

Oracle® GoldenGate
Oracle Installation and Setup Guide
11*g* Release 1 (11.1.1)
E17799-01

August 2010

ORACLE®

Copyright © 1995, 2010 Oracle and/or its affiliates. All rights reserved.

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

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

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

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

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

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

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

.....

Chapter 1	System requirements and preinstallation instructions	4
	Overview of Oracle GoldenGate for Oracle.....	4
	Supported Platforms.....	4
	Operating system requirements.....	4
	Database requirements.....	8
	Supported data types.....	10
	Numeric data types.....	10
	Character data types.....	11
	Multi-byte character types.....	11
	Binary data types.....	11
	Date and timestamp data types.....	11
	Large object data types.....	11
	XML data types.....	12
	User defined types.....	13
	Other supported data types.....	14
	Non-supported data types.....	14
	Supported objects and operations for DML.....	14
	Tables, views, and materialized views.....	14
	Sequences.....	17
	Supported objects and operations for DDL.....	18
	Non-supported objects and operations for DML.....	18
	Non-supported objects and operations for DDL.....	18
	Oracle-reserved schemas.....	18
	Oracle recycle bin.....	19
	Supported and non-supported object names and case.....	19
	Object names and owners.....	19
	Case sensitivity.....	19
	Supported characters.....	20
	Non-supported characters.....	21

.....

Chapter 2	Installing Oracle GoldenGate	22
	Installation overview	22
	Upgrades	22
	New installations	22
	Downloading Oracle GoldenGate	22
	Setting ORACLE_HOME and ORACLE_SID	23
	Setting library paths for dynamic builds on UNIX systems	24
	Installing Oracle GoldenGate on Linux and UNIX	26
	Installing into a UNIX or Linux cluster	26
	Installing the Oracle GoldenGate files	26
	Configuring Manager and other processes	27
	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
	Configuring Manager and other processes	30
Chapter 3	Installing Oracle GoldenGate DDL support for an Oracle database	31
	Overview of DDL support	31
	Oracle GoldenGate DDL objects in an Oracle standby configuration	32
	Installing the DDL objects	33
Chapter 4	Preparing the database for Oracle GoldenGate	35
	Ensuring ASM connectivity	35
	Configuring character sets	35
	Configuring the Oracle redo logs	36
	Setting redo parallelism for Oracle 9i sources	36
	Avoiding log-read bottlenecks	36
	Mounting logs that are stored on other platforms	37
	Ensuring data availability	37
	Configuring Oracle GoldenGate to read archived logs only	38
	Limitations of ALO mode	38
	Configuring Extract for ALO mode	39
	Adjusting cursors	39
	Setting fetch options	40
	Preparing tables for processing	41
	Disabling triggers and cascade delete constraints	41

Assigning row identifiers.....	42
Getting the database to log key values.....	42
Limiting row changes in tables that do not have a key.....	44
Deferring constraint checking.....	44
Ensuring correct handling of Oracle Spatial objects.....	45
Mapping the tables.....	45
Sizing the XML memory buffer.....	45
Handling triggers on the georaster tables.....	45
Consolidating fetching.....	47
Managing LOB caching.....	48
Additional requirements and procedures for Oracle RAC.....	48
General requirements.....	48
Oracle GoldenGate parameters settings for RAC.....	48
Special procedures on RAC.....	49
Chapter 5 Managing the Oracle DDL replication environment.....	50
Enabling and disabling the DDL trigger.....	50
Maintaining the DDL marker table.....	50
Deleting the DDL marker table.....	51
Maintaining the DDL history table.....	51
Deleting the DDL history table.....	51
Purging the DDL trace file.....	52
Applying database patches and upgrades when DDL support is enabled.....	52
Applying Oracle GoldenGate patches and upgrades when DDL support is enabled...	52
Changing DDL object names after installation.....	53
Restoring an existing DDL environment to a clean state.....	55
Removing the DDL objects from the system.....	56
Chapter 6 Uninstalling Oracle GoldenGate.....	57
Uninstalling Oracle GoldenGate from UNIX.....	57
Uninstalling Oracle GoldenGate from Windows (non-cluster).....	57
Uninstalling Oracle GoldenGate from Windows Cluster.....	58
Appendix 1 Oracle GoldenGate installed components.....	59
Oracle GoldenGate Programs and Utilities.....	59
Oracle GoldenGate subdirectories.....	61
Other Oracle GoldenGate files.....	63
Oracle GoldenGate checkpoint table.....	66
Index.....	67

CHAPTER 1

System requirements and preinstallation instructions

.....

Overview of Oracle GoldenGate for Oracle

With Oracle GoldenGate, you can replicate Oracle DML and DDL operations.

You can move data between similar or dissimilar supported Oracle versions, or you can move data between an Oracle database and a database of another type. Oracle GoldenGate supports the filtering, mapping, and transformation of data, unless otherwise noted in this documentation.

You can replicate DDL operations between similar Oracle databases. When DDL support is active, Oracle GoldenGate does not support the filtering, mapping, and transformation of data.

Supported Platforms

Databases

- Oracle 9.2 (DML and DDL support)
- Oracle 10.1 and 10.2 (DML and DDL support)
- Oracle 11g (DML and DDL support)

Operating systems

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 email 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, there is a primary Extract process that captures source data, a secondary Extract data-pump process that transfers data

.....

across the network, and one or more Replicat processes that apply the replicated data to the target.

- 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 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 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 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.
- Additional disk space on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. The space that is consumed by the trails varies, depending on the volume of data that will be processed. A good starting point is 1 GB. See 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 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 off-loaded 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 `PURGEOLDEXTRACTS`. 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.

[log volume in one hour] x [number of hours downtime] x .4 = 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.

- To run Oracle GoldenGate for multiple Oracle instances on a Windows system, you must install an instance of Oracle GoldenGate for each one

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 they are running fails, you can start them on another node without modifying parameter files, because the processing checkpoints are preserved in the installation directory.
- All nodes in the RAC cluster must have synchronized system clocks. These clocks must be synchronized with the clock on the system where Extract is executed. Oracle GoldenGate compares the time of the local system to the commit timestamps to make critical decisions. For information about synchronizing system clocks, consult www.ntp.org or your systems administrator. See also the IOLATENCY option of the THREADOPTIONS parameter in the *Oracle GoldenGate Windows and UNIX Reference Guide*.

TCP/IP

- 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 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 user installing Oracle GoldenGate must log in as Administrator.
- To install on UNIX, the user installing Oracle GoldenGate must have read and write privileges on the Oracle GoldenGate installation directory.
- The Oracle GoldenGate processes require an operating system user that has privileges to read, write, and delete files and subdirectories in the Oracle GoldenGate directory. In addition, the user for the Manager process requires privileges to control Oracle GoldenGate processes.

- The Extract process requires 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 Extract manually, and then run Manager and Extract as Administrator whenever you start them.
- Dedicate these 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.

Third-party 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 Oracle 11g, you would install the RHAS 4.0 build for Oracle 11g, just as you would on an actual Linux machine.

Database requirements

Database configuration

See Chapter 4 on page 35 for database configuration guidelines.

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

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

Table 1 Database user privileges

User privilege	Extract	Replicat	Manager
CREATE SESSION, ALTER SESSION	X	X	
RESOURCE	X	X ¹	
CONNECT	X	X ²	
SELECT ANY DICTIONARY	X	X	
FLASHBACK ANY TABLE or FLASHBACK ON <owner.table>	X		
SELECT ANY TABLE or SELECT ON <owner.table>	X	X	
INSERT, UPDATE, DELETE ON <target tables>		X	
CREATE TABLE ³		X	
Privileges required to issue DDL operations to target tables (DDL support only).		X	
EXECUTE on DBMS_FLASHBACK package ⁴	X		
GGS_GGSUSER_ROLE ⁵	X		

Table 1 Database user privileges (continued)

User privilege	Extract	Replicat	Manager
DELETE ON Oracle GoldenGate DDL objects ⁶			X
Oracle 10 g ASM privileges	X		
LOCK ANY TABLE ⁷		X	

- ¹ 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.
- ² 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.
- ³ Required if using ADD CHECKPOINTTABLE in GGSCI to enable the database checkpoint feature.
- ⁴ Oracle GoldenGate must make a call to DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER.
- ⁵ 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.
- ⁶ Required only if using parameters that maintain the Oracle GoldenGate DDL database objects.
- ⁷ Required if using the Oracle GoldenGate initial load method that inserts data by means of a direct bulk load to SQL*Loader.

Table 2 Database user privileges — ASM instance

ASM password configuration ¹	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.

¹ 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';

Supported data types

Numeric data types

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

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. This configuration is required 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 non-supported data types for exclusions)

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. SECUREFILE LOBs are always fetched from the database, whether or not they are stored in-row.
 - **Oracle 10g and later:** All 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 LOB is created with any of following options: deduplicate, no_logging, filesystem_like_logging, cache.
 - ▶ 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().
 - ▶ Extract detects an anomaly in the LOB data, such as a missing page number, missing ENDMARKER, or a mismatch between the size that was captured and the expected size.
- When changing a SECUREFILE LOB from one storage to another (such as from ENCRYPT to DECRYPT), Oracle updates the whole table, and Extract captures those updates from the log. Therefore, it will appear as though Oracle updated all of the data blocks that are associated with the table.

NOTE This also can happen when an ALTER TABLE command sets a DEFAULT value to a column that has null values.
- If CLOB columns can store binary data, set the NLS_LANG system environment variable and the NLS_LANGUAGE database parameter to the same value.
- When the size of a large object exceeds 4K, Oracle GoldenGate stores the data in segments within the Oracle GoldenGate trail. The first 4K is stored in the base segment, and the rest is stored in a series of 2K segments. Oracle GoldenGate does not support the filtering, column mapping, or manipulation of large objects of this size. Full Oracle GoldenGate functionality can be used for objects that are 4K or smaller.

XML data types

- XMLType is supported.

Limitations of support

- The source and target objects must be identical. Filtering and manipulation are not supported. However, you can map the XML representation of an object to a character column by means of a COLMAP clause in a TABLE or MAP statement.
- Oracle GoldenGate treats XMLType data as a LOB. There is no size limitation, but see “Large object data types” on page 11 for additional support and limitations.
- A table that contains XMLType columns must have one of the following: a primary key, column(s) with a unique constraint, or a unique index.

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 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 45.

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 data types

- ORDDICOM
- ANYDATA
- ANYDATASET
- ANYTYPE
- BFILE
- MLSLABEL
- TIMEZONE_ABBR
- TIMEZONE_REGION
- URITYPE
- UROWID

Supported objects and operations for 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. This 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 the synchronization of TRUNCATE statements as part of the full DDL synchronization feature or as standalone functionality independent of full DDL synchronization. The standalone TRUNCATE feature supports the replication of TRUNCATE TABLE, but no other DDL. 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.
- Oracle GoldenGate supports direct-load INSERTs. The redo logs must have compatibility 10g or later. Supplemental logging must be enabled. The database must be in archive log mode, and the affected object must be in logging mode (the default). The following methods are supported.
 - /*+ APPEND */ hint (These are not directly captured or replicated, but you can configure Replicat to use an APPEND hint when applying data to a target table.)
 - /*+ BUFFER */ hint
 - /*+ PARALLEL */ hint
 - 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 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.

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.
- Full refreshes are supported for Oracle 10g and later.

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.

Supported objects and operations for DDL

NOTE For detailed information about configuring Oracle GoldenGate for Oracle DDL support, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

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 all DDL operations 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

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.

Non-supported objects and operations for DML

- REF
- Tablespaces and tables created or altered with COMPRESS
- Synonyms
- Database Replay

Non-supported objects and operations for 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.

When you install the Oracle GoldenGate DDL support objects, the script prompts you to permit it to purge the recycle bin, and then it will do so automatically if permission is granted. However, you still must disable the recycle bin manually.

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 database documentation for the correct syntax.

For more information about DDL support, including limitations of support and configuration guidelines, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.

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

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

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

Character	Description
[]	Begin and end brackets
{ }	Begin and end curly brackets (braces)

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

Non-supported characters

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

Table 4 Non-supported characters in object and column names¹

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

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

CHAPTER 2

Installing Oracle GoldenGate



Installation overview

These instructions are for installing Oracle GoldenGate for the first time. Installing Oracle GoldenGate installs all of the components required to run and manage the processing (exclusive of 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.

Upgrades

To upgrade Oracle GoldenGate from one version to another, follow the instructions on <http://www.oracle.com/technology/software/products/goldengate/index.html>.

New installations

To install Oracle GoldenGate for the first time, 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

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

Downloading Oracle GoldenGate

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.

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, you only need to 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>")
```

These 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 ggsa, PASSWORD ggsa  
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 this time **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 systems

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. When Oracle GoldenGate connects to the database locally, all of the following must have the same bit type, either all 32-bit, all 64-bit, or all IA64:

- Oracle GoldenGate version
 - Oracle library versions
 - Database versions
2. When Oracle GoldenGate connects through SQL*Net, the Oracle client library and the Oracle GoldenGate build must match. This means that the Oracle version, the bit type (32-bit, 64-bit, IA64) and the operating system version all must match. If you are using the TRANLOGOPTIONS parameter with the LOGSOURCE option and Oracle GoldenGate connects to transaction logs from a different operating system, the Oracle versions must also be the same.
 3. 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.
 4. 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 /ggs/10.0, the second command in the following table requires these variables to be set:

Command	Requires GG libraries in environment variable?
\$ ggs/10.0 > ./ggsci	No
\$ ggs > ./10.0/ggsci	Yes

To set the variables in Korn shell

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

To set the variables in Bourne shell

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

To set the variables in C shell

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

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

UNIX/Linux library path variables per platform

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

¹ A specific platform may or may not be supported by Oracle GoldenGate for your database. See the Systems Requirements for supported platforms.

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 into a UNIX or Linux cluster

- To install Oracle GoldenGate into a cluster environment, 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 Oracle GoldenGate mediapack.zip file to the UNIX or Linux system and directory where you want Oracle GoldenGate to be installed.
2. Run the command shell and change directories to the new Oracle GoldenGate directory.
3. From the Oracle GoldenGate directory, run the GGSCI program.
GGSCI
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

Configuring Manager and other processes

- To use Oracle GoldenGate, you must configure the Manager process. You must specify a TCP/IP port for Manager to use, and you can specify optional parameters that control dynamic port assignments, trail file maintenance, and other properties.
- To configure other required processes, Oracle GoldenGate security, and other features that enable you to customize Oracle GoldenGate, see the *Oracle GoldenGate Windows and UNIX Administrator's Guide*.

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. For the Oracle GoldenGate installation location, choose a drive that is a resource within the same cluster group that contains the database instance.
3. Ensure that this 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”.
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, verify 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`
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 referenced during installation of the Windows service and during data processing.

Installing Manager as a Windows service

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

To install Manager as a Windows service

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

`install <option> [...]`

Where: <option> is one of the following:

Table 5 INSTALL options

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

Table 5 INSTALL options (continued)

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

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

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

NOTE If Manager is not installed as a service, Oracle GoldenGate users will receive a UAC prompt to confirm the elevation of privileges for Manager when it is started from the GGSCI command prompt. Running other Oracle GoldenGate programs also will trigger 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 group
 - The disk resource containing the Oracle GoldenGate directory
 - The disk resource containing the database transaction log files
 - The disk resource containing 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 GGS_MGR or, if applicable, the custom name specified 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 select Properties.
12. Click the Advanced tab, and 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 has been successfully moved to the second node, the Manager resource should still be offline.
17. Log onto the second node.
18. Install Oracle GoldenGate Manager as a service on this node by running the **install** program as you did on the previous node. If you created a custom name for Manager in the GLOBALS file, that name will be used.
19. Bring the resource online to verify that it is running correctly on this node.
20. Repeat from step 16 for each additional node in the cluster.

Configuring Manager and other processes

- To use Oracle GoldenGate, you must configure the Manager process. You must specify a TCP/IP port for Manager to use, and you can specify optional parameters that control dynamic port assignments, trail file maintenance, and other properties.
- To configure other required processes, Oracle GoldenGate security, and other features that enable you to customize Oracle GoldenGate, see the *Oracle GoldenGate Windows and UNIX Administrator's Guide*.

CHAPTER 3

Installing Oracle GoldenGate DDL support for an Oracle database

.....

Overview of DDL support

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 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.

To install the Oracle GoldenGate DDL environment, you will be installing the database objects shown in Table 6.

WARNING Do not include any Oracle GoldenGate-installed DDL objects in a DDL parameter, in a TABLE parameter, or in a MAP parameter, nor in a TABLEEXCLUDE or MAPEXCLUDE parameter. Make certain that wildcard specifications in those parameters do not include Oracle GoldenGate-installed DDL objects. These objects must not be part of the Oracle GoldenGate configuration, but the Extract process must be aware of operations on them, and that is why you must not explicitly exclude them from the configuration with an EXCLUDE, TABLEEXCLUDE, or MAPEXCLUDE parameter statement.

Table 6 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

Table 6 DDL synchronization objects (continued)

Object	Purpose	Default name
Object ID history table	Contains object IDs of configured objects.	GGG_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.	GGG_DDL_TRIGGER_BEFORE
DDL schema	Contains the DDL synchronization objects.	None; must be specified during installation and in the GLOBALS file.
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

Oracle GoldenGate DDL objects in an Oracle standby configuration

In a physical or logical Oracle standby configuration, the Oracle GoldenGate DDL objects are propagated by the Oracle system (along with the other changes) from the primary server to the standby server. The Oracle GoldenGate DDL trigger will perform its function on either the primary server or the logical standby server, but not on a *physical* standby server, and only if the mode of operation on the server is READ WRITE. On a physical standby, Oracle GoldenGate DDL objects are not active.

If DDL replication by Oracle GoldenGate will not be used on a logical standby server, you can disable the DDL trigger there after it has been installed and enabled on the primary server. If the logical standby server will be used as a source of DDL replication, however, the trigger should be enabled.

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 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 6, *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 6 (except the schema), do one or both of the following:
 - All name changes must be recorded in the params.sql script. Edit this script and change the appropriate parameters. Do not run this script.
 - Some name changes must also be listed in the GLOBALS file. These are listed in Table 7. The correct parameters to use are listed in the Parameter column.

Table 7 GLOBALS parameters for changing DDL object names

Object	Parameter
Marker table	MARKERTABLE <new_table_name> ¹
History table	DDLTABLE <new_table_name>

¹ 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. The trigger will be installed in the SYS schema, which is required by Oracle, and all other DDL synchronization objects will be installed in the schema you created in step 1.

10. Run the `marker_setup` script. This script installs support for the Oracle GoldenGate marker system, which is required for DDL support. You will be prompted for the name of the Oracle GoldenGate schema.
11. Run the `ddl_setup` script. You will be prompted to:
 - Close any open Oracle sessions. Any open sessions are listed in the prompt.
 - 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, which assumes 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 55.
12. Run the `role_setup` script. This script drops and creates the role needed for DDL synchronization. It grants DML permissions on the Oracle GoldenGate DDL objects.
13. Grant the role to all Oracle GoldenGate Extract users. You may need to make multiple grants if the processes have different user names.
14. Run the `ddl_enable.sql` script to enable the DDL trigger.

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:

```
SQL> @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 home 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

Ensuring ASM connectivity

To ensure that Oracle GoldenGate is able to connect to an ASM instance, do the following.

- Make certain that the ASM instance is listed in the `tnsnames.ora` file.
- Make certain that the Oracle listener is listening for new connections to the ASM instance. The `listener.ora` file should contain an entry similar to the following (specifically, the second `SID_DESC`).

```
SID_LIST_LISTENER_DARAN =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /rdbms/oracle/ora1012r/64)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (ORACLE_HOME = /rdbms/oracle/ora1012r/64)
      (SID_NAME = +ASM1)
    )
  )
```

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.

Configuring the Oracle redo logs

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.

NOTE You can also configure Oracle GoldenGate to read exclusively from the archived logs. See page 38.

To ensure the continuity and integrity of processing when Oracle GoldenGate is reading from the online logs, configure the logs as follows.

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.

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.

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 the associated 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. See the TRANLOGOPTIONS documentation in the Oracle GoldenGate *Windows and UNIX Reference Guide*.

NOTE If used, put the TRANLOGOPTIONS statement on one line. Do not use ampersand (&) line terminators to split it into multiple lines.

Ensuring data availability

For best results, enable archive logging and keep the archived logs on the system for the longest time possible. The archives provide a secondary data source should the online logs recycle before Extract is finished with them. Extract may need to recapture transaction data from them in the event of a process or system failure. The following are things to consider when deciding how many archives to keep available.

- Under normal circumstances, 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 must have access to the archived log that contains the beginning of the oldest open transaction at the time of failure and all logs thereafter. It is recommended that you have ready access to these logs if they are moved off the system. The TRANLOGOPTIONS parameter provides a way to specify an alternate archive log location.

If you cannot enable archive logging, configure the online logs to retain enough data so that Extract can 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.

The recommended retention period is at least 24 hours worth of transaction data, including both online and archived logs (if enabled). 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 tables, 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, any 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.

Making logs available in a RAC configuration

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.

Specifying the archive configuration

If the archived logs reside in a location other than the Oracle default, specify that location 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.

Configuring Oracle GoldenGate to read archived logs only

You can configure the Extract process to read exclusively from the archived logs. This is known as *Archived Log Only* (ALO) mode. In this mode, Extract only reads from archived logs that are stored in a specified location. ALO mode allows Oracle GoldenGate to use production logs that are shipped over to a secondary database (such as a standby) as the data source for Oracle GoldenGate. The online logs will not be used. Oracle GoldenGate will connect 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 of ALO mode

- Log resets (RESETLOG) cannot be done on the source database after the standby database is created.
- To replicate DDL when Extract is in ALO mode, Extract must have permission to maintain a SQL*Net connection to the source database.
- 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 RAC requires a dedicated connection to the source server. If that connection is lost, Oracle GoldenGate processing will stop.
- Supplemental logging at the table level and the database level must be enabled for the tables from the source database.

Configuring Extract for ALO mode

1. To connect to the database when Oracle GoldenGate is running on a different server, use a SQL*Net connect string in the USERID parameter. This connect string can then be used for logins for all of the Oracle GoldenGate processes and from GGSCI. Make certain that SQL*Net is configured properly to connect to a remote server, including the correct entries in a TNSNAMES file. 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.

2. 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 for a primary or logical standby database 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.
3. 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.
4. 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*.
5. Add the Extract group by using a timestamp or by using the SEQNO and RBA options of the ADD EXTRACT command. 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.

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. By default,

Extract maintains as many cursors as allowed by the Extract MAXFETCHSTATEMENTS parameter. You might find that the value of this parameter needs to be increased. If so, you might also need to adjust 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 operations that contain LOBs (see the following note), user-defined types, nested tables, and XMLType. 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.

NOTE Fetching of LOBs does not apply to Oracle 10g and later databases, because LOBs are captured directly from the redo log for those versions.

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:

- $\langle \text{undo space} \rangle$ is the number of undo blocks.
- $\langle \text{UNDO_RETENTION} \rangle$ is the value of the UNDO_RETENTION parameter (in seconds).
- $\langle \text{UPS} \rangle$ is the number of undo blocks for each second.
- $\langle \text{overhead} \rangle$ is the minimal overhead for metadata (transaction tables, etc.).

Use the system view V\$UNDOSTAT to estimate $\langle \text{UPS} \rangle$ and $\langle \text{overhead} \rangle$.

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 “To configure Oracle GoldenGate fetch options”). 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>
```

To configure Oracle GoldenGate fetch options

Oracle GoldenGate provides the following parameters to manage fetching.

- To view fetch statistics on-demand, use the `STATS EXTRACT` command with the `REPORTFETCH` option.
- To set the `STATS EXTRACT` command so that it always shows fetch statistics, use the `Extract` parameter `STATOPTIONS` with the `REPORTFETCH` option.
- To control the number of open cursors for prepared queries that Extract maintains in the source database, use the `Extract` parameter `MAXFETCHSTATEMENTS`.
- To control the default fetch behavior of Extract for an Oracle 9i database, use the `FETCHOPTIONS` parameter with the `USESNAPOSHOT` or `NOUSESNAPOSHOT` option. They control whether Extract performs a flashback query or fetches the current image from the table.
- To handle failures of flashback queries, use the `FETCHOPTIONS` parameter with the `USELATESTVERSION` or `NOUSELATESTVERSION` option. A flashback query can fail if the undo retention has expired or the structure of a table has changed. These options control whether Extract fetches the current image from the table or ignores the failure.
- To control the response by Replicat when it processes trail records that include fetched data or column-missing conditions, use the `Replicat` parameter `REPFETCHEDCOLOPTIONS`.

Preparing tables for processing

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

Disabling triggers and cascade delete constraints

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

You have the following options to handle triggers and integrity constraints on the target:

- For Oracle 10.2.0.5 and later patches, and for Oracle 11.2.0.2 and later, 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 `DEFERREFCONST` option to delay checking and enforcement of integrity 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.

Assigning row identifiers

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

How Oracle GoldenGate determines the kind of row identifier to use

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

1. Primary key
2. First unique key alphanumerically 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, or 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.

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.

How to specify your own key for Oracle GoldenGate to use

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

Getting the database to log key values

Use the `ADD TRANDATA` command in `GGSCI` to configure the database to log the 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.

Use ADD TRANDATA to do one of the following:

- Force the logging of primary or unique key columns, or all of the columns to use as a key if there are no defined keys (except non-supported column types). ADD TRANDATA creates a supplemental log group that contains the required columns.
- Log non-key columns that will be used in a KEYCOLS clause. This prevents ADD TRANDATA from logging all of the columns of the table when it determines there is no primary or unique key; the KEYCOLS specification in the parameter file is not checked until processing starts.

ADD TRANDATA must be performed before you start Oracle GoldenGate processing. Before using ADD TRANDATA, see the Oracle GoldenGate *Windows and UNIX Reference Guide* for additional instructions and options.

To capture key values with ADD TRANDATA

1. On the source system, run GGSCI from the Oracle GoldenGate directory.
2. 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 privilege to enable table-level supplemental logging, and <password> is that user's password.

3. Issue the ADD TRANDATA command.

```
ADD TRANDATA <table> [, COLS <columns>] [, NOKEY]
```

Where:

- <table> is the owner and name of the table. You can use a wildcard for the table name but not the owner name.
 - COLS <columns> logs non-key columns that are specified with KEYCOLS.
 - NOKEY prevents the logging of the primary key or unique key. Requires using a KEYCOLS clause in TABLE or MAP and logging the KEYCOLS columns with COL.
4. Log in to SQL*Plus as a user with ALTER SYSTEM privilege, and issue the following command to enable minimal supplemental logging at the database level. This is required to process updates to primary keys and chained rows.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```
 5. To start the supplemental logging, switch the log files.

```
ALTER SYSTEM SWITCH LOGFILE;
```
 6. 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 of the query must be YES.
 - For Oracle 10g, the output of the query must be YES or IMPLICIT.
 7. If using the COLS option to log KEYCOLS columns, create a unique index for these columns on the target, to optimize row retrieval.

Limiting row changes in tables that do not have a key

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

Deferring constraint checking

Constraints might need to be modified in the following cases:

Source and target not both set to DEFERRED

If constraints are DEFERRABLE on the source, the constraints on the target must also be DEFERRABLE. As an alternative, you can use the following parameter statement at the root level of the Replicat parameter file to defer the constraints only for Replicat sessions.

```
SQLEXEC ("alter session set constraint deferred")
```

Key updates affect multiple rows

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:

- Create the constraints as DEFERRABLE on the target tables. Either INITIALLY DEFERRED or INITIALLY IMMEDIATE can be specified.
- Use the Replicat parameter HANDLETPKUPDATE to enable Replicat to set constraints as INITIALLY DEFERRED in its transactions. The constraints are checked when Replicat commits the 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.

Ensuring correct handling of Oracle Spatial objects

If you are replicating 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 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

You might need to change the size of the memory buffer that stores the embedded SYS.XMLTYPE attribute of the SDO_GEOASTER data type. This buffer is controlled by the DBOPTIONS parameter with the XMLBUFSIZE option. The default is 1048576 bytes (1MB) and the maximum is 10 MB. If the data exceeds the buffer size, Extract will abend.

Evaluate your spatial data before starting Oracle GoldenGate processes, so that you can adjust the buffer to suit the size of the XML data. If the METADATA attribute of the SDO_GEOASTER data type in any of the values exceeds the default of 1 MB, you will have to increase this buffer.

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 associated with it to populate the raster data table.

- This trigger must always be enabled in both the source and target environments to ensure consistency of the spatial data.
- When a row in the source georaster table is deleted, the trigger cascades the delete to the raster data table for certain rows. Oracle GoldenGate replicates both deletes. The replicated child delete arrives after the parent delete has already caused the trigger on the target to fire. This redundant delete causes ORA-01403 No data found errors unless the error is handled programmatically through Replicat.

To handle the cascaded deletes, you can use the REPERROR option of the MAP parameter statement that you create for each raster data table. Use Oracle error 1403 as the SQL error, and use any of the response options as 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, a more comprehensive method is to 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), and again to the exceptions table, which will capture 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 prevents having 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*.

Consolidating fetching

(Oracle 9i) Oracle creates multiple updates for the same row when updating an out-of-row LOB, each of which triggers a fetch to the database by Extract to get the data. To avoid multiple fetches and to prevent raster captures from expanding the trail files, you can configure Extract to combine the series of operations into one operation that reflects the net result. Only a single fetch would then be required for that single operation. Use the Extract parameter FETCHOPTIONS with the undocumented options of SUPPRESSDUPLICATES to consolidate the operations that require fetching.

Managing LOB caching

Because Replicat must write LOB data to the target database in fragments, there is the potential for excessive I/O between Replicat and the database. To minimize the effect of I/O on the system, Replicat caches LOB fragments in a buffer of a specified size and only performs a write when the buffer is full. The default LOB write size is 32k.

For example, if the buffer is 25,000 bytes in size, then Replicat writes a block of 25,000 bytes to the database. Given a LOB of 100,000 bytes, Replicat generates I/O only four times to process this LOB.

- To control the size of the LOB buffer, use the Replicat parameter DBOPTIONS with the LOBWRITESIZE <size> option. The higher the value, the fewer I/O calls that are made by Replicat to the database server to write the whole LOB to the database.
- To disable Oracle's LOB caching, use the DISABLELOBCACHING option of DBOPTIONS. By default, Replicat enables Oracle's LOB caching mechanism. If Oracle's LOB caching is disabled, whatever is sent by Replicat to Oracle in one I/O call is written directly to the database media. If Oracle's caching is enabled, whatever is sent by Replicat to Oracle in one I/O call will most likely be cached in Oracle's internal LOB buffer before being applied to the database media.

Additional requirements and procedures for Oracle RAC

General requirements

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

Oracle GoldenGate parameters settings for RAC

- On AIX and Solaris machines, use the Extract parameter THREADOPTIONS with the BINDCPU <n> option. This parameter handles thread-safety issues related to memory that is updated by different processors.
- Oracle GoldenGate queues data in memory before sending it to the target system. The INQUEUESIZE and OUTQUEUESIZE options of the THREADOPTIONS parameter determine how much data to queue. If needed, you can increase the performance of Extract on Oracle RAC by tuning these parameters.
- Oracle GoldenGate detects 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. To control

this behavior, use the `TRANLOGOPTIONS` parameter with the `PURGEORPHANEDTRANSACTIONS | NOPURGEORPHANEDTRANSACTIONS` and `TRANSCLEANUPFREQUENCY` options. This functionality can 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 will abend. To resume processing, you can restart the instance, or you can 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.
- Any time 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 those in the current 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 those 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 will update the Extract checkpoint and confirm to the process that all available archive logs are available to read. It also confirms that all transactions in those archive logs are captured and written to the trail in the correct order.

CHAPTER 5

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 that you installed in Chapter 3 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 `GG_MARKER`.

`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 will fail 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 will be 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:

- “Changing DDL object names after installation” on page 53
- “Restoring an existing DDL environment to a clean state” on page 55
- “Removing the DDL objects from the system” on page 56

Maintaining the DDL history table

You can purge the DDL history table to control its size, but this must be done 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. This will 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. If you delete this table prematurely, you could compromise the integrity of the target DDL.

The history table and the DDL trigger are interdependent. An attempt to drop the history table will fail 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 will be 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:

- “Changing DDL object names after installation” on page 53
- “Restoring an existing DDL environment to a clean state” on page 55
- “Removing the DDL objects from the system” on page 56

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 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 this case, follow the instructions in “Restoring an existing DDL environment to a clean state” on page 55.

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 the Extract process to stop DDL capture.
`STOP EXTRACT <group>`
3. Stop the Replicat process 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 will be prompted for:
 - The name of the Oracle GoldenGate 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 the Replicat process to start DDL replication.
`START REPLICAT <group>`

Changing DDL object names after installation

Follow these steps to change the names of the Oracle GoldenGate DDL schema or other DDL objects after they are installed. This procedure preserves the continuity of source and target DDL operations.

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 the Replicat process 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. To change the DDL schema name, specify the new name in the local GLOBALS file.

```
GGSCHEMA <new_schema_name>
```
9. To change the names of any other objects, do the following:
 - Specify the new names in the params.sql script. Do not run this script.
 - If changing objects in Table 8, specify the new names in the local GLOBALS file. The correct parameters to use are listed in the Parameter column of this table.

Table 8 GLOBALS parameters for changing DDL object names

Object	Parameter
Marker table	MARKERTABLE <new_table_name> ¹
History table	DDLTABLE <new_table_name>

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

10. If using a new schema for the DDL synchronization objects, create it now.
11. Change directories to the Oracle GoldenGate installation directory.
12. Run SQL*Plus and log in as a user that has SYSDBA privileges.
13. Run the ddl_setup script. You will be prompted for:
 - The name of the Oracle GoldenGate schema. If you changed the schema name, use the new one.
 - The installation mode: Select the NORMAL mode to recompile the DDL environment without removing the DDL history table.
14. Run the ddl_enable.sql script to enable the DDL trigger.
15. In GGSCI, start Extract to resume DDL capture.

```
START EXTRACT <group>
```
16. Start the Replicat process 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 has processed all of the DDL and replicated 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 the Replicat process 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 current user 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 in case they are needed for debugging.
12. Run the `marker_setup` script to reinstall the Oracle GoldenGate marker support system. You will be prompted for the name of the Oracle GoldenGate schema.
13. Run the `ddl_setup` script. You will be 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 that you just created to all Oracle GoldenGate users under which the following Oracle GoldenGate processes run: Extract, Replicat, GGSCI, and Manager. You might need to make multiple grants if the processes have different user names.
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 the Replicat process 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 in case they are needed for debugging.

CHAPTER 6

Uninstalling Oracle GoldenGate

.....

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

Uninstalling Oracle GoldenGate from UNIX

1. Run the command shell.
2. (Suggested) Log on as the system administrator, or as a user with permission to issue Oracle GoldenGate commands, and to delete files and directories from the operating system.
3. 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 deletesevents deletesevice
```

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 GGSMMSG.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 deletesevents deletesevice
```

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 GGSMMSG.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 9 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.
defgen	Generates data definitions and is referenced by Oracle GoldenGate processes when source and target tables have dissimilar definitions.
emslnt	Sends event messages created by Collector and Replicat on Windows or UNIX systems to EMS on NonStop systems.
extract	Performs capture from database tables or transaction logs or receives transaction data from a vendor access module.
ggmxinstall	Oracle GoldenGate installation script for the SQL/MX database.



Table 9 Programs and utilities (continued)

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

Oracle GoldenGate subdirectories

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

NOTE Some directories may not exist in all installations.

Table 10 Subdirectories

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

Table 10 Subdirectories (continued)

Directory	Description
dirpcs	<p>Default location for status files. File name format is <group>.<extension> where <group> is the name of the group and <extension> is either pce (Extract), pcr (Replicat), or pcm (Manager).</p> <p>These files are only created while a process is running. The file shows the program name, the process name, the port number, and the process ID.</p> <p>Do not edit these files.</p> <p>Examples: mgr.pcm ext.pce</p>
dirprm	<p>The default location for Oracle GoldenGate parameter files created by Oracle GoldenGate users to store run-time parameters for Oracle GoldenGate process groups or utilities. Written in external ASCII format. File name format is <group name/user-defined name>.prm or mgr.prm.</p> <p>These files may be edited to change Oracle GoldenGate parameter values after stopping the process. They can be edited directly from a text editor or by using the EDIT PARAMS command in GGSCI.</p> <p>Examples: defgen.prm finance.prm</p>
dirrec	<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 <group name><sequence number>.rpt where <sequence number> is a sequential number appended to aged files.</p> <p>Do not edit these files.</p> <p>Examples: fin2.rpt mgr4.rpt</p>
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>

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 11 Other files

Component	Description
bcpfmt.tpl	Template for use with Replicat when creating a run file for the Microsoft BCP/DTS bulk-load utility.
blowfish.txt	Blowfish encryption software license agreement.
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_cleartrace.sql	Script that removes the DDL trace file. (Oracle installations)
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_ora9.sql	A script that gets tablespace information from an Oracle 9 database.
ddl_ora10.sql	A script that disables the Oracle recyclebin and gets tablespace information from an Oracle 10 database.
ddl_pin.sql	Script that pins DDL tracing, the DDL package, and the DDL trigger for performance improvements. (Oracle installations)
ddl_remove.sql	Script that removes the DDL extraction trigger and package. (Oracle installations)
ddl_setup.sql	Script that installs the Oracle GoldenGate DDL extraction and replication objects. (Oracle installations)
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)

Table 11 Other files (continued)

Component	Description
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	Script that creates demonstration tables in the database associated with the Oracle GoldenGate installation.
demo_<db>_insert.sql	Script that inserts initial test data into the demonstration tables.
demo_<db>_misc.sql	Script that simulates transaction activity on the demonstration tables.
ENCKEYS	User-created file that stores encryption keys. Written in external ASCII format.
exitdemo.c	User exit example.
ggmessage.dat	Data file that contains error, informational, and warning messages that are returned by the Oracle GoldenGate processes. The version of this file is checked upon process startup and must be identical to that of the process in order for the process to operate.
ggserr.log	File that logs processing events, messages, errors, and warnings generated by Oracle GoldenGate.
ggsmsg.dll	Windows dynamic link library used by the INSTALL program.
GLOBALS	User-created file that stores parameters applying to the Oracle GoldenGate instance as a whole.
help.txt	Help file for the GGSCI command interface.
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)

Table 11 Other files (continued)

Component	Description
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)
params.sql	Script that contains configurable parameters for DDL support. (Oracle installations)
pthread-win32.txt	License agreement for pthread-VC.dll.
pthread-VC.dll	POSIX threads library for Microsoft Windows.
role_setup.sql	Script that creates the database role necessary for Oracle GoldenGate DDL support. (Oracle installations)
sqlldr.tpl	Template for use with Replicat when creating a control file for the Oracle SQL*Loader bulk-load utility.
start.prm stop.prm	z/OS parmlib members to start and stop the Manager process.
startmgr stopmgr	z/OS Unix System Services scripts to start the Manager process from GGSCI.
startmgrcom stopmgrcom	z/OS system input command for the Manager process.
tcperrs	File containing user-defined instructions for responding to TCP/IP errors.
usrdecs.h	Include file for user exit API.
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 12 Checkpoint table definitions

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

Index

.....

Symbols

\$LD_LIBRARY_PATH variable 26

\$PATH variable 25

A

ADD TRANDATA command 42

ADDEVENTS Windows service option 28

ADDSERVICE Windows service option 29

ALTARCHIVEDLOGFORMAT option, **TRANLOGOPTIONS** 38

ALTARCHIVELOGDEST option, **TRANLOGOPTIONS** 38

ANYDATA data type 14

ANYDATASET data type 14

ANYTYPE data type 14

archive logs, configuring 36

ASMPASSWORD option, **TRANLOGOPTIONS** 9

ASMUSER option, **TRANLOGOPTIONS** 9

Automatic Storage Management (ASM) 9, 35

AUTOSTART Windows service option 29

B

BASICFILE LOB 12

BFILE data type 14

binary data types 11

BINARY DOUBLE data type 10

BINARY FLOAT data type 10

BINARY_INTEGER data type 14

BINDCPU option, **THREDOPTIONS** 48

BLOB data type 11

C

cascade deletes, disabling 41

case, supported 19

category.dll 28

CHAR data type 11

character data types 11

character set, configuring 35

characters

 multibyte 11

 supported in object names 20

CLOB data type 11

clocks, synchronizing 7, 48

cluster, installing on 6, 26, 27, 29

clustered database, installing on 7

clustered tables 17

code points of character set, representing 11

collection types, support for 13

COLS option, **ADD TRANDATA** 43

columns

 number and size supported 15

 unused 15

 virtual 15

components, Oracle GoldenGate 59

COMPRESS option of **CREATE** 18

compression, table 16

connections, to database 9

constraint checking, deferring 44

constraints, integrity 41

CREATE SUBDIRS command 26

cursors, open 39

D

database

- multiple instances 6
- preparing for Oracle GoldenGate 35
- requirements 8
- versions supported 4

Database Replay 18

date and time data types 11

DATE data type 11

DBLOGIN command 43

DDL

- installing support for 31
- managing replication environment 50
- nonsupported objects and operations 18
- supported objects and operations 18

ddl_cleartrace script 52

ddl_disable script 50

ddl_enable script 50

ddl_pin script 34

ddl_remove script 55, 56

ddl_remove_files 56

ddl_remove_spool 55

ddl_setup script 34, 53, 54

deletes, cascaded 41

deleting

- DDL history table 51
- DDL marker table 51
- DDL objects from system 56
- DDL trace file 52

DICOM, support for 14

disk requirements 5

E

environment variables, setting 23, 24

escape sequence for Unicode data 11

EXTERNAL attribute of tables 16

F

fetch options, setting 40

FETCHOPTIONS parameter 41

files installed by Oracle GoldenGate 59

firewall, configuring 7

fixed-point numbers 10

Flashback Query, using 40

floating-point numbers 10

G

GEORASTER, capturing 45

ggmessage.dat file 64

GG\$ tables for DDL support 31

ggs_ddl_trace log 52

GGSMGR default Manager name 27

ggsmsg.dll 28

globalization settings, viewing and configuring 35

GLOBALS file 28

H

HANDLETPKUPDATE parameter 44

I

I/O on redo logs 36

INQUEUESIZE option, THREADOPTIONS 48

installing on

- Linux and UNIX 26
- Windows 27

INTERVAL data type 14

interval partitioning 15

Itanium preinstallation requirements 8

K

key

- absence of 44
- assigning 42
- including in redo record 42
- name, supported characters 20
- transient updates 44

KEYCOLS option, TABLE or MAP 42

L

large objects, limitations on 12

LIBPATH variable 26

libraries, Visual C++ 8

LIMITROWS option, DBOPTIONS 44

Linux, installing on 26

LOB data types

- retention, setting 40
- support for 11

log retention, calculating 37

LOG_ARCHIVE_FORMAT parameter 38

LOG_PARALLELISM parameter 36

logs, configuring 36

LOGSOURCE option, TRANLOGOPTIONS 37

LONG data types 11

M

Manager

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

MANUALSTART Windows service option 29

marker_remove script 55, 56

marker_remove_spool file 55, 56

marker_setup script 34

materialized views 17

MAXFETCHSTATEMENTS parameter 40, 41

Memory requirements for Oracle GoldenGate 4

MGRSERVNAME parameter 28

Microsoft Visual C ++ 2005 SP1 Redistributable Package 8

MLSLABEL data type 14

multi-byte data types 11

multimedia types, support for 14

multiple database instances 6

N

name

- non-supported characters in 21
- supported characters in 20

names, supported 19

NCHAR data type 11

NCLOB data type 11

nested table 13

NLS_LANG parameter 36

NLS_NCHAR_CHARACTERSET parameter 11

NOKEY option, ADD TRANDATA 43

NUMBER data type 10

numeric data types 10

NVARCHAR2 data type 11

O

object tables, support for 13

objects, supported 14

operating systems supported 4

operations, supported 14

Oracle GoldenGate

- installed programs and files 59
- installing 22
- uninstalling 57

ORACLE_HOME and ORACLE_SID settings 23

Oracle, versions supported 4

ORDDicom type, support for 14

OUTQUEUESIZE option, THREADOPTIONS 48

P

parallelism, redo logs 36

partitioning, interval 15

PASSWORD Windows service option 29

PATHMAP option, TRANLOGOPTIONS 37

PCTVERSION parameter 40

platforms, supported 4

ports, required by Oracle GoldenGate 7

pre-installation instructions 4

privileges

- database 9
- operating system 7

PURGEDDLHISTORY parameter 51

PURGEMARKERHISTORY parameter 50

PURGEORPHANEDTRANSACTIONS option, TRANLOGOPTIONS 49

Q

queries

- on undo tablespace 40
- prepared, number of 41

R

RAID drives 37

RAW data type 11

Real Application Cluster 7

recyclebin, Oracle 19

redo logs, configuring 36

REFs 18

removing

- DDL objects 56
- Oracle GoldenGate from system 57

REPFETCHEDCOLOPTIONS parameter 41

REPORTFETCH option, STATOPTIONS 41

role_setup script 34

ROWID data type 14

rows, number and size supported 15

S

SDO_GEOMETRY type 14

SDO_GEORASTER type 14

SDO_TOPO_GEOMETRY type 14

SECUREFILE LOB 12

semantics, considerations for 11

sequences, supported 17

SETENV parameter 23

SHLIB_PATH variable 26

snapshot, using for fetches 40

spaces

- in folder names 27
- in object and column names 21

Spatial types

- capturing 45
- support for 14

SQL statements, prepared 39

subdirectories, creating 26

synonyms 18

system requirements 4

T

tables

- clustered 17
- created as EXTERNAL 16
- preparing for processing 41
- supported types 14

TCP/IP, configuring 7

THREADOPTIONS with BINDCPU 48

TIMESTAMP data type 11

TIMEZONE_ABBR 14

TIMEZONE_REGION 14

transaction logs, configuring 36

TRANSCLEANUPFREQUENCY option, TRANLOGOPTIONS 49

transient primary key update 44

triggers, disabling on target 41

U

undo tablespace, Oracle 40

UNDO_MANAGEMENT parameter 40

UNDO_RETENTION parameter 40

uninstalling Oracle GoldenGate 57

UNIX, installing on 26

URITYPE data type 14

UROWID data type 14

user defined type 13

USER Windows service option 29

user, Oracle GoldenGate 9

USERID parameter 9

V

- VAMSERV program** 60
- VARCHAR2 data type** 11
- VARRAY data type** 14
- VARWIDTHNCHAR parameter** 11
- vcredist_IA64.exe runtime library** 8
- versions of Oracle supported** 4
- views** 16
- virtual columns** 15

virtual machine, support for 8

Visual C ++ 2005 SP1 Redistributable Package 8

W

Windows, installing on 27

X

XMLType 12