTIONS with one of the following:

- INCLUDEREGIONID to replicate TIMESTAMP WITH TIME ZONE as TZR from an Oracle source to an Oracle target of the same version or later.

- INCLUDEREGIONIDWITHOFFSET to replicate TIMESTAMP WITH TIMEZONE as TZR from an Oracle source that is at least v10g to an earlier Oracle target, or to a non-Oracle target.

These options allow replication to Oracle versions that do not support TIMESTAMP WITH TIMEZONE as TZR and to database systems that only support time zone as a UTC offset. For more information, see TRANLOGOPTIONS in the *Windows and UNIX Reference Guide*.

**NOTE** Oracle GoldenGate does not support TIMESTAMP WITH TIMEZONE as TZR for initial loads, situations where the column must be fetched from the database, or for the SQLEXEC feature. In these cases, the region ID is converted to a time offset by the Oracle database when the column is selected. Replicat replicates the column data as date and time data with a time offset value.

## Controlling Replicat COMMIT options on an Oracle target

When an online Replicat group is configured to use a checkpoint table (recommended), it takes advantage of the asynchronous COMMIT feature that was introduced in Oracle 10gR2. When applying a transaction to the Oracle target, Replicat includes the NOWAIT option in the COMMIT statement. This improves performance by allowing Replicat to continue processing immediately after applying the COMMIT, while the database engine logs the transaction in the background.

The checkpoint table supports data consistency with asynchronous COMMITs because it makes the Replicat checkpoint part of the Replicat transaction itself. The checkpoint either succeeds with, or fails with, that transaction. Asynchronous COMMIT is also the default for initial loads and batch processing.

You can disable the default asynchronous COMMIT behavior by using the DBOPTIONS parameter with the DISABLECOMMITNOWAIT option in the Replicat parameter file.

If a checkpoint table is not used for a Replicat group, the checkpoints are maintained in a file on disk, and Replicat uses the synchronous COMMIT option by default (COMMIT with WAIT), which forces Replicat to wait until the transaction is logged before it can continue processing. This prevents inconsistencies that can result after a database failure, where the state of the transaction that is recorded in the checkpoint file might be different than its state after the recovery.

**NOTE** For more information about the different processing methods that are supported by Oracle GoldenGate, and about the checkpoint table, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

## Supporting delivery to Oracle Exadata with EHCC compressed data

Oracle GoldenGate supports delivery to Oracle Exadata with Hybrid Columnar Compression (EHCC) enabled for insert operations. To ensure that this data is applied correctly, use the INSERTAPPEND parameter in the Replicat parameter file. INSERTAPPEND causes Replicat to use an APPEND hint for inserts. Without this hint, the record will be inserted uncompressed. For more information about INSERTAPPEND, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

**NOTE** Capture from Exadata is not supported at this time.

## Managing LOB caching on a target Oracle database

Replicat writes LOB data to the target database in fragments. To minimize the effect of this I/O on the system, Replicat enables Oracle's LOB caching mechanism, caches the fragments in a buffer, and performs a write only when the buffer is full. For example, if the buffer is 25,000 bytes in size, Replicat only performs I/O four times given a LOB of 100,000 bytes.

- To optimize the buffer size to the size of your LOB data, use the DBOPTIONS parameter with the LOBWRITESIZE <size> option. The higher the value, the fewer the I/O calls made by Replicat to write one LOB.
- To disable Oracle's LOB caching, use the DBOPTIONS parameter with the DISABLELOBCACHING option. When LOB caching is disabled, whatever is sent by Replicat to Oracle in one I/O call is written directly to the database media.

## Additional requirements for Oracle RAC

This topic covers additional configuration requirements that apply when Oracle GoldenGate will be operating in an Oracle Real Application Clusters (RAC) environment.

### General requirements

- All nodes in the RAC cluster must 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](http://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*.
- All nodes in the cluster must have the same COMPATIBLE parameter setting.

The following table shows some Oracle GoldenGate parameters that are of specific benefit in Oracle RAC.

**Table 11 Oracle GoldenGate parameters for Oracle RAC**

| Parameter                                                             | Description                                                                                                                                                             |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| THREADOPTIONS with the BINDCPU <n> option                             | Use for Extract on AIX and Solaris machines to handle thread-safety issues related to memory that is updated by different processors.                                   |
| THREADOPTIONS parameter with the INQUEUESIZE and OUTQUEUESIZE options | Sets the amount of data that Extract queues in memory before sending it to the target system. Tuning these parameters might increase Extract performance on Oracle RAC. |



**Table 11 Oracle GoldenGate parameters for Oracle RAC**

| Parameter                                                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TRANLOGOPTIONS parameter with the PURGEORPHANEDTRANSACTIONS   NOPURGEORPHANEDTRANSACTIONS and TRANSCLEANUPFREQUENCY options | Controls how Extract handles orphaned transactions, which can occur when a node fails during a transaction and Extract cannot capture the rollback. Although the database performs the rollback on the failover node, the transaction would otherwise remain in the Extract transaction list indefinitely and prevent further checkpointing for the Extract thread that was processing the transaction. By default, Oracle GoldenGate purges these transactions from its list after they are confirmed as orphaned. This functionality can also be controlled on demand with the SEND EXTRACT command in GGSCI. |

### Special procedures on RAC

- If the primary database instance against which Oracle GoldenGate is running stops or fails for any reason, Extract abends. To resume processing, you can restart the instance or mount the Oracle GoldenGate binaries to another node where the database is running and then restart the Oracle GoldenGate processes. Stop the Manager process on the original node before starting Oracle GoldenGate processes from another node.
- Whenever the number of redo threads changes, the Extract group must be dropped and re-created. For the recommended procedure, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
- To write SQL operations to the trail, Extract must verify that there are no other operations from other RAC nodes that precede the ones in the redo log that it is reading. For example, if a log contains operations that were performed from 1:00 a.m. to 2:00 a.m., and the log from Node 2 contains operations that were performed from 1:30 a.m. to 2:30 a.m., then only the operations up to, and including, the 2:00 a.m. one can be moved to the server where the main Extract is coordinating the redo data. Extract must ensure that there are no more operations between 2:00 a.m. and 2:30 a.m. that need to be captured.
- In active-passive environments, the preceding requirement means that you might need to perform some operations and archive log switching on the passive node to ensure that operations from the active node are passed to the passive node. This eliminates any issues that could arise from a slow archiver process, failed network links, and other latency issues caused by moving archive logs from the Oracle nodes to the server where the main Extract is coordinating the redo data.
- To process the last transaction in a RAC cluster before shutting down Extract, insert a dummy record into a source table that Oracle GoldenGate is replicating, and then switch log files on all nodes. This updates the Extract checkpoint and confirms that all available archive logs can be read. It also confirms that all transactions in those archive logs are captured and written to the trail in the correct order.

## Additional requirements for ASM

This topic covers additional configuration requirements that apply when Oracle GoldenGate operates in an Oracle Automatic Storage Management (ASM) instance.

### Ensuring ASM connectivity

To ensure that the Oracle GoldenGate Extract process can connect to an ASM instance, do the following.

- List the ASM instance in the `tnsnames.ora` file. The recommended method for connecting to an ASM instance when Oracle GoldenGate is running on the database host machine is to use a bequeath (BEQ) protocol.

**NOTE** A BEQ connection does not work when using a remote Extract configuration. In that case, configure `TNSNAMES` with the TCP/IP protocol.

- If using the TCP/IP protocol, verify that the Oracle listener is listening for new connections to the ASM instance. The `listener.ora` file must contain an entry similar to the following.

```
SID_LIST_LISTENER_ASM =  
  (SID_LIST =  
    (SID_DESC =  
      (GLOBAL_DBNAME = ASM)  
      (ORACLE_HOME = /u01/app/grid)  
      (SID_NAME = +ASM1)  
    )  
  )  
)
```

**NOTE** The BEQ protocol does not require a listener.

### Optimizing the ASM connection

Use the `TRANLOGOPTIONS` parameter with the `DBLOGREADER` option in the Extract parameter file if the ASM instance is one of the following versions:

- Oracle 10.2.0.5 or later 10g R2 versions
- Oracle 11.2.0.2 or later 11g R2 versions

A newer ASM API is available in those releases (but not in Oracle 11g R1 versions) that uses the database server to access the redo and archive logs. When used, this API enables Extract to use a read buffer size of up to 4 MB in size. A larger buffer may improve the performance of Extract when redo rate is high. You can use the `DBLOGREADERBUFSIZE` option of `TRANLOGOPTIONS` to specify a buffer size.

For more information about `TRANLOGOPTIONS`, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

## CHAPTER 5

# Preparing DBFS for active-active propagation with Oracle GoldenGate

.....

This chapter contains steps to configure Oracle GoldenGate to function within an active-active bi-directional or multi-directional environment where Oracle Database File System (DBFS) is in use on both (or all) systems.

## Supported operations and prerequisites

Oracle GoldenGate for DBFS supports the following:

- Supported DDL (like TRUNCATE or ALTER) on DBFS objects except for CREATE statements on the DBFS objects. CREATE on DBFS must be excluded from the configuration, as must any schemas that will hold the created DBFS objects. The reason to exclude CREATES is that the metadata for DBFS must be properly populated in the SYS dictionary tables (which itself is excluded from Oracle GoldenGate capture by default).
- Capture and replication of DML on the tables that underlie the DBFS filesystem.

The procedures that follow assume that Oracle GoldenGate is configured properly to support active-active configuration. This means that it must be:

- Installed according to the instructions in this guide.
- Configured according to the instructions in the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

## Applying the required patch

Apply the Oracle DBFS patch for bug-9651229 on both databases. To determine if the patch is installed, run the following query:

```
connect / as sysdba
select procedure_name
from dba_procedures
where object_name = 'DBMS_DBFS_SFS_ADMIN'
and procedure_name = 'PARTITION_SEQUENCE';
```

The query should return a single row. Anything else indicates that the proper patched version of DBFS is not available on your database.

## Examples used in these procedures

The following procedures assume two systems and configure the environment so that DBFS users on both systems see the same DBFS files, directories, and contents that are

.....

kept in synchronization with Oracle GoldenGate. It is possible to extend these concepts to support three or more peer systems.

## Partitioning the DBFS sequence numbers

DBFS uses an internal sequence-number generator to construct unique names and unique IDs. These steps partition the sequences into distinct ranges to ensure that there are no conflicts across the databases. After this is done, further DBFS operations (both creation of new filesystems and subsequent filesystem operations) can be performed without conflicts of names, primary keys, or IDs during DML propagation.

1. Connect to each database as sysdba.
2. Issue the following query on each database.

```
select last_number
from dba_sequences
where sequence_owner = 'SYS'
      and sequence_name = 'DBFS_SFS_$FSSEQ'
```

3. From this query, choose the maximum value of LAST\_NUMBER across both systems, or pick a high value that is significantly larger than the current value of the sequence on either system.
4. Substitute this value (“maxval” is used here as a placeholder) in both of the following procedures. These procedures logically index each system as myid=0 and myid=1.

### Node 1

```
declare
begin
    dbms_dbfs_sfs_admin.partition_sequence(nodes => 2, myid => 0,
        newstart => :maxval);
commit;
end;
/
```

### Node 2

```
declare
begin
    dbms_dbfs_sfs_admin.partition_sequence( nodes => 2, myid => 1,
        newstart => :maxval);
commit;
end;
/
```

**NOTE** Notice the difference in the value specified for the myid parameter. These are the different index values.

For a multi-way configuration among three or more databases, you could make the following alterations:

- Adjust the maximum value that is set for “maxval” upward appropriately, and use that value on all nodes.

- Vary the value of “myid” in the procedure from 0 for the first node, 1 for the second node, 2 for the third one, and so on.
5. (Recommended) After (and only after) the DBFS sequence generator is partitioned, create a new DBFS filesystem on each system, and use only these filesystems for DML propagation with Oracle GoldenGate. See “Configuring the DBFS filesystem”

**NOTE** DBFS filesystems that were created before the patch for bug-9651229 was applied or before the DBFS sequence number was adjusted can be configured for propagation, but that requires additional steps not described in this document. If you must retain old filesystems, open a service request with Oracle Support.

## Configuring the DBFS filesystem

To replicate DBFS filesystem operations, use a configuration that is similar to the standard bi-directional configuration for DML.

- Use matched pairs of identically structured tables.
- Allow each database to have write privileges to opposite tables in a set, and set the other one in the set to read-only. For example:
  - Node1 writes to local table "t1" and these changes are replicated to t1 on Node2.
  - Node2 writes to local table “t2” and these changes are replicated to t2 on Node1.
  - On Node1, t2 is read-only. On Node2, t1 is read-only.

DBFS filesystems make this kind of table pairing simple because:

- The tables that underlie the DBFS filesystems have the same structure.
- These tables are modified by simple, conventional DML during higher-level filesystem operations.
- The DBFS ContentAPI provides a way of unifying the namespace of the individual DBFS stores by means of mount points that can be qualified as read-write or read-only.

The following steps create two DBFS filesystems (in this case named FS1 and FS2) and set them to be read-write or read, as appropriate.

1. Run the following procedure to create the two filesystems. (Substitute your store names for “FS1” and “FS2.”)

**Example**

```
declare
  dbms_dbfs_sfs.createFilesystem('FS1');
  dbms_dbfs_sfs.createFilesystem('FS2');

  dbms_dbfs_content.registerStore('FS1',
    'posix', 'DBMS_DBFS_SFS');
  dbms_dbfs_content.registerStore('FS2',
    'posix', 'DBMS_DBFS_SFS');
  commit;
end;
```

2. Run the following procedure to give each filesystem the appropriate access rights. (Substitute your store names for “FS1” and “FS2.”)

**Example Node 1**

```
declare
  dbms_dbfs_content.mountStore('FS1', 'local');
  dbms_dbfs_content.mountStore('FS2', 'remote',
    read_only => true);
  commit;
end;
/
```

**Example Node 2**

```
declare
  dbms_dbfs_content.mountStore('FS1', 'remote',
    read_only => true);
  dbms_dbfs_content.mountStore('FS2', 'local');
  commit;
end;
/
```

In this example, note that on Node 1, store "FS1" is read-write and store "FS2" is read-only, while on Node 2 the converse is true: store "FS1" is read-only and store "FS2" is read-write.

Note also that the read-write store is mounted as "local" and the read-only store is mounted as "remote". This provides users on each system with an identical namespace and identical semantics for read and write operations. Local path names can be modified, but remote path names cannot.

## Mapping local and remote peers correctly

The names of the tables that underlie the DBFS filesystems are generated internally and dynamically. Continuing with the preceding example, there are:

- Two nodes (Node 1 and Node 2 in the example).
- Four stores: two on each node (FS1 and FS2 in the example).
- Eight underlying tables: two for each store (a table and a ptable). These tables must be identified, specified in Extract TABLE statements, and mapped in Replicat MAP statements.

1. To identify the table names that back each filesystem, issue the following query. (Substitute your store names for "FS1" and "FS2.")

**Example**

```
select fs.store_name, tb.table_name, tb.ptable_name
from table(dbms_dbfs_sfs.listTables) tb,
     table(dbms_dbfs_sfs.listFilesystems) fs
where  fs.schema_name = tb.schema_name
      and fs.table_name = tb.table_name
      and fs.store_name in ('FS1', 'FS2')
;
```

The output looks like the following examples.

**Example output: Node 1** (Your table names will be different.)

| STORE_NAME | TABLE_NAME    | PTABLE_NAME    |
|------------|---------------|----------------|
| FS1        | SFS\$_FST_100 | SFS\$_FSTP_100 |
| FS2        | SFS\$_FST_118 | SFS\$_FSTP_118 |

**Example output: Node 2** (Your table names will be different.)

| STORE_NAME | TABLE_NAME    | PTABLE_NAME    |
|------------|---------------|----------------|
| FS1        | SFS\$_FST_101 | SFS\$_FSTP_101 |
| FS2        | SFS\$_FST_119 | SFS\$_FSTP_119 |

- Identify the tables that are *locally read-write* to Extract by creating the following TABLE statements in the Extract parameter files. (Substitute your owner and table names.)

**Example: Node 1**

```
TABLE owner.SFS$_FST_100;
TABLE owner.SFS$_FSTP_100;
```

**Example: Node 2**

```
TABLE owner.SFS$_FST_119;
TABLE owner.SFS$_FSTP_119;
```

- Link changes on each remote filesystem to the corresponding local filesystem by creating the following MAP statements in the Replicat parameter files. (Substitute your owner and table names.)

**Example: Node 1**

```
MAP owner.SFS$_FST_119,
   TARGET owner.SFS$_FST_118;
MAP owner.SFS$_FSTP_119,
   TARGET owner.SFS$_FSTP_118;
```

**Example: Node 2**

```
MAP owner.SFS$_FST_100,
   TARGET owner.SFS$_FST_101;
MAP owner.SFS$_FSTP_100,
   TARGET owner.SFS$_FSTP_101;
```

This mapping captures and replicates local read-write “source” tables to remote read-only peer tables:

- Filesystem changes made to FS1 on Node 1 propagate to FS1 on Node 2.
- Filesystem changes made to FS2 on Node 2 propagate to FS2 on Node1.

Changes to the filesystems can be made through the DBFS ContentAPI (package DBMS\_DBFS\_CONTENT) of the database or through dbfs\_client mounts and conventional filesystems tools.

All changes are propagated in both directions.

- A user at the virtual root of the DBFS namespace on each system sees identical content.
- For mutable operations, users use the "/local" sub-directory on each system.
- For read operations, users can use either of the "/local" or "/remote" sub-directories, depending on whether they want to see local or remote content.

## CHAPTER 6

# Configuring the Oracle redo logs



This chapter contains information for configuring the online and archive logs to support data capture and to ensure the continuity and integrity of capture processing.

## Ensuring data availability

When operating in its normal mode, Oracle GoldenGate reads the online logs by default, but will read the archived logs if an online log is not available. Therefore, for best results, enable archive logging. The archives provide a secondary data source should the online logs recycle before Extract is finished with them. The archive logs for open transactions must be retained on the system in case Extract needs to recapture data from them to perform a recovery.

If you cannot enable archive logging, configure the online logs according to the following guidelines to retain enough data for Extract to capture what it needs before the logs recycle. Allow for Extract backlogs caused by network outages and other external factors, as well as long-running transactions.

In a RAC configuration, Extract must have access to the online and archived logs for all nodes in the cluster, including the one where Oracle GoldenGate is installed.

### Log retention requirements per Extract recovery mode

The following summarizes the different recovery modes that Extract might use and their log-retention requirements:

- By default, the Bounded Recovery mode is in effect, and Extract requires access to the logs only as far back as twice the Bounded Recovery interval that is set with the BR parameter. This interval is an integral multiple of the standard Extract checkpoint interval, as controlled by the CHECKPOINTSECS parameter. These two parameters control the Oracle GoldenGate Bounded Recovery feature, which ensures that Extract can recover in-memory captured data after a failure, no matter how old the oldest open transaction was at the time of failure. For more information about this requirement, see the BR parameter documentation in the Oracle GoldenGate *Windows and UNIX Reference Guide*.
- In the unlikely event that the Bounded Recovery mechanism fails when Extract attempts a recovery, Extract reverts to normal recovery mode and must have access to the archived log that contains the beginning of the oldest open transaction in memory at the time of failure and all logs thereafter.

### Log retention options

Depending on the version of Oracle, there are different options for ensuring that the required logs are retained on the system.





### **Oracle Enterprise Edition 10.2 and later**

For these versions, Extract works with Oracle Recovery Manager (RMAN) to retain the logs that Extract needs for recovery. This feature is enabled by default when you add or register an Extract group in GGSCI with ADD EXTRACT (TRANLOG option) or REGISTER EXTRACT.

By default, Extract retains enough logs to perform a Bounded Recovery, but you can configure Extract to retain enough logs through RMAN for a normal recovery by using the TRANLOGOPTIONS parameter with the LOGRETENTION option set to SR. There also is an option to disable the use of RMAN log retention. Review the options of LOGRETENTION in the Oracle GoldenGate *Windows and UNIX Reference Guide* before you configure Extract. If you set LOGRETENTION to DISABLED, see “Determining how much data to retain”.

**NOTE** To support RMAN log retention on Oracle RAC, you must download and install the database patch that is provided in BUGFIX 11879974 before you add the Extract groups.

The RMAN log retention feature creates an underlying (but non-functioning) Oracle Streams Capture process for each Extract group. The name of the Capture is based on the name of the associated Extract group. The log retention feature can operate concurrently with other local Oracle Streams installations. When you create an Extract group, the logs are retained from the current database SCN.

**NOTE** If the Oracle Flashback storage area is full, RMAN purges the archive logs even when needed by Extract. This limitation exists so that the requirements of Extract (and other Oracle replication components) do not interfere with the availability of redo to the database.

### **All other Oracle versions**

For versions of Oracle other than Enterprise Edition 10.2 and later, you must manage the log retention process yourself with your preferred administrative tools. Follow the directions in “Determining how much data to retain”.

## **Determining how much data to retain**

When managing log retention, try to ensure rapid access to the logs that Extract would require to perform a normal recovery (not a Bounded Recovery). See “Log retention requirements per Extract recovery mode”. If you must move the archives off the database system, the TRANLOGOPTIONS parameter provides a way to specify an alternate location. See “Specifying the archive location”.

The recommended retention period is at least 24 hours worth of transaction data, including both online and archived logs. To determine the oldest log that Extract might need at any given point, issue the SEND EXTRACT command with the SHOWTRANS option. You might need to do some testing to determine the best retention time given your data volume and business requirements.

If data that Extract needs during processing was not retained, either in online or archived 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 data, and then start the Oracle GoldenGate environment over again.

## Purging log archives

Make certain not to use backup or archive options that cause old archive files to be overwritten by new backups. Ideally, 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.

## Specifying the archive location

If the archived logs reside somewhere other than the Oracle default directory, specify that directory with the `ALTARCHIVELOGDEST` option of the `TRANLOGOPTIONS` parameter in the Extract parameter file.

You might also need to use the `ALTARCHIVEDLOGFORMAT` option of `TRANLOGOPTIONS` if the format that is specified with the Oracle parameter `LOG_ARCHIVE_FORMAT` contains sub-directories. `ALTARCHIVEDLOGFORMAT` specifies an alternate format that removes the sub-directory from the path. For example, `%T/log_%t_%s%.arc` would be changed to `log_%t_%s%.arc`. As an alternative to using `ALTARCHIVEDLOGFORMAT`, you can create the sub-directory manually, and then move the log files to it.

## Mounting logs that are stored on other platforms

If the online and archived redo logs are stored on a different platform from the one that Extract is built for, do the following:

- NFS-mount the archive files.
- Map the file structure to the structure of the source system by using the `LOGSOURCE` and `PATHMAP` options of the Extract parameter `TRANLOGOPTIONS`. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

## Configuring Oracle GoldenGate to read only the archived logs

You can configure Extract to read exclusively from the archived logs. This is known as *Archived Log Only (ALO)* mode. In this mode, Extract reads exclusively from archived logs that are stored in a specified location. ALO mode enables Extract to use production logs that are shipped to a secondary database (such as a standby) as the data source. The online logs are not used at all. Oracle GoldenGate connects to the secondary database to get metadata and other required data as needed. As an alternative, ALO mode is supported on the production system.

### Limitations and requirements of ALO mode

- Log resets (`RESETLOG`) cannot be done on the source database after the standby database is created.
- ALO cannot be used on a standby database if the production system is Oracle RAC and the standby database is non-RAC. In addition to both systems being Oracle RAC, the number of nodes on each system must be identical.
- ALO on Oracle RAC requires a dedicated connection to the source server. If that connection is lost, Oracle GoldenGate processing will stop.

- On Oracle RAC, the directories that contain the archive logs must have unique names across all nodes; otherwise, Extract may return “out of order SCN” errors.
- ALO mode does not support archive log files in ASM mode. The archive log files must be outside the ASM environment for Extract to read them.

### Configuring Extract for ALO mode

1. Enable supplemental logging at the table level and the database level for the tables in the source database. (See “Configuring the database to log key values” on page 38.)
2. When Oracle GoldenGate is running on a different server from the source database, make certain that SQL\*Net is configured properly to connect to a remote server, such as providing the correct entries in a TNSNAMES file. Extract must have permission to maintain a SQL\*Net connection to the source database.
3. Use a SQL\*Net connect string in:
  - The USERID parameter in the parameter file of every Oracle GoldenGate process that connects to that database.
  - The DBLOGIN command in GGSCI.

Example USERID statement:

```
USERID ggext@ora10g01, PASSWORD ggs123
```

**NOTE** If you have a standby server that is local to the server that Oracle GoldenGate is running on, you do not need to use a connect string in USERID. You can just supply the user login name.

4. Use the Extract parameter TRANLOGOPTIONS with the ARCHIVEDLOGONLY option. This option forces Extract to operate in ALO mode against a primary or logical standby database, as determined by a value of PRIMARY or LOGICAL STANDBY in the db\_role column of the v\$database view. The default is to read the online logs. TRANLOGOPTIONS with ARCHIVEDLOGONLY is not needed if using ALO mode against a physical standby database, as determined by a value of PHYSICAL STANDBY in the db\_role column of v\$database. Extract automatically operates in ALO mode if it detects that the database is a physical standby.
5. Other TRANLOGOPTIONS options might be required for your environment. For example, depending on the copy program that you use, you might need to use the COMPLETEARCHIVEDLOGONLY option to prevent Extract errors.
6. Use the MAP parameter for Extract to map the table names to the source object IDs. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.
7. Add the Extract group by issuing the ADD EXTRACT command with a timestamp as the BEGIN option, or by using ADD EXTRACT with the SEQNO and RBA options. It is best to give Extract a known start point at which to begin extracting data, rather than by using the NOW argument. The start time of “NOW” corresponds to the time of the current *online* redo log, but an ALO Extract cannot read the online logs, so it must wait for that log to be archived when Oracle switches logs. The timing of the switch depends on the size of the redo logs and the volume of database activity, so there might be a lag between when you start Extract and when data starts being captured. This can happen in both regular and RAC database configurations.

**NOTE** If Extract appears to stall while operating in ALO mode, see the Oracle GoldenGate *Windows and UNIX Troubleshooting and Tuning Guide* for help with diagnosing the problem.

## Setting redo parallelism for Oracle 9i sources

If using Oracle GoldenGate for an Oracle 9i source database, set the `_LOG_PARALLELISM` parameter to 1. Oracle GoldenGate does not support values higher than 1.

## Avoiding log-read bottlenecks

When Oracle GoldenGate captures data from the redo logs, I/O bottlenecks can occur because Extract is reading the same files that are being written by the database logging mechanism. Performance degradation increases with the number of Extract processes that read the same logs. You can:

- Try using faster drives and a faster controller. Both Extract and the database logging mechanism will be faster on a faster I/O system.
- Store the logs on RAID 0+1. Avoid RAID 5, which performs checksums on every block written and is not a good choice for high levels of continuous I/O. For more information, see the Oracle documentation or search related web sites.

## CHAPTER 7

# Managing the Oracle DDL replication environment



This chapter contains instructions for making changes to the database environment or the Oracle GoldenGate environment when the Oracle GoldenGate DDL objects still exist on the system.

For instructions on configuring Oracle GoldenGate DDL support, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

## Enabling and disabling the DDL trigger

You can enable and disable the trigger that captures DDL operations without making any configuration changes within Oracle GoldenGate. The following scripts control the DDL trigger.

- `ddl_disable`: Disables the trigger. No further DDL operations are captured or replicated after you disable the trigger.
- `ddl_enable`: Enables the trigger. When you enable the trigger, Oracle GoldenGate starts capturing current DDL changes, but does not capture DDL that was generated while the trigger was disabled.

Before running these scripts, disable all sessions that ever issued DDL, including those of the Oracle GoldenGate processes, SQL\*Plus, business applications, and any other software that uses Oracle. Otherwise the database might generate an ORA-04021 error. Do not use these scripts if you intend to maintain consistent DDL on the source and target systems.

## Maintaining the DDL marker table

You can purge rows from the marker table at any time. It does not keep DDL history. To purge the marker table, use the Manager parameter `PURGEMARKERHISTORY`. Manager gets the name of the marker table from one of the following:

1. The name given with the `MARKERTABLE <table>` parameter in the `GLOBALS` file, if specified.
2. The default name of `GGSMARKER`.

`PURGEMARKERHISTORY` provides options to specify maximum and minimum lengths of time to keep a row, based on the last modification date. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

## Deleting the DDL marker table

Do not delete the DDL marker table unless you want to discontinue synchronizing DDL. The marker table and the DDL trigger are interdependent. An attempt to drop the marker



table fails if the DDL trigger is enabled. This is a safety measure to prevent the trigger from becoming invalid and missing DDL operations. If you remove the marker table, the following error is generated:

```
"ORA-04098: trigger 'SYS.GGS_DDL_TRIGGER_BEFORE' is invalid and failed re-validation"
```

The proper way to remove an Oracle GoldenGate DDL object depends on your plans for the rest of the DDL environment. To choose the correct procedure, see one of the following:

- “Restoring an existing DDL environment to a clean state” on page 66
- “Removing the DDL objects from the system” on page 67

## Maintaining the DDL history table

You can purge the DDL history table to control its size, but do so carefully. The DDL history table maintains the integrity of the DDL synchronization environment. Purges to this table cannot be recovered through the Oracle GoldenGate interface.

### To maintain the DDL history table

1. To prevent any possibility of DDL history loss, make regular full backups of the history table.
2. To ensure the recoverability of purged DDL, enable Oracle Flashback for the history table. Set the flashback retention time well past the point where it could be needed. For example, if your full backups are at most one week old, retain two weeks of flashback. Oracle GoldenGate can be positioned backward into the flashback for reprocessing.
3. If possible, purge the DDL history table manually to ensure that essential rows are not purged accidentally. If you require an automated purging mechanism, use the PURGEDDLHISTORY parameter in the Manager parameter file. You can specify maximum and minimum lengths of time to keep a row. For more information, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

**NOTE** Temporary tables created by Oracle GoldenGate to increase performance might be purged at the same time as the DDL history table, according to the same rules. The names of these tables are derived from the name of the history table, and their purging is reported in the Manager report file. This is normal behavior.

## Deleting the DDL history table

Do not delete the DDL history table unless you want to discontinue synchronizing DDL. The history table contains a record of DDL operations that were issued.

The history table and the DDL trigger are interdependent. An attempt to drop the history table fails if the DDL trigger is enabled. This is a safety measure to prevent the trigger from becoming invalid and missing DDL operations. If you remove the history table, the following error is generated:

```
"ORA-04098: trigger 'SYS.GGS_DDL_TRIGGER_BEFORE' is invalid and failed re-validation"
```

The proper way to remove an Oracle GoldenGate DDL object depends on your plans for the

rest of the DDL environment. To choose the correct procedure, see one of the following:

- “Restoring an existing DDL environment to a clean state” on page 66
- “Removing the DDL objects from the system” on page 67

## Purging the DDL trace file

To prevent the DDL trace file from consuming excessive disk space, run the `ddl_cleartrace` script on a regular basis. This script deletes the trace file, but Oracle GoldenGate will create it again.

The default name of the DDL trace file is `ggs_ddl_trace.log`. It is in the `USER_DUMP_DEST` directory of Oracle. The `ddl_cleartrace` script is in the Oracle GoldenGate directory.

## Applying database patches and upgrades when DDL support is enabled

Database patches and upgrades usually invalidate the Oracle GoldenGate DDL trigger and other Oracle GoldenGate DDL objects. Before applying a database patch, do the following.

1. Disable the Oracle GoldenGate DDL trigger by running the following script.

```
@ddl_disable
```

2. Apply the patch.

3. Enable the DDL trigger by running the following script.

```
@ddl_enable
```

**NOTE** Database upgrades and patches generally operate on Oracle objects. Because Oracle GoldenGate filters out those objects automatically, DDL from those procedures is not replicated when replication starts again.

To avoid recompile errors after the patch or upgrade, which are caused if the trigger is not disabled before the procedure, consider adding calls to `@ddl_disable` and `@ddl_enable` at the appropriate locations within your scripts.

## Applying Oracle GoldenGate patches and upgrades when DDL support is enabled

**NOTE** If there are instructions like these in the release notes or upgrade instructions that accompany a release, follow those instead of these. Do not use this procedure for an upgrade from an Oracle GoldenGate version that does not support DDL statements that are larger than 30K (pre-version 10.4). To upgrade in that case, follow the instructions in “Restoring an existing DDL environment to a clean state” on page 66.

Follow these steps to apply a patch or upgrade to the DDL objects. This procedure may or may not preserve the current DDL synchronization configuration, depending on whether the new build requires a clean installation.

1. Run `GGSCI`. Keep the session open for the duration of this procedure.

2. Stop Extract to stop DDL capture.  
`STOP EXTRACT <group>`
3. Stop Replicat to stop DDL replication.  
`STOP REPLICAT <group>`
4. Download or extract the patch or upgrade files according to the instructions provided by Oracle GoldenGate.
5. Change directories to the Oracle GoldenGate installation directory.
6. Run SQL\*Plus and log in as a user that has SYSDBA privileges.
7. Disconnect all sessions that ever issued DDL, including those of Oracle GoldenGate processes, SQL\*Plus, business applications, and any other software that uses Oracle. Otherwise the database might generate an ORA-04021 error.
8. Run the `ddl_disable` script to disable the DDL trigger.
9. Run the `ddl_setup` script. You are prompted for:
  - The name of the Oracle GoldenGate DDL schema. If you changed the schema name, use the new one.
  - The installation mode: Select either NORMAL or INITIALSETUP mode, depending on what the installation or upgrade instructions require. NORMAL mode recompiles the DDL environment without removing DDL history. INITIALSETUP removes the DDL history.
10. Run the `ddl_enable.sql` script to enable the DDL trigger.
11. In GGSCI, start Extract to resume DDL capture.  
`START EXTRACT <group>`
12. Start Replicat to start DDL replication.  
`START REPLICAT <group>`

## Restoring an existing DDL environment to a clean state

Follow these steps to completely remove, and then reinstall, the Oracle GoldenGate DDL objects. This procedure creates a new DDL environment, but removes DDL history.

**NOTE** Due to object interdependencies, all objects must be removed and reinstalled in this procedure.

1. If you are performing this procedure in conjunction with the installation of a new Oracle GoldenGate version, download and install the Oracle GoldenGate files, and create or update process groups and parameter files as necessary.
2. (Optional) To preserve the continuity of source and target structures, stop DDL activities and then make certain that Replicat finished processing all of the DDL and DML data in the trail. To determine when Replicat is finished, issue the following command until you see a message that there is no more data to process.

```
INFO REPLICAT <group>
```



**NOTE** Instead of using INFO Replicat, you can use the EVENTACTIONS option of TABLE and MAP to stop the Extract and Replicat processes after the DDL and DML has been processed.

3. Run GGSCI.
4. Stop Extract to stop DDL capture.  
`STOP EXTRACT <group>`
5. Stop Replicat to stop DDL replication.  
`STOP REPLICAT <group>`
6. Change directories to the Oracle GoldenGate installation directory.
7. Run SQL\*Plus and log in as a user that has SYSDBA privileges.
8. Disconnect all sessions that ever issued DDL, including those of Oracle GoldenGate processes, SQL\*Plus, business applications, and any other software that uses Oracle. Otherwise the database might generate an ORA-04021 error.
9. Run the `ddl_disable` script to disable the DDL trigger.
10. Run the `ddl_remove` script to remove the Oracle GoldenGate DDL trigger, the DDL history and marker tables, and other associated objects. This script produces a `ddl_remove_spool.txt` file that logs the script output and a `ddl_remove_set.txt` file that logs environment settings in case they are needed for debugging.
11. Run the `marker_remove` script to remove the Oracle GoldenGate marker support system. This script produces a `marker_remove_spool.txt` file that logs the script output and a `marker_remove_set.txt` file that logs environment settings in case they are needed for debugging.
12. Run the `marker_setup` script to reinstall the Oracle GoldenGate marker support system. You are prompted for the name of the Oracle GoldenGate schema.
13. Run the `ddl_setup` script. You are prompted for:
  - The name of the Oracle GoldenGate DDL schema.
  - The installation mode. To reinstall DDL objects, use the `INITIALSETUP` mode. This mode drops and recreates existing DDL objects before creating new objects.
14. Run the `role_setup` script to recreate the Oracle GoldenGate DDL role.
15. Grant the role to all Oracle GoldenGate users under which the following Oracle GoldenGate processes run: Extract, Replicat, GGSCI, and Manager. You might need to make multiple grants if the processes have different user names.
16. Run the `ddl_enable.sql` script to enable the DDL trigger.

## Removing the DDL objects from the system

This procedure removes the DDL environment and removes the history that maintains

continuity between source and target DDL operations.

**NOTE** Due to object interdependencies, all objects must be removed.

1. Run GGSCI.
2. Stop Extract to stop DDL capture.  
`STOP EXTRACT <group>`
3. Stop Replicat to stop DDL replication.  
`STOP REPLICAT <group>`
4. Change directories to the Oracle GoldenGate installation directory.
5. Run SQL\*Plus and log in as a user that has SYSDBA privileges.
6. Disconnect all sessions that ever issued DDL, including those of Oracle GoldenGate processes, SQL\*Plus, business applications, and any other software that uses Oracle. Otherwise the database might generate an ORA-04021 error.
7. Run the `ddl_disable` script to disable the DDL trigger.
8. Run the `ddl_remove` script to remove the Oracle GoldenGate DDL trigger, the DDL history and marker tables, and the associated objects. This script produces a `ddl_remove_spool.txt` file that logs the script output and a `ddl_remove_set.txt` file that logs current user environment settings in case they are needed for debugging.
9. Run the `marker_remove` script to remove the Oracle GoldenGate marker support system. This script produces a `marker_remove_spool.txt` file that logs the script output and a `marker_remove_set.txt` file that logs environment settings in case they are needed for debugging.

## CHAPTER 8

# 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 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.
7. Exit GGSCI.
8. Remove the Oracle GoldenGate files by removing the installation directory.
9. 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. From the Oracle GoldenGate installation folder, run GGSCI.
3. Stop all Oracle GoldenGate processes.
4. Stop the Manager program or service.
5. Exit GGSCI.
6. Click **Start > Run**, and type cmd in the **Run** dialog box to open the command console.
7. Change directories to the Oracle GoldenGate installation directory.
8. 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.

.....

9. Delete the CATEGORY.DLL and GGSMSG.DLL files from the Windows SYSTEM32 folder.
10. Delete the Oracle GoldenGate installation folder.
11. Drop any Oracle GoldenGate-related objects from the database as needed.

## Uninstalling 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. Delete the Oracle GoldenGate installation folder.
10. 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 12 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. |
| emscnt       | 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.                    |
| ggmxcinstall | Oracle GoldenGate installation script for the SQL/MX database.                                                                         |
| ggsci        | User interface to Oracle GoldenGate for issuing commands and managing parameter files.                                                 |



**Table 12 Programs and utilities (continued)**

| Program                                                    | Description                                                                                                                                                               |
|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ggsmgr.jcl<br>ggsmgr.proc<br>ggsmgrst.jcl<br>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 13 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. |

**Table 13 Subdirectories (continued)**

| Directory | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 &lt;group name&gt;&lt;sequence number&gt;.&lt;ext&gt; where &lt;sequence number&gt; is a sequential number appended to aged files and &lt;ext&gt; is either cpe for Extract checkpoint files or cpr for Replicat checkpoint files.</p> <p>Do not edit these files.</p> <p>Examples:<br/>ext1.cpe<br/>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:<br/>rt000001<br/>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:<br/>defs.dat</p>                                                        |
| dirjar    | <p>Contains the Java executable files that support Oracle GoldenGate Monitor.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| dirout    | <p>This directory is not used any more.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

**Table 13 Subdirectories (continued)**

| Directory | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dirpcs    | <p>Default location for status files. File name format is &lt;group&gt;.&lt;extension&gt; where &lt;group&gt; is the name of the group and &lt;extension&gt; 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:<br/>mgr.pcm<br/>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 &lt;group name/user-defined name&gt;.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:<br/>defgen.prm<br/>finance.prm</p> |
| dirrec    | <p>Not used by Oracle GoldenGate.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 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 &lt;group name&gt;&lt;sequence number&gt;.rpt where &lt;sequence number&gt; is a sequential number appended to aged files.</p> <p>Do not edit these files.</p> <p>Examples:<br/>fin2.rpt<br/>mgr4.rpt</p>                                                                                                            |
| dirsql    | <p>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.</p>                                                                                                                                                                                                                                                                                                                                                       |
| dirtmp    | <p>The default location for storing large transactions when the size exceeds the allocated memory size. Do not edit these files.</p>                                                                                                                                                                                                                                                                                                                                                                                                                   |
| dirver    | <p>An Oracle GoldenGate Veridata directory. Not used unless this software is installed in the Oracle GoldenGate location.</p>                                                                                                                                                                                                                                                                                                                                                                                                                          |



**Table 13 Subdirectories (continued)**

| Directory        | Description                                                                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 14 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.                                                                |

**Table 14 Other files (continued)**

| Component                                                               | Description                                                                                                             |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 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<br>ddl_ora10.sql<br>ddl_ora11.sql<br>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<br>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)               |

**Table 14 Other files (continued)**

| Component                                                                                                                                                | Description                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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<br>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<br>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<br>demo_more_<db>_create.sql<br>demo_<db>_insert.sql<br>demo_more_<db>_insert.sql<br>demo_<db>_lob_create.sql<br>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.                                                                                                                                                                                                                                                  |
| 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. |

**Table 14 Other files (continued)**

| Component                                 | Description                                                                                           |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------|
| 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<br>icuin38.dll<br>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<br>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.                                                                 |

**Table 14 Other files (continued)**

| Component                             | Description                                                                                               |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------|
| pthread-VC.dll                        | POSIX threads library for Microsoft Windows.                                                              |
| prvtclkm.plb                          | Supports the replication of Oracle encrypted data.                                                        |
| pw_agent_util.bat<br>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<br>stop.prm                 | z/OS paramlib members to start and stop the Manager process.                                              |
| startmgr<br>stopmgr                   | z/OS Unix System Services scripts to start the Manager process from GGSCI.                                |
| startmgrcom<br>stopmgrcom             | z/OS system input command for the Manager process.                                                        |
| tcperr                                | 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 15 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. |

**Table 15 Checkpoint table definitions**

| Column                  | Description                                                                                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 26  
**\$PATH** variable 25

## A

**ADD SCHEMATRANDATA** command 38  
**ADD TRANDATA** command 38  
**ADDEVENTS** Windows service option 28  
**ADDSERVICE** Windows service option 28  
**ALTARCHIVEDLOGFORMAT** option, **TRANLOGOPTIONS** 60  
**ALTARCHIVELOGDEST** option, **TRANLOGOPTIONS** 60  
**ANYDATA** data type 13  
**ANYDATASET** data type 13  
**ANYTYPE** data type 13  
archive logs, configuring 58  
**ASMPASSWORD** option, **TRANLOGOPTIONS** 6  
**ASMUSER** option, **TRANLOGOPTIONS** 6  
asynchronous **COMMIT** feature 49  
**Automatic Storage Management (ASM)** 7, 52  
**AUTOSTART** Windows service option 29

## B

**BASICFILE LOB** 10  
**BFILE** data type 13  
binary data types 9  
**BINARY DOUBLE** data type 8  
**BINARY FLOAT** data type 8  
**BINARY\_INTEGER** data type 13  
**BINDCPU** option, **THREDOPTIONS** 50  
**BLOB** data type 10

## C

cascade deletes, disabling 36  
cascade updates, disabling 36  
case, supported 19

**category.dll** 28  
**CHAR** data type 8  
character data types 8  
character sets, configuring 40  
characters  
    multibyte 9  
    supported in object names 19  
**checkpoint table** 49  
**CLOB** data type 10  
clocks, synchronizing 50  
cluster  
    installing on 2, 26, 27, 29  
    removing from 70  
**clustered database, installing on** 3  
**clustered tables** 16  
**code points of character set, representing** 9  
**collection types, support for** 12  
**COLS** option, **ADD TRANDATA** 39  
columns  
    number and size supported 13  
    unused 14  
    virtual 14  
**COMMIT, asynchronous (NOWAIT)** 49  
**components, Oracle GoldenGate** 71  
**COMPRESS** option of **CREATE** 17  
**compression, table** 15  
**connections, to database** 6  
**constraints, preparing for processing** 36  
**CREATE SUBDIRS** command 26  
**cursors, open** 41

## D

**database**  
    character sets 40  
    multiple instances 5  
    preparing for Oracle GoldenGate 36  
    versions supported 1



**database filesystem, replicating** 53  
**Database Replay** 17  
**date and time data types** 9  
**DATE data type** 9  
**DBFS, replicating** 53  
**DBLOGIN command** 39  
**dbms\_internal\_cklm.get\_key procedure** 45  
**dbms\_internal\_cklm.plb package** 44  
**DDL**  
     installing support for 32  
     managing replication environment 63  
     nonsupported objects and operations 18  
     supported objects and operations 17  
**ddl\_cleartrace script** 65  
**ddl\_disable script** 63  
**ddl\_enable script** 63  
**ddl\_pin script** 35  
**ddl\_remove\_files** 68  
**ddl\_remove script** 67, 68  
**ddl\_remove\_spool** 67  
**ddl\_setup script** 34, 66  
**DECRYPTPASSWORD option, DBOPTIONS** 45  
**deleteevents** 69  
**deletes, cascaded** 36  
**deleteservice** 69  
**deleting**  
     DDL history table 64  
     DDL marker table 63  
     DDL objects from system 67  
     DDL trace file 65  
**DICOM, support for** 13  
**direct load** 15  
**direct-load INSERTs** 15  
**DISABLECOMMITNOWAIT option, DBOPTIONS** 49  
**disk requirements** 2  
**E**  
**EHCC compressed data** 49

**ENCRYPT PASSWORD command** 45  
**encrypted data, replicating** 43  
**environment variables, setting** 23, 24  
**escape sequence for Unicode data** 9  
**Exadata Hybrid Columnar Compression** 49  
**EXTERNAL attribute of tables** 15

## F

**fetch options, setting** 41  
**FETCHOPTIONS parameter** 42  
**files, installed by Oracle GoldenGate** 71  
**firewall, configuring** 4  
**fixed-point numbers** 8  
**Flashback Query, using** 41  
**floating-point numbers** 8

## G

**GEORASTER, capturing** 46  
**ggmessage.dat file** 77  
**GGs\_tables for DDL support** 32  
**ggs\_ddl\_trace log** 65  
**GGSMGR default Manager name** 27  
**ggsmsg.dll** 28  
**globalization settings, viewing and configuring** 40  
**GLOBALS file** 28

## H

**HANDLETPKUPDATE parameter** 37

## I

**I/O on redo logs** 62  
**INQUEUESIZE option, THREADOPTIONS** 50  
**installing on**  
     Linux and UNIX 26  
     Windows 27  
**INTERVAL data type** 13  
**interval partitioning** 14



**Itanium preinstallation requirements** 4

## K

### key

- absence of 40
- assigning 37
- including in redo record 38
- name, supported characters 19
- transient updates 36

**KEYCOLS option, TABLE or MAP** 38

## L

**large objects, limitations on** 11

**LIBPATH variable** 26

**libraries, Visual C++** 5

**LIMITROWS option, DBOPTIONS** 40

**Linux, installing on** 26

### LOB data types

- retention, setting 42
- support for 10

**log retention, calculating** 58

**LOG\_ARCHIVE\_FORMAT parameter** 60

**LOG\_PARALLELISM parameter** 62

**logs, configuring** 58

**LOGSOURCE option, TRANLOGOPTIONS** 60

**LONG data types** 8, 9

## M

### Manager

- as Windows service 28
- multiple on same system 27
- name, customizing 27

**MANUALSTART Windows service option** 29

**marker\_remove script** 67, 68

**marker\_remove\_spool file** 67, 68

**marker\_setup script** 34

**materialized views** 15

**MAXFETCHSTATEMENTS parameter** 41, 42

**Memory requirements for Oracle GoldenGate** 1

**MGRSERVNAME parameter** 28

**Microsoft Visual C++ 2005 SP1 Redistributable Package** 5

**MKSTORE command** 44

**MLSLABEL data type** 13

**multi-byte data types** 9

**multimedia types, support for** 13

**multiple database instances** 5

## N

### name

- non-supported characters in 21
- supported characters in 19

**names, supported** 19

**NCHAR data type** 9

**NCLOB data type** 10

**nested table** 12

**network configuration** 4

**NLS\_LANG parameter** 40

**NLS\_NCHAR\_CHARACTERSET parameter** 9

**NOKEY option, ADD TRANDATA** 39

**NOWAIT in COMMIT statement** 49

**NUMBER data type** 8

**numeric data types** 8

**NVARCHAR2 data type** 9

## O

**object tables, support for** 12

**objects, supported** 13

**operating systems supported** 1

**operations, supported** 13

### Oracle GoldenGate

- installed programs and files 71
- installing 22
- uninstalling 69

**Oracle, versions supported** 1

**ORACLEGG entry in wallet** 44

**ORACLE\_HOME and ORACLE\_SID settings** 23

**ORDDicom type, support for** 13

**OUTQUEUESIZE option, THREADOPTIONS** 50

## P

**parallelism, redo logs** 62

**partitioning, interval** 14

**PASSWORD Windows service option** 29  
**PATHMAP option, TRANLOGOPTIONS** 60  
**PCTVERSION parameter** 42  
**platforms, supported** 1  
**PLS\_INTEGER data type** 13  
**ports, required by Oracle GoldenGate** 4  
**pre-installation instructions** 1  
**privileges**  
     database 6  
     operating system 4  
**PURGEDDLHISTORY parameter** 64  
**PURGEMARKERHISTORY parameter** 63  
**PURGEORPHANEDTRANSACTIONS option, TRANLOGOPTIONS** 51

## Q

**queries**  
     on undo tablespace 41  
     prepared, number of 42

## R

**RAID drives** 62  
**RAW data type** 9  
**Real Application Cluster** 3  
**recyclebin, Oracle** 18  
**redo logs, configuring** 58  
**REFs** 17  
**removing**  
     DDL objects 67  
     Oracle GoldenGate 69  
**REPFETCHEDCOLOPTIONS parameter** 43  
**REPORTFETCH option, STATOPTIONS** 42  
**role\_setup script** 34  
**ROWID data type** 13  
**rows, number and size supported** 13

## S

**SDO\_GEOMETRY type** 13  
**SDO\_GEORASTER type** 13  
**SDO\_TOPO\_GEOMETRY type** 13  
**SECUREFILE LOB** 10, 11

**semantics, considerations for** 9  
**sequences, supported** 16  
**SETENV parameter** 23  
**shared secret** 43, 45  
**SHLIB\_PATH variable** 26  
**snapshot, using for fetches** 41  
**spaces**  
     in folder names 27  
     in object and column names 21  
**Spatial types**  
     capturing 46  
     support for 13  
**SQL statements, prepared** 41  
**subdirectories, creating** 26  
**synonyms** 17  
**system requirements** 1

## T

**table keys, sharing** 43  
**tables**  
     clustered 16  
     created as EXTERNAL 15  
     supported types 13  
**TCP/IP, configuring** 4  
**THREDOPTIONS with BINDCPU** 50  
**TIMESTAMP data type** 9  
**TIMESTAMP with TIME\_ZONE** 48  
**TIMEZONE\_ABBR** 13  
**transaction logs, configuring** 58  
**TRANSCLEANUPFREQUENCY option, TRANLOGOPTIONS** 51  
**transient primary key update** 36  
**Transparent Data Encryption (TDE)** 43  
**triggers, disabling on target** 36

## U

**undo tablespace, Oracle** 41  
**UNDO\_MANAGEMENT parameter** 41  
**UNDO\_RETENTION parameter** 41  
**uninstalling Oracle GoldenGate** 69  
**UNIX, installing on** 26  
**URITYPE data type** 13

**UROWID data type** 13  
**user defined type** 12  
**USER Windows service option** 29  
**user, Oracle GoldenGate** 6  
**USERID parameter** 6

## **V**

**VAMSERV program** 72  
**VARCHAR2 data type** 8  
**VARRAY data type** 13  
**VARWIDTHNCHAR parameter** 9  
**vcredist\_IA64.exe runtime library** 4  
**versions of Oracle supported** 1

**views** 15  
**virtual columns** 14  
**virtual machine, support for** 5  
**Visual C ++ 2005 SP1 Redistributable Package** 5

## **W**

**wallet, Oracle** 43  
**Windows, installing on** 27

## **X**

**XA distributed transactions** 17  
**XMLType** 11