Oracle® GoldenGate

SQL Server Installation and Setup Guide

11*g* Release 1 Patch Set 1 (11.1.1.1)

E21507-01

April 2011

Oracle GoldenGate SQL Server Installation and Setup Guide 11g Release 1 Patch Set 1 (11.1.1.1)

E21406-01

Contents

Chapter 1 System requirements and preinstallation instructions 1

Overview of Oracle GoldenGate for SQL Server 1

Supported editions of SQL Server. 1

Supported Platforms 1

Operating system requirements 1

Memory requirements. 1

Disk requirements 2

Storage for Oracle GoldenGate trails 3

Network 3

Operating system privileges. 4

Other programs 4

SQL Server requirements 4

Instance configuration. 4

Database configuration. 5

Database connection 5

Database connectivity drivers 5

Database user for Oracle GoldenGate processes 6

Supported SQL Server data types 6

Non-supported SQL Server data types. 7

Supported objects and operations for SQL Server 7

Non-supported objects and operations for SQL Server 8

Supported and non-supported object names and case. 9

Object names and owners 9

Case sensitivity. 9

Supported characters. 9

Non-supported characters 11

Chapter 2 Installing Oracle GoldenGate 12

Installation overview 12

Downloading Oracle GoldenGate. 12

Installing Oracle GoldenGate on Windows and Windows Cluster 13

Installing Oracle GoldenGate into a Windows Cluster 13

Installing the Oracle GoldenGate files. 13

Specifying a custom Manager name 14

Installing Manager as a Windows service 14

Adding Oracle GoldenGate as a Windows cluster resource 16

Configuring Manager and other processes 17

Chapter 3 Preparing the system for Oracle GoldenGate 18

Choosing Replicat database connection options. 18

Using ODBC or default OLE DB 18

Using OLE DB with NOT FOR REPLICATION 19

Configuring an ODBC connection. 20

Preparing tables for processing 21

Disabling triggers and cascade constraints on the target 21

Assigning row identifiers 21

Limiting row changes in tables that do not have a key 22

Improving IDENTITY replication with array processing. 23

Configuring character sets 23

Chapter 4 Preparing the transaction logs for Oracle GoldenGate. 24

Setting the database to full recovery model 24

Backing up the transaction log. 24

Retaining the log backups. 25

Enabling supplemental logging. 25

Managing the secondary truncation point (SQL Server 2005/2008) 27

Oracle GoldenGate manages the secondary truncation point 27

SQL Server manages the secondary truncation point 28

Making a full database backup before you start Oracle GoldenGate 28

Chapter 5 Uninstalling Oracle GoldenGate 30

Disabling supplemental logging 30

Uninstalling Oracle GoldenGate from Windows (non-cluster) 31

Uninstalling Oracle GoldenGate from Windows Cluster. 32

Appendix 1 Configuring the replication components (SQL Server 2005 pre-CU6 for SP2) 33

What the replication components do for Oracle GoldenGate. 33

Installing and configuring the SQL Server 2005 replication components 33

Configuring and cleaning up the distribution database 34

Appendix 2 Oracle GoldenGate installed components 36

Oracle GoldenGate Programs and Utilities. 36

Oracle GoldenGate subdirectories 37

Other Oracle GoldenGate files 40

Oracle GoldenGate checkpoint table 44

Index 46

CHAPTER 1

System requirements and preinstallation instructions

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

Overview of Oracle GoldenGate for SQL Server

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

Supported editions of SQL Server

- SQL Server 2000/2005: Enterprise and Standard Edition
- SQL Server 2008: Enterprise (capture or delivery) or Standard Edition (delivery only)
- SQL Server 2008 R2: Enterprise (capture or delivery) or Standard Edition (delivery only)

Supported Platforms

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

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

Operating system requirements

Memory requirements

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

- The Oracle GoldenGate GGSCI command interface fully supports up to 300 concurrent Extract and Replicat processes per instance of Oracle GoldenGate. An instance of Oracle GoldenGate equates to one Manager process, which is the main controller process.
- Each Extract and Replicat process needs approximately 25-55 MB of memory, or more depending on the size of the transactions and the number of concurrent transactions.

The actual amount of physical memory that is used by any Oracle GoldenGate process is controlled by the operating system, not the Oracle GoldenGate program. The Oracle GoldenGate cache manager takes advantage of the memory management functions of the operating system to ensure that Oracle GoldenGate processes work in a sustained and efficient manner. Within its cache, it makes use of modern virtual memory techniques by:

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

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

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

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

Disk requirements

Assign the following free disk space:

- 50-150 MB, depending on the database and platform. This includes space for the compressed download file and space for the uncompressed files. You can delete the download file after the installation is complete.
- 40 MB for the working directories and binaries for each instance of Oracle GoldenGate that you are installing on the system. For example, to install two builds of Oracle GoldenGate into two separate directories, allocate 80 MB of space.
- To install Oracle GoldenGate into a cluster environment, install the Oracle GoldenGate binaries and files on a shared file system that is available to all cluster nodes.

- An additional 1 GB of disk space on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. You may need more or less than this amount, because the space that is consumed by the trails depends on the volume of data that will be processed. Start with 1 GB and adjust as needed. See also the following guidelines.

Storage for Oracle GoldenGate trails

To prevent trail activity from interfering with business applications, assign a separate disk or file system to contain the trail files. These files are created during processing to store all of the data that is captured by Oracle GoldenGate. The default size is 10 megabytes, but can be changed during the configuration process. Trail files accumulate but can be purged according to rules set with the PURGEOLDEXTRACTS parameter.

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

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

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

To estimate required trail space

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

   [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.

Network

- Configure the system to use TCP/IP services, including DNS.
- Configure the network with the host names or IP addresses of all systems that will be hosting Oracle GoldenGate processes and to which Oracle GoldenGate will be connecting. Host names are easier to use.

- Oracle GoldenGate requires the following unreserved and unrestricted TCP/IP ports:
  - One port for communication between the Manager process and other Oracle GoldenGate processes.
  - A range of ports for local Oracle GoldenGate communications: can be the default range starting at port 7840 or a customized range of up to 256 other ports.
- Keep a record of the ports that you assigned to Oracle GoldenGate. You will specify them with parameters when configuring the Manager process.
- Configure your firewalls to accept connections through the Oracle GoldenGate ports.

Operating system privileges

Manager: The Manager process can run as a Windows service, or it can run interactively as the current user. The Manager process requires:

- Full control over the files and folders within the Oracle GoldenGate directories.
- Full control over the trail files, if stored in a location other than the Oracle GoldenGate directory.
- Membership in the local Administrators Group (on all nodes in a cluster).

The programs that capture and replicate data (Extract and Replicat) run under the Manager account and inherit those Administrator rights.

Extract and Replicat: See "Database user for Oracle GoldenGate processes" on page 6.

Other programs

- Before installing Oracle GoldenGate on a Windows system, install and configure the Microsoft Visual C ++ 2005 SP1 Redistributable Package. **Make certain it is the SP1 version of this package, and make certain to get the correct bit version for your server**. This package installs runtime components of Visual C++ Libraries. For more information, and to download this package, go to http://www.microsoft.com.
- Oracle GoldenGate fully supports virtual machine environments created with any virtualization software on any platform. When installing Oracle GoldenGate into a virtual machine environment, select a build that matches the database and the operating system of the virtual machine, not the host system.

SQL Server requirements

Instance configuration

- For SQL Server 2008: the following must be true:
  - To capture from SQL Server 2008, the instance must be the Enterprise Edition of SQL Server.
  - Change Data Capture (CDC) must be enabled for Oracle GoldenGate and will be enabled by Oracle GoldenGate by means of the ADD TRANDATA command. See "Enabling supplemental logging" on page 25.
- To capture from a SQL Server 2005 database, you can do one of the following:

- ❍ Run Oracle GoldenGate replication without having to install the SQL Server 2005 replication components. This requires the Microsoft Cumulative Update Package 6 (CU6) or greater for SQL Server 2005 Service Pack 2, or install Service Pack 3 or greater.
- ❍ Run Oracle GoldenGate in conjunction with the SQL Server 2005 replication components. This configuration can be used whether or not Cumulative Update Package 6 (CU6 for SP2) or greater is installed, but is required if that upgrade is not installed. Oracle GoldenGate will use this configuration automatically if CU6 for SP2 or greater is not detected. To use this configuration, all tables that will be replicated must have a declared primary key. See "Configuring the replication components (SQL Server 2005 pre-CU6 for SP2)" on page 33 for more information.

Database configuration

- A SQL Server source database must be set to use the full recovery model.
- Oracle GoldenGate does not support system databases.
- After the source database is set to full recovery, a full database backup must be taken. This backup could be one that was already done, prior to the installation of Oracle GoldenGate, for a database that was previously using the full or bulk-logged recovery model. If you need to make a backup, see "Making a full database backup before you start Oracle GoldenGate" on page 28.
- The log chain on the source database must not be broken between the time of the last full database backup and the time when Oracle GoldenGate is installed. (The log chain is broken if the log was backed up with the "no_log" or "truncate_only" options, or if the recovery model was set to "simple" or "bulk logged" at any time after the initial full database backup was completed.) In addition, the log chain must not be broken after Oracle GoldenGate is installed. For more information, see the Microsoft SQL Server documentation on "Log Chains."

Database connection

Oracle GoldenGate uses ODBC and/or OLE DB to connect to a database:

- **ODBC**: The Extract process uses ODBC (Open Database Connectivity) to connect to a source SQL Server database. The Replicat process uses ODBC to connect to a target SQL Server database to obtain metadata, but can optionally use it for its delivery transactions as well. ODBC must be properly configured. For more information, see "Configuring an ODBC connection" on page 20.
- **OLE DB**: By default, the Replicat process uses OLE DB to connect to a target SQL Server database to perform DML operations. (Thus there are always least two Replicat connections: ODBC for metadata and OLE DB for data). For more information about Replicat connection options, see page 18.

Database connectivity drivers

The following drivers are required:

- SQL Server 2000 ODBC/OLE DB: SQL Server driver
- SQL Server 2005 ODBC/OLE DB: SQL Native Client driver
- SQL Server 2008 ODBC/OLE DB: SQL Server Native Client 10.0 driver

Database user for Oracle GoldenGate processes

The Oracle GoldenGate processes can use either Windows Authentication or SQL Server Authentication to connect to a database.

- To use Windows authentication, the Extract and Replicat processes inherit the login credentials of the Manager process, as identified by the Log On account specified in the Properties of the Manager service. This account must have the privileges listed in Table 1 on the source and target systems.
- To use SQL Server authentication, create a dedicated SQL Server login for Extract and Replicat and assign the privileges listed in Table 2. If using SQL Server authentication, you will need to specify the user and password with the USERID parameter (including the PASSWORD option) in the Extract or Replicat parameter file.

Table 1 Required SQL Server privileges for Manager when using Windows authentication

Oracle GoldenGate Process	Manager privileges if using Local System account	Manager privileges if using local or domain account
Extract (source system)	BUILTIN\Administrators account must be a member of the SQL Server fixed server role System Administrators.	Account must be a member of the SQL Server fixed server role System Administrators.
Replicat (target system)	BUILTIN\Administrators account must be at least a member of the db_owner fixed database role of the target database.	Account must be at least a member of the db_owner fixed database role of the target database.

Table 2 Required SQL Server privileges for Extract and Replicat when using SQL Server authentication

Extract login	Replicat login
Member of the SQL Server fixed server role System Administrators.	At least a member of the db_owner fixed database role of the target database.

Supported SQL Server data types

Oracle GoldenGate supports most SQL Server 2000, 2005, and 2008 data types except those listed under “Non-supported SQL Server data types”.

Limitations of support

- 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.
- (SQL Server 2005 and 2008) Oracle GoldenGate treats XML data as a large object (LOB), as does SQL Server when the XML does not fit into a row. SQL Server 2008 extended XML enhancements (such as lax validation, DATETIME, union functionality) are not supported.

- A system-assigned TIMESTAMP column or a non-materialized computed column cannot be part of a key. A table containing a TIMESTAMP column must have a key, which can be a primary key, a unique constraint, or a substitute key specified with a KEYCOLS clause in the TABLE or MAP statement. See "Assigning row identifiers" on page 14.
- Oracle GoldenGate supports multi-byte character data types and multi-byte data stored in character columns. Multi-byte data is only supported in a like-to-like configuration. Transformation, filtering, and other types of manipulation are not supported for multi-byte character data.
- If data for TEXT, NTEXT, IMAGE, or (if SQL Server 2005 or 2008) VARCHAR (MAX) , NVARCHAR(MAX) and VARBINARY(MAX) columns will exceed the SQL Server default size set for the 'max text repl size option, extend the size. Use sp_configure to view the current value of max text repl size.
- IDENTITY columns are supported as follows:
    - Fully in a uni-directional configuration when the increment values are configured properly.
    - Fully, or with limitations, in a bi-directional configuration, depending on how the Replicat connection is defined. Full support, including support for IDENTITY ranges, requires OLE DB to be used with NOT FOR REPLICATION enabled for the IDENTITY columns and with Replicat operating as the replication user.
    - See Chapter 3 for more information.
- Oracle GoldenGate supports UDT and UDA data of up to 2 GB in size. All UDTs except SQL_Variant and CLR-based ones are supported.
- The support of range and precision for floating-point numbers depends on the host machine. In general, the precision is accurate to 16 significant digits, but you should review the database documentation to determine the expected approximations. Oracle GoldenGate rounds or truncates values that exceed the supported precision.

Non-supported SQL Server data types

- SQL_Variant and CLR-based UDTs
- CLR (common language runtime). This includes SQL Server 2008 built-in CLR data types (such as geometry, geography and hierarchyid).
- (SQL Server 2008) VARBINARY (MAX) column with the FILESTREAM attribute
- Negative dates

Supported objects and operations for SQL Server

- Oracle GoldenGate supports the extraction and replication of DML operations on tables that contain rows of up to 512 KB in length. TEXT, NTEXT, IMAGE, VARBINARY, VARCHAR (MAX), and NVARCHAR(MAX) columns are supported in their full size.
- For SQL Server 2000 and SQL Server 2005, Oracle GoldenGate supports the maximum number of columns and the maximum column size per table that is supported by the database. For SQL Server 2008, Oracle GoldenGate supports the maximum sizes that are permitted for tables that are tracked by CDC.
- Oracle GoldenGate supports SQL Server 2008 tables that use data compression. This includes row compressed format in both ROW and PAGE mode.

- Oracle GoldenGate supports partitioned tables if the table has the same physical layout across all partitions.

Limitations on computed columns

- Oracle GoldenGate supports tables with non-persisted computed columns, but does not capture change data for these columns, because the database does not write it to the transaction log. Replicat does not apply DML to any computed column, even if the data for that column is in the trail, because the database does not permit DML on that type of column. Data from a source *persisted* computed column can be applied to a target column that is not a computed column.
- In an initial load, all of the data is selected directly from the source tables, not the transaction log. Therefore, in an initial load, data values for all columns, including non-persisted computed columns, gets written to the trail or sent to the target, depending on the method that is being used. As when applying change data, however, Replicat does not apply initial load data to computed columns, because the database does not permit DML on that type of column.
- Oracle GoldenGate does not permit a non-persisted computed column to be used in a KEYCOLS clause in a TABLE or MAP statement.
- If a unique key includes a non-persisted computed column, and Oracle GoldenGate must use that key, the non-persisted computed column will be ignored. This might affect data integrity if the remaining columns do not enforce uniqueness.
- If a unique index is defined on any non-persisted computed columns, it will not be used.
- If a unique key or index contains a non-persisted computed 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 non-persisted computed column cannot be used in this identifier, it is possible that Replicat could apply operations containing this identifier to the wrong target rows.

Non-supported objects and operations for SQL Server

- (SQL Server 2005) Operations that are not supported by SQL Server replication. For SQL Server 2005 earlier than CU6 for SP2, tables without a declared primary key also are not supported. SQL Server 2005 tables that are in the Extract configuration are marked for SQL Server transactional replication when you enable supplemental logging in Chapter 4. *Refer to SQL Server Books Online* for a complete list of the operations that are limited by enabling SQL Server Replication.
- (SQL Server 2008) Operations that are not supported by SQL Server Change Data Capture. SQL Server 2008 tables that are in the Extract configuration are marked for SQL Server Change Data Capture when you enable supplemental logging (see page 25). Refer to SQL Server Books Online for a complete list of the operations that are limited by enabling SQL Server Change Data Capture.
- Extraction or replication of DDL (data definition language) operations.
- Extraction from views. The underlying tables can be extracted and replicated.
- Operations by the TextCopy utility and WRITETEXT and UPDATETEXT statements. These features perform operations that either are not logged by the database or are only partially logged, so they cannot be supported by the Extract process.

- Non-native SQL Server transaction log backups, such as those offered by compression utilities. Do not use Oracle GoldenGate Extract on a database where this log backup technology is in use.
- SQL Server 2008 MERGE operations
- SQL Server 2008 compressed log backups
- TDE (Transparent Data Encryption)
- Partitioned tables that have more than one physical layout across partitions
- Partition switching

Supported and non-supported object names and case

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

Object names and owners

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

Case sensitivity

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

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

To preserve case-sensitivity in an Oracle GoldenGate configuration

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

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

For example:

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

Supported characters

Oracle GoldenGate supports alphanumeric characters in 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 names[1]

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

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

Non-supported characters

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

Table 4 Non-supported characters in object and column names[1]

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

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

CHAPTER 2

Installing Oracle GoldenGate

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

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

Installing Oracle GoldenGate installs all of the components that are required to run and manage the processing (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.

Installation overview

The Oracle GoldenGate that will be configured for SQL Server data capture (Extract configuration) must only be installed on the SQL Server database server. The Oracle GoldenGate that will apply data to a SQL Server target can be installed on any Windows server with the proper client connectivity drivers installed.

To install Oracle GoldenGate on a source or target system, the following steps are required:

- Downloading Oracle GoldenGate
- Installing the Oracle GoldenGate software

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

Downloading Oracle GoldenGate

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

1. Navigate to http://edelivery.oracle.com.
2. On the **Welcome** page:
   - Select your language.
   - Click **Continue**.
3. On the **Export Validation** page:
   - Enter your identification information.
   - Accept the **Trial License Agreement** (even if you have a permanent license).
   - Accept the **Export Restrictions**.
   - Click **Continue**.
4. On the **Media Pack Search** page:

- Select the **Oracle Fusion Middleware** Product Pack.
- Select the platform on which you will be installing the software.
- Click **Go**.

5. In the **Results List**:
   - Select the Oracle GoldenGate Media Pack that you want.
   - Click **Continue**.

6. On the **Download** page:
   - Click **Download** for each component that you want. Follow the automatic download process to transfer the mediapack.zip file to your system.

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

Installing Oracle GoldenGate on Windows and Windows Cluster

These instructions apply to all versions of SQL Server. Additional database preparation is required before running the Oracle GoldenGate processes. See Chapter 3.

Installing Oracle GoldenGate into a Windows Cluster

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

Installing the Oracle GoldenGate files

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

   C:\“Oracle GoldenGate” is not valid.

   C:\Oracle_GoldenGate is valid.
3. From the Oracle GoldenGate folder, run the GGSCI program.
4. In GGSCI, issue the following command to create the Oracle GoldenGate working directories.

   ```
   CREATE SUBDIRS
   ```
5. Issue the following command to exit GGSCI.

   ```
   EXIT
   ```

Specifying a custom Manager name

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

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

To specify a custom Manager name

1. From the directory that contains the Manager program, run GGSCI.
2. Issue the following command.

   ```
   EDIT PARAMS ./GLOBALS
   ```

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

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

   ```
   MGRSERVNAME <name>
   ```

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

Installing Manager as a Windows service

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

To install Manager as a Windows service

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

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

Where: <option> is one of the following:

Table 6 INSTALL options

Option	Description
`ADDEVENTS`	Adds Oracle GoldenGate events to the Windows Event Manager. By default, Oracle GoldenGate errors are generic. To produce more specific error content, copy the following files from the Oracle GoldenGate installation directory to the SYSTEM32 directory. category.dll ggsmsg.dll
`ADDSERVICE`	Adds Manager as a service with the name that is specified with the MGRSERVNAME parameter in the GLOBALS file, if one exists, or by the default of GGSMGR. ADDSERVICE configures the service to run as the Local System account, the standard for most Windows applications because the service can be run independently of user logins and password changes. To run Manager as a specific account, use the USER and PASSWORD options.[1] The service is installed to start at system boot time (see AUTOSTART). To start it after installation, either reboot the system or start the service manually from the Services applet of the Control Panel.
`AUTOSTART`	Sets the service that is created with ADDSERVICE to start at system boot time. This is the default unless MANUALSTART is used.
`MANUALSTART`	Sets the service that is created with ADDSERVICE to start manually through GGSCI, a script, or the Services applet of the Control Panel. The default is AUTOSTART.
`USER <name>`	Specifies a domain user account that executes Manager. For <name>, include the domain name, a backward slash, and the user name, for example HEADQT\GGSMGR. By default, the Manager service is installed to use the Local System account.
`PASSWORD <password>`	Specifies the password for the user that is specified with USER.

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

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

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

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

22. In GGSCI, issue the following command.

```
EDIT PARAMS ./GLOBALS
```

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

```
GGSCHEMA <schema>
```

Configuring Manager and other processes

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

- Configure the Manager process with a TCP/IP port and other optional parameters that control dynamic port assignments, trail file maintenance, automatic startup, and other properties.
- Configure Extract and Replicat processes to support reporting, high availability, disaster recovery, and other topologies.
- Configure security to control user access, file security, and data encryption.
- Configure integration, manipulation, and conversion features that enable you to customize Oracle GoldenGate and support the delivery of data across heterogeneous environments.
- Configure utilities and other tools that support Oracle GoldenGate.

CHAPTER 3

Preparing the system for Oracle GoldenGate

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

Choosing Replicat database connection options

The following are the ways in which Replicat can connect to the target database to perform DML operations.

- Connect through ODBC.
- Connect through OLE DB. This is the default and provides slightly better performance than using ODBC.
- Connect through OLE DB as the SQL Server replication user. NOT FOR REPLICATION must be set on IDENTITY columns, foreign key constraints, and triggers.

NOTE In all cases, Replicat always uses ODBC to query for metadata.

Review the following guidelines and procedures to evaluate the advantages and disadvantages of these methods before selecting one to use.

Using ODBC or default OLE DB

If Replicat connects through ODBC or the default OLE DB connection, the following limitations apply:

- To keep IDENTITY columns identical on source and target when using ODBC or default OLE DB, Replicat creates special operations in its transaction to ensure that the seeds are incremented on the target. These steps may reduce delivery performance.
- You must adjust or disable triggers and constraints on the target tables to eliminate the potential for redundant operations.

To use Replicat with ODBC or default OLE DB

1. To use ODBC exclusively, include the DBOPTIONS parameter with the USEODBC option in the Replicat parameter file. (To use default OLE DB, no parameter is required.)
2. Disable triggers and constraints on the target tables. See “Disabling triggers and cascade constraints on the target” on page 21.
3. To use IDENTITY columns in a bidirectional SQL Server configuration, define the IDENTITY columns to have an increment value equal to the number of servers in the configuration, with a different seed value for each one. For example, a two-server installation would be as follows:
    - Sys1 sets seed value at 1 with an increment of 2.
    - Sys2 sets seed value at 2 with an increment of 2.

A three-server installation would be as follows:

- Sys1 sets seed value at 1 with an increment of 3.
- Sys2 sets seed value at 2 with an increment of 3.
- Sys3 sets seed value at 3 with an increment of 3.

4. Configure an ODBC data source. See "Configuring an ODBC connection" on page 20.

NOTE OLE DB uses the ODBC connection settings to derive connection information for OLE DB together with information on which driver to use.

Using OLE DB with NOT FOR REPLICATION

If Replicat connects as the SQL Server replication user through OLE DB, and NOT FOR REPLICATION is enabled for IDENTITY, triggers, and foreign key constraints, the following benefits and limitations apply.

- IDENTITY seeds are not incremented when Replicat performs an insert if the IDENTITY property is marked with NOT FOR REPLICATION. You must ensure data integrity by partitioning the IDENTITY values or by configuring the target database as read-only.
- Triggers are disabled for the Replicat user automatically on the target to prevent redundant operations; however triggers fire on the target for other users.
- Foreign key constraints are not enforced on the target for Replicat transactions. CASCADE updates and deletes are not performed. These, too, prevent redundant operations.
- CHECK constraints are not enforced on the target for Replicat transactions. Even though these constraints are enforced on the source before data is captured, consider whether their absence on the target could cause data integrity issues.

NOTE Normal IDENTITY, trigger, and constraint functionality remains in effect for any users other than the Replicat replication user.

To use Replicat with NOT FOR REPLICATION

1. In SQL Server Management Studio (or other interface) set the NOT FOR REPLICATION flag on the following objects. For active-passive configurations, set it only on the passive database. For active-active configurations, set it on both databases.
   - Foreign key constraints
   - Check constraints
   - IDENTITY columns
   - Triggers (requires textual changes to the definition; see the Microsoft SQL Server documentation for more information.)
2. Partition IDENTITY values or configure the target database as read-only.
3. In the Replicat MAP statements, map the source tables to appropriate targets, and map the child tables that the source tables reference with triggers or foreign-key cascade constraints. Triggered and cascaded child operations are replicated by Oracle GoldenGate, so the referenced tables must be mapped to appropriate targets to preserve data integrity. Include the same parent and child source tables in the Extract TABLE parameters.

NOTE If referenced tables are omitted from the MAP statements, there are no errors to alert you to integrity violations, such as if a row gets inserted into a table that contains a foreign key to a non-replicated table.

4. In the Replicat parameter file, include the DBOPTIONS parameter with the USEREPLICATIONUSER option. For active-passive configurations, use it only on the passive database. For active-active configurations, use it on both databases.
5. Configure an ODBC data source. See "Configuring an ODBC connection".

Configuring an ODBC connection

Follow these instructions to create a SQL Server system data source name (DSN). A DSN stores information about how to connect to a SQL Server database through ODBC (Open Database Connectivity). Create a DSN on each SQL Server source and target system.

NOTE Replicat will always use ODBC to query the target database for metadata.

To create a SQL Server DSN

1. Run one of the following ODBC clients:
   - If using a 32-bit version of Oracle GoldenGate on a 64-bit system, create the DSN by running the ODBCAD32.EXE client from the %SystemRoot%\SysWOW64 folder.
   - If using a 64-bit version of Oracle GoldenGate on a 64-bit system, create a DSN by running the default ODBCAD32.EXE client in **Control Panel**>**Administrative Tools**>**Data Sources (ODBC)**.
   - If using a version of Oracle GoldenGate other than the preceding, use the default ODBC client in **Control Panel>Administrative Tools>Data Sources (ODBC)**.
2. In the **ODBC Data Source Administrator** dialog box of the ODBC client, select the **System DSN** tab, and then click **Add**.
3. Under **Create New Data Source**, select the correct SQL Server driver as follows:
   - SQL Server 2000: SQL Server driver
   - SQL Server 2005: SQL Native Client driver
   - SQL Server 2008: SQL Server Native Client 10.0 driver
4. Click **Finish**. The **Create a New Data Source to SQL Server** wizard is displayed.
5. Supply the following:
   - **Name**: Can be of your choosing. In a Windows cluster, use one name across all nodes in the cluster.
   - **Description**: (Optional) Type a description of this data source.
   - **Server**: Select the SQL Server instance name.
6. Click **Next**.
7. For login authentication, select **With Windows NT authentication using the network login ID** for Oracle GoldenGate to use Windows authentication, or select **With SQL Server authentication using a login ID and password entered by the user** for Oracle GoldenGate to use database credentials. Supply login information if selecting SQL Server authentication.

8. Click **Next**.
9. If the default database is not set to the one that Oracle GoldenGate will connect to, click **Change the default database to**, and then select the correct name. Set the other settings to use ANSI.
10. Click **Next**.
11. Leave the next page set to the defaults.
12. Click **Finish**.
13. Click **Test Data Source** to test the connection.
14. Close the confirmation box and the **Create a New Data Source** box.
15. Repeat this procedure from step 1 on each SQL Server source and target system.

Preparing tables for processing

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

Disabling triggers and cascade constraints on the target

This step applies only if you configured Replicat to connect to the target database through ODBC or the default OLE DB connection. Skip this step if Replicat will be using OLE DB and will operate as the replication user.

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

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

Assigning row identifiers

Oracle GoldenGate requires some form of unique row identifier on the source and target tables to locate the correct target rows for replicated updates and deletes. Different SQL Server versions may have different requirements with respect to row identifiers:

- **SQL Server 2000**: The source tables can have any type of key listed in “How Oracle GoldenGate determines the kind of row identifier to use”.

- **SQL Server 2005 pre-Cumulative Update 6 for SP2**: All of the source tables must have a *primary key*. This is a requirement of the SQL Server Replication component, which Oracle GoldenGate uses as part of its capture methodology.
- **SQL Server 2005 with Cumulative Update 6 for SP2 or higher**: Source tables can have any kind of key listed in "How Oracle GoldenGate determines the kind of row identifier to use".
- **SQL Server 2008**: Source tables can have any kind of key listed in "How Oracle GoldenGate determines the kind of row identifier to use". If there is no primary key identified on a table and there are fixed-length columns, the length of one of those fixed-length columns must be below 3800 bytes. In addition, limitations that apply to Change Data Capture also apply to the source tables.

How Oracle GoldenGate determines the kind of row identifier to use

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

1. Primary key
2. First unique key alphanumerically that does not contain a timestamp or non-materialized computed column.
3. If none of the preceding key types exist (even though there might be other types of keys defined on the table) Oracle GoldenGate constructs a pseudo key of all columns that the database allows to be used in a unique key, excluding those that are not supported by Oracle GoldenGate in a key or those that are excluded from the Oracle GoldenGate configuration. For SQL Server, Oracle GoldenGate enforces the length of row data in target tables without a primary key to be less than 8000 bytes.

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

Using KEYCOLS to specify a custom key

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

Limiting row changes in tables that do not have a key

If a target table has no primary key or unique key, duplicate rows can exist. It is possible for Oracle GoldenGate to update or delete too many rows in such a 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.

Improving IDENTITY replication with array processing

Replicat must continuously toggle IDENTITY_INSERT off and on when applying IDENTITY data to multiple tables in a session, because only one table per session can have IDENTITY_INSERT set to ON. To improve the performance of Replicat in this situation, use the BATCHSQL parameter. BATCHSQL causes Replicat to use array processing instead of applying SQL statements one at a time.

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.

CHAPTER 4

Preparing the transaction logs for Oracle GoldenGate

Setting the database to full recovery model

Oracle GoldenGate requires a SQL Server source database to be set to the full recovery model.

To verify or set the recovery model

1. Connect to the SQL Server instance with either Enterprise Manager for SQL Server 2000 or SQL Server Management Studio for SQL Server 2005 and 2008.
2. Expand the **Databases** folder.
3. Right-click the source database, and then select **Properties**.
4. Select the **Options** tab.
5. Under **Recovery**, set **Model** to **Full if not already**.
6. If the database was in Simple recovery or never had a Full database backup, take a Full database backup before starting Extract. See page 28.
7. Click **OK**.

Backing up the transaction log

The Extract process may occasionally require access to a log backup on the source system. This happens when the required log records are no longer available in the online log and have been moved to log backups.

Oracle GoldenGate requires the log backup files on a source system to meet the following conditions:

- The backup file must remain in the original location where the backup was made. If the log backup files must be removed from this location, you can override this condition by using the TRANLOGOPTIONS parameter with the ALTARCHIVELOGDEST option. This parameter enables Oracle GoldenGate to search for the backup file in a different location, but must only be used when the backup files are not in the original location.
- The backup must not be striped across multiple files.
- The backup must be made to a DISK device.
- The backup must be a native SQL Server backup made by issuing the BACKUP LOG command (or the corresponding GUI command). Third-party log backup tools are not supported.
- The backup files must be non-encrypted and non-compressed.
- Every log backup must have a distinct file name. Do not overwrite backup files to the same name as old ones.

For optimal performance of the Extract process, do the following:

- Make only one log backup per backup file.
- Do not mix backup file types in one file.

Retaining the log backups

Retain enough log backups so that if you stop Extract or there is an unplanned outage, Extract can start again from its checkpoints. Extract must have access to the data in the transaction log or a log backup that contains the start of the oldest uncommitted unit of work, and all log backups thereafter.

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

- Alter Extract to capture from a later point in time for which log data is available (and accept possible data loss on the target).
- Resynchronize the source and target tables, and then start the Oracle GoldenGate environment over again.

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

Enabling supplemental logging

These instructions apply to new installations of Oracle GoldenGate for all supported SQL Server versions. You will enable supplemental logging with the ADD TRANDATA command so that Extract can capture the information that is required to reconstruct SQL operations on the target. This is more information than what SQL Server logs by default.

ADD TRANDATA must be issued for all tables that will be replicated with Oracle GoldenGate. It does the following:

- **SQL Server 2000**: ADD TRANDATA sets a flag on the sysobjects table that tells SQL Server to log full before and after images.

  NOTE This flag is shared by update triggers and replication. Avoid dropping an update trigger on a table that is in the Extract configuration, because this will drop the supplemental logging for that table and generate an error: “Updates are not supported on tables that do not have TRANDATA added.” There is a resolution for this in the Oracle GoldenGate *Windows and UNIX Troubleshooting and Tuning Guide*.

- **SQL Server 2005 updated to CU6 for SP2 or later**: ADD TRANDATA calls the sys.sp_extended_logging stored procedure.

  WARNING For SQL Server with CU6 for SP2 or later, if Extract and transactional replication will co-exist, *do not* issue ADD TRANDATA for tables that do not have a declared primary key. Oracle GoldenGate enables those tables for replication, but tables without a primary key are not supported by SQL Server transactional replication and will cause the log reader agent to fail.

- **SQL Server 2005 pre-CU6 for SP2**: ADD TRANDATA creates the following:
  - ❍ A replication publication named [<source database name>]: GoldenGate<source database name> Publisher. To view this publication, look under **Replication>Local Publications** in SQL Server Management Studio. This procedure adds the specified table to the publication as an article.
  - ❍ A SQL Server Log Reader Agent job for the publication. This job cannot run concurrently with an Extract process in this configuration. To disable this job, see "Managing the secondary truncation point (SQL Server 2005/2008)" on page 27.

- **SQL Server 2008**: ADD TRANDATA enables Change Data Capture (CDC) and creates a minimal Change Data Capture on the specified table.
  - ❍ Oracle GoldenGate does not use the CDC tables other than as necessary to enable supplemental logging.
  - ❍ As part of enabling CDC, SQL Server creates two jobs per database: <dbname>_capture and <dbname>_cleanup. The <dbname>_capture job adjusts the secondary truncation point and captures data from the log to store in the CDC tables. The <dbname>_cleanup job ages and deletes data captured by CDC.
  - ❍ Using the TRANLOGOPTIONS parameter with the MANAGESECONDARYTRUNCATIONPOINT option for Extract removes the <dbname_capture> job, preventing the overhead of the job loading the CDC tables.
  - ❍ The alternative (using TRANLOGOPTIONS with NOMANAGESECONDARYTRUNCATIONPOINT) requires the SQL Server Agent to be running and requires the <dbname>_capture and <dbname>_cleanup jobs to be retained. You will probably need to adjust the <dbname>_cleanup data retention period if the default of three days is not acceptable for storage concerns.
  - ❍ For more information on TRANLOGOPTIONS, see"Managing the secondary truncation point (SQL Server 2005/2008)" on page 27.

To enable supplemental logging

1. On the source system, run GGSCI.
2. Log into the database from GGSCI.

```
DBLOGIN SOURCEDB <DSN>[, USERID <user>, PASSWORD <password>]
```

Where:

- ❍ SOURCEDB <DSN> is the name of the SQL Server data source.
- ❍ USERID <user> is the Extract login and PASSWORD <password> is the password that is required if Extract uses SQL Server authentication.

3. In GGSCI, issue the following command for each table that is, or will be, in the Extract configuration. You can use a wildcard to specify multiple table names, but not owner names.

```
ADD TRANDATA <owner>.<table>
```

NOTE The ADD TRANDATA command for SQL Server automatically filters out tables with dbo.MS and dbo.sys in their names, because it is assumed they are system tables which should not be replicated. If you have tables that contain either of these naming conventions, do not use wildcards to specify them with ADD TRANDATA. Instead, issue the ADD TRANDATA command separately for each one that you want to be replicated.

Managing the secondary truncation point (SQL Server 2005/2008)

When you enable supplemental logging with the ADD TRANDATA command for at least one table in a SQL Server 2005 or SQL Server 2008 database, a secondary truncation point is created in the transaction log that has to be moved for log space to be released as needed, following subsequent log backups. Use the TRANLOGOPTIONS parameter to control whether Extract or SQL Server manages the secondary truncation point. This is a required parameter.

Oracle GoldenGate manages the secondary truncation point

Use TRANLOGOPTIONS with the MANAGESECONDARYTRUNCATIONPOINT option if Extract will *not* be running concurrently (for the same source database) with:

- (SQL Server 2005, any version) SQL Server transactional replication
- (SQL Server 2008) SQL Server transactional replication and/or CDC that is configured for applications other than Oracle GoldenGate

MANAGESECONDARYTRUNCATIONPOINT enables Extract to manage the secondary truncation point. Extract will move the secondary truncation point forward at a defined interval by issuing the following T-SQL statement:

```
EXEC sp_repldone @xactid = NULL, @xact_segno = NULL, @numtrans = 0,
@time = 0, @reset = 1
```

When Extract manages the secondary truncation point for SQL Server 2005 pre-CU6 for SP2, you must stop and disable the Log Reader Agent job as follows:

1. In SQL Server Management Studio, connect to the SQL Server 2005 instance.
2. Start SQL Server Agent (if not running).
3. Expand the **SQL Server Agent** folder.
4. Expand the **Jobs** folder.
5. Find the job that was created by the ADD TRANDATA command. The name shows the server, the instance, and the database name, plus the number of the publication. It looks similar to the following:

   ```
   KTANCO\SQL2005-SQLLBE_SRC-1
   ```

6. Right click the job, and then select **Stop Job**.
7. Right click the job again, and then select **Disable**.

NOTE If the Extract process running against a pre-CU6 source is suspended for a longer time than the normal log backup frequency, you will need to re-enable and start the SQL Server Replication Log Reader Agent job temporarily to manage the last distributed transaction. Stop and disable the job before you restart Extract.

For SQL Server 2008, MANAGESECONDARYTRUNCATIONPOINT also routinely checks and deletes the CDC capture job. This prevents change data from being collected for all of the tables in the database, resulting in:

- better performance by Extract
- less storage space used by captured data in the CDC tables
- fewer records in the transaction log

NOTE Using TRANLOGOPTIONS MANAGESECONDARYTRUNCATIONPOINT for Extract when either SQL Server transactional replication and/or CDC for applications other than Oracle GoldenGate are running at the same time causes the SQL Server log reader agent or CDC capture job to fail.

Extract when it manages the secondary truncation point

When Extract manages the secondary truncation point and is stopped for a longer period of time than the log backup frequency, data in the transaction log will not be freed after log backups. This causes the transaction log to grow. To allow space to be freed from the log after backups, mark the data as 'distributed' by doing one of the following, depending on whether you intend to continue capture with the same Extract or delete that Extract.

To continue capture with the same Extract group

Before starting this Extract group again, retain all of the log backups that contain data that still needs to be processed.

1. To determine the log-read checkpoint that shows the oldest log that Extract needs, use the INFO EXTRACT command with the SHOWCH option in GGSCI.

   ```
   INFO EXTRACT <group>, SHOWCH
   ```

2. Either manually or from within a SQL Server Agent job, move the secondary truncation point by issuing the following T-SQL command against the source database. Run this command approximately every minute until you start Extract.

   ```
   EXEC sp_repldone @xactid = NULL, @xact_segno = NULL, @numtrans = 0,
   @time = 0, @reset = 1
   ```

3. Stop and disable the SQL Server Agent job or disconnect the query session prior to restarting Extract.

To remove capture with the Extract group

To delete this Extract group, you can disable Oracle GoldenGate capture instead of manually moving the secondary truncation point. To disable Oracle GoldenGate capture, follow the steps for disabling supplemental logging as directed in “Uninstalling Oracle GoldenGate from Windows (non-cluster)” on page 70.

SQL Server manages the secondary truncation point

Use TRANLOGOPTIONS with the NOMANAGESECONDARYTRUNCATIONPOINT option if Extract *will* run concurrently (for the same source database) with:

- (SQL Server 2005, any version) SQL Server transactional replication
- (SQL Server 2008) SQL Server transactional replication and/or CDC that is configured for applications other than Oracle GoldenGate

SQL Server will manage the secondary truncation point.

Making a full database backup before you start Oracle GoldenGate

These instructions are for making a full database backup according to the Oracle GoldenGate requirements stated on page 4.

To make a full backup of the database

1. Configure Oracle GoldenGate to your requirements, and then return to this procedure when you are ready to begin initial synchronization and start change replication. To configure Oracle GoldenGate, see the Oracle GoldenGate *Windows and UNIX Administrator's Guide*.
2. Connect to the SQL Server instance with either Enterprise Manager for SQL Server 2000 or SQL Server Management Studio for SQL Server 2005 and 2008.
3. Expand the **Databases** folder.
4. Right click the source database name, and then select **All Tasks > Backup Database**.
5. Select **Database - Complete**. This option makes a full database backup and ensures that no transaction information is lost when Oracle GoldenGate starts.
6. Under **Destination**, click **Add** to specify the backup file name and location.
7. Click **OK**. The backup file is added to the **Destination** list box in the **SQL Server Backup** dialog box.
8. Click **OK** to start the backup.

CHAPTER 8
Uninstalling Oracle GoldenGate

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

Disabling supplemental logging

These steps must be performed before you uninstall Oracle GoldenGate.

To disable supplemental logging from SQL Server 2000

1. Stop Extract.

   ```
   STOP EXTRACT <group>
   ```

2. Issue the DELETE TRANDATA command for all of the tables that are listed in the TABLE parameter in the Extract parameter file. A wildcard can be used for the table name but not the owner name.

   ```
   DELETE TRANDATA <owner.table>
   ```

To disable supplemental logging from SQL Server 2005 pre-CU6 for SP2

1. Stop Extract.

   ```
   STOP EXTRACT <group>
   ```

2. Run DBCC OPENTRAN against the database to verify that there are no open transactions.
3. Wait for open transactions to commit before proceeding to the next step.
4. Manually delete the Oracle GoldenGate publication from the database.

To disable supplemental logging from SQL Server 2005 with CU6 for SP2 or later

When there are concurrent native SQL Server transactional replication publications for the source SQL Server 2005 with CU6 for SP2 database, issue the DELETE TRANDATA command for all of the tables that are listed in the TABLE parameter. A wildcard can be used for the table name, but not the owner name. The syntax is:

```
DELETE TRANDATA <owner>.<table>
```

When there are no concurrent native SQL Server transactional replication publications for the source SQL Server 2005 with CU6 for SP2 database, follow these steps:

1. Pause application and transactional activity. There should be no new records occurring between the timing of step 6 and the completion of step 8.
2. Stop Extract.

   ```
   STOP EXTRACT <group>
   ```

3. Run DBCC OPENTRAN against the database to verify that there are no open transactions.
4. Wait for open transactions to commit before proceeding to the next step.
5. Execute EXECUTE sp_repltrans to verify that there are no undistributed transactions in the database.
6. If the EXECUTE sp_repltrans returned any records, execute the following command against the database to clear any remaining undistributed transactions. Otherwise, skip to step 8.

```
EXECUTE sp_repldone @xactid=NULL, @xact_segno=NULL, @numtrans=0,
@time=0, @reset=1
```

7. Execute EXECUTE sp_repltrans again to verify that all transactions are distributed.
8. Issue the DELETE TRANDATA command for all of the tables that are listed in the TABLE parameter in the Extract parameter file. A wildcard can be used for the table name but not the owner name.

```
DELETE TRANDATA <owner.table>
```

To disable supplemental logging from SQL Server 2008

When there are concurrent native SQL Server transactional replication publications for a SQL Server 2008 database, or there are Change Data Capture configurations for applications other than Oracle GoldenGate, issue the DELETE TRANDATA command for all of the tables that are listed in the TABLE parameter in the Extract parameter file. A wildcard can be used for the table name but not the owner name. The syntax is:

```
DELETE TRANDATA <owner>.<table>
```

When there are no concurrent native SQL Server transactional replication publications for a SQL Server 2008 database, and there are no Change Data Capture configurations for applications other than Oracle GoldenGate, follow these steps:

1. Verify that there are no open transactions by running DBCC OPENTRAN against the database.
2. Wait for any open transactions to commit before proceeding to the next step.
3. Run the following against the source database:

```
EXEC sys.sp_cdc_disable_db
```

Uninstalling Oracle GoldenGate from Windows (non-cluster)

1. (Suggested) Log on as the system administrator or as a user with permission to issue Oracle GoldenGate commands and to delete files and directories from the operating system.
2. From the Oracle GoldenGate installation folder, run GGSCI.
3. Stop all Oracle GoldenGate processes.
4. Stop the Manager program or service.
5. Exit GGSCI.
6. Click **Start** > **Run**, and type cmd in the **Run** dialog box to open the command console.

7. Change directories to the Oracle GoldenGate installation directory.
8. Run the install program using the following syntax.

   ```
   install deleteevents deleteservice
   ```

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

Uninstalling Oracle GoldenGate from Windows Cluster

1. Working from the node in the cluster that owns the cluster group that contains the Manager resource, run GGSCI and then stop any Extract and Replicat processes that are still running.
2. Use the Cluster Administrator tool to take the Manager resource offline.
3. Right click the resource and select **Delete** to remove it.
4. Click **Start** > **Run**, and type cmd in the **Run** dialog box to open the command console.
5. Change directories to the Oracle GoldenGate installation directory.
6. Run the install program using the following syntax.

   ```
   install deleteevents deleteservice
   ```

   This command stops Oracle GoldenGate events from being reported to the Windows Event Manager and removes the Manager service.
7. Delete the CATEGORY.DLL and GGSMSG.DLL files from the Windows SYSTEM32 folder.
8. Move the cluster group to the next node in the cluster, and repeat from step 4.
9. Delete the Oracle GoldenGate installation folder.
10. Drop any Oracle GoldenGate-related objects from the database as needed.

APPENDIX 1

Configuring the replication components (SQL Server 2005 pre-CU6 for SP2)

Follow these instructions only if the following are true:

- Your SQL Server 2005 source is *not* updated to Microsoft Cumulative Update Package 6 (CU6) for SQL Server 2005 Service Pack 2 or later.
- The SQL Server native replication components and a distribution database *are not* installed and configured in this SQL Server source.

For a SQL Server 2005 source that is not upgraded to CU6 for SP2, the native replication components must be used with a distribution database to support Oracle GoldenGate replication. Install the SQL Server replication components only if they are not installed and configured already.

What the replication components do for Oracle GoldenGate

- Oracle GoldenGate can operate concurrently with SQL Server 2005 Replication against the same database. Oracle GoldenGate issues a warning message when it detects a Log Reader Agent that is already attached to the database.
- One distribution database can be used for all SQL Server source databases. Oracle GoldenGate does not depend on the distribution database, but instead reads the logs directly, so you can set transaction retention to zero.

Installing and configuring the SQL Server 2005 replication components

These steps install the SQL Server replication components on the local hard drive.

1. On the source system, run Setup.exe in the Servers folder of the SQL Server installation directory.
2. Complete the initial licensing pages.
3. On the **Components to Install** page, select the database features that you want to install.
4. Click **Advanced** to open the **Feature Selection** page.
5. Expand **Database Services**.
6. Click **Replication**, and then select **Will be installed on local hard drive** from the drop-down menu.
7. Click **Next**.
8. Complete the database setup according to your requirements.
9. Run SQL Server Management Studio.

10. Expand the SQL Server instance.
11. Select the **Replication** folder.
12. Right click **Replication**, and then select **Configure Distribution** to start the **Configure Distribution** wizard.
13. Select the local instance as its own distribution database, or select a remote one.
14. Click **Next**.
15. Set the SQL Server Agent service to start automatically, if possible.
16. Click **Next**.
17. Accept the default Snapshot Folder, or choose a new location. Oracle GoldenGate does not use the Snapshot Folder.
18. Click **Next**.
19. Accept the default database name and file locations, or modify them as needed.
20. Click **Finish**, and then click **Finish** again to create the distribution database and finish the setup.

Configuring and cleaning up the distribution database

After the replication components are installed, perform the following steps to:

- Set transaction retention to 0.
- Disable replication alerts.
- Stop and disable SQL Server Agent replication jobs, which are created during the distribution database setup.

To configure transaction retention

1. In SQL Server Management Studio, expand the SQL Server instance.
2. Right-click the **Replication** folder, and then select **Distributor Properties**.
3. Click **General Properties**.
4. To the right of the **History Retention** column, next to the distribution database name, click the ellipsis (...) button to open the **Distribution Database Properties**.
5. Set **Transaction retention** to:
    - **At least 0 Hours**
    - **But not more than 0 Hours**
6. On the same page, set **History retention** to 0.
7. Click **OK**.

To stop and disable SQL Server Agent replication jobs and alerts

1. In SQL Server Management Studio, connect to the SQL Server instance.
2. Start SQL Server Agent.
3. Expand the **SQL Server Agent** folder, and then expand the **Jobs** folder.

4. Right click each of the following jobs, and then select **Stop Job** (if running), then **Disable**.
    - Agent history clean up: <distribution database name>
    - Distribution clean up: <distribution database name>
    - Expired subscription clean up
    - Reinitialize subscriptions having data validation failures
    - Replication agents checkup
5. Under the **SQL Server Agent** folder, expand the **Alerts** folder.
6. Select all alerts that begin with the name "Replication," and then select Disable.

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 13 Programs and utilities

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

Table 13 Programs and utilities (continued)

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

Oracle GoldenGate subdirectories

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

NOTE Some directories may not exist in all installations.

Table 14 Subdirectories

Directory	Description
br	Contains the checkpoint files for the bounded recover feature.
cfg	Contains the property and xml files that are used to configure Oracle GoldenGate Monitor.

Table 14 Subdirectories (continued)

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

Table 14 Subdirectories (continued)

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

Table 14 Subdirectories (continued)

Directory	Description
dirwal	Contains the Oracle Wallet that supports Oracle GoldenGate Monitor. This directory is not installed until the utility that creates the wallet is run.
UserExitExamples	Contains sample files to help with the creation of user exits.

Other Oracle GoldenGate files

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

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

Table 15 Other files

Component	Description
bcpfmt.tpl	Template for use with Replicat when creating a run file for the Microsoft BCP/DTS bulk-load utility.
bcrypt.txt	Blowfish encryption software license agreement.
cagent.dll	Contains the Windows dynamic link library for the Oracle GoldenGate Monitor C sub-agent.
category.dll	Windows dynamic link library used by the INSTALL program.
chkpt_<db>_create.sql	Script that creates a checkpoint table in the local database. A different script is installed for each database type.
db2cntl.tpl	Template for use with Replicat when creating a control file for the IBM LOADUTIL bulk-load utility.
ddl_access.tpl	Template used by the DDLGEN utility to convert source DDL to Microsoft Access DDL.
ddl_cleartrace.sql	Script that removes the DDL trace file. (Oracle installations)
ddl_db2.tpl	Template used by the DDLGEN utility to convert source DDL to DB2 DDL (Linux, UNIX, Windows).
ddl_db2_os390.tpl	Template used by the DDLGEN utility to convert source DDL to DB2 DDL (z/OS systems).
ddl_ddl2file.sql	Script that saves DDL from the marker table to a file.

Table 15 Other files (continued)

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

Table 15 Other files (continued)

Component	Description
ddl_sqlmx.tpl	Template used by the DDLGEN utility to convert Tandem Enscribe DDL to NonStop SQL/MX DDL.
ddl_status.sql	Script that verifies whether or not each object created by the Oracle GoldenGate DDL support feature exists and is functioning properly. (Oracle installations)
ddl_staymetadata_off.sql ddl_staymetadata_on.sql	Scripts that control whether the Oracle DDL trigger collects metadata. This script runs programmatically; do not run it manually.
ddl_sybase.tpl	Template used by the DDLGEN utility to convert source DDL to Sybase DDL.
ddl_tandem.tpl	Template used by the DDLGEN utility to convert source DDL to NonStop SQL DDL.
ddl_trace_off.sql ddl_trace_on.sql	Scripts that control whether DDL tracing is on or off.
ddl_tracelevel.sql	Script that sets the level of tracing for the DDL support feature. (Oracle installations)
debug files	Debug text files that may be present if tracing was turned on.
demo_<db>_create.sql demo_more_<db>_create.sql demo_<db>_insert.sql demo_more_<db>_insert.sql demo_<db>_lob_create.sql demo_<db>_misc.sql	Scripts that create and populate demonstration tables for use with tutorials and basic testing.
.dmp files	Dump files created by Oracle GoldenGate processes for tracing purposes.
ENCKEYS	User-created file that stores encryption keys. Written in external ASCII format.
exitdemo.c	User exit example.
freeBSD.txt	License agreement for FreeBSD.
ggmessage.dat	Data file that contains error, informational, and warning messages that are returned by the Oracle GoldenGate processes. The version of this file is checked upon process startup and must be identical to that of the process in order for the process to operate.

Table 15 Other files (continued)

Component	Description
ggserr.log	File that logs processing events, messages, errors, and warnings generated by Oracle GoldenGate.
ggsmsg.dll	Windows dynamic link library used by the INSTALL program.
GLOBALS	User-created file that stores parameters applying to the Oracle GoldenGate instance as a whole.
help.txt	Help file for the GGSCI command interface.
icudt38.dll icuin38.dll icuuc38.dll	Windows shared libraries for International Components for Unicode.
jagent.bat	Windows batch file for the Java Agent for Oracle GoldenGate Monitor.
jagent.log jagentjni.log	Log files for the Oracle GoldenGate Monitor Agent.
jagent.sh	UNIX shell script for the Java Agent for Oracle GoldenGate Monitor
LGPL.txt	Lesser General Public License statement. Applies to free libraries from the Free Software Foundation.
libxml2.dll	Windows dynamic link library containing the XML library for the Oracle GoldenGate XML procedures.
libxml2.txt	License agreement for libxml2.dll.
marker.hist	File created by Replicat if markers were passed from a NonStop source system.
marker_remove.sql	Script that removes the DDL marker table. (Oracle installations)
marker_setup.sql	Script that installs the Oracle GoldenGate DDL marker table. (Oracle installations)
marker_status.sql	Script that confirms successful installation of the DDL marker table. (Oracle installations)
notices.txt	Third-party software license file.
params.sql	Script that contains configurable parameters for DDL support. (Oracle installations)
pthread-win32.txt	License agreement for pthread-VC.dll.

Table 15 Other files (continued)

Component	Description
pthread-VC.dll	POSIX threads library for Microsoft Windows.
prvtclkm.plb	Supports the replication of Oracle encrypted data.
pw_agent_util.bat pw_agent_util.sh	Script files that support the Oracle GoldenGate Monitor Agent.
role_setup.sql	Script that creates the database role necessary for Oracle GoldenGate DDL support. (Oracle installations)
sqlldr.tpl	Template for use with Replicat when creating a control file for the Oracle SQL*Loader bulk-load utility.
start.prm stop.prm	z/OS paramlib members to start and stop the Manager process.
startmgr stopmgr	z/OS Unix System Services scripts to start the Manager process from GGSCI.
startmgrcom stopmgrcom	z/OS system input command for the Manager process.
tcperrs	File containing user-defined instructions for responding to TCP/IP errors.
usrdecs.h	Include file for user exit API.
xerces-c_2_8.dll	Apache XML parser library.
zlib.txt	License agreement for zlib compression library.

Oracle GoldenGate checkpoint table

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

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

Table 16 Checkpoint table definitions

Column	**Description**
GROUP_NAME (primary key)	The name of a Replicat group using this table for checkpoints. There can be multiple Replicat groups using the same table.

Table 16 Checkpoint table definitions

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

Index

A

ADD TRANDATA command 25, 26

ADDEVENTS Windows service option 15

ADDSERVICE Windows service option 15

AUTOSTART Windows service option 15

B

BATCHSQL parameter 23

before images, logging 25

C

capture job, creating 26

cascade deletes, disabling 21

cascade updates, disabling 21

category.dll 15

Change Data Capture (CDC)
- disabling 30
- use by Oracle GoldenGate 26

character sets, configuring 23

characters
- multi-byte 7
- supported in object names 10

cleanup job, creating 26

cluster
- installing on 2, 13, 16
- removing from 32

columns
- datatypes supported 6
- number and size supported 7

components, Oracle GoldenGate 36

constraints, integrity
- disabling 21
- requirements 18, 19

CU6 for SP2 update 5

D

data source name (DSN), creating 20

data types, supported 6

database
- character sets 23
- distribution, SQL Server 2005 33
- requirements 4
- versions supported 1

DBCC OPENTRAN 30

DELETE TRANDATA command 30

deletes, cascaded 21

disk requirements 2

distribution database, configuring 33, 34

E

Enterprise Edition 4

F

files, installed by Oracle GoldenGate 36

firewall, configuring 4

full database backup 5, 28

full recovery model, setting 24

G

ggmessage.dat file 42

GGSMGR default Manager name 14

ggsmsg.dll 15

GLOBALS file 14

I

IDENTITY columns 18, 19, 23

IMAGE data type 7

INFO EXTRACT command 25

installing on Windows 13

K

key
- absence of 22
- assigning 21
- name, supported characters 9
- TIMESTAMP in 7

KEYCOLS option, TABLE or MAP 22

L

large objects, limitations on 6
libraries, Visual C++ 4
LIMITROWS option, DBOPTIONS 22
log chain, retaining 5
Log Reader Agent job
- creating 26
- disabling 27

logging, configuring 24
logging,supplemental 25
logs
- configuring 24
- non-native backups 9
- supplemental 25

M

Manager
- as Windows service 14
- multiple on same system 14
- name, customizing 14

MANAGESECONDARYTRUNCATIONPOINT option 27
MANUALSTART Windows service option 15
Memory requirements for Oracle GoldenGate 1
MGRSERVNAME parameter 14
Microsoft Cumulative Update Package 6 (CU6) 5
Microsoft Visual C ++ 2005 SP1 Redistributable Package 4
multi-byte characters 7

N

name
- non-supported characters in 11
- supported characters in 10

network configuration 3
NOMANAGESECONDARYTRUNCATIONPOINT option 28
NOT FOR REPLICATION, enabling 19
NTEXT data type 7

O

objects, supported 7
ODBC connection
- about 18
- configuring 20

ODBCAD32.EXE client 20
OLE DB connections, using 18
operating systems supported 1
operations, supported 7
Oracle GoldenGate
- installed programs and files 36
- installing 12
- uninstalling 30

P

PASSWORD Windows service option 15
platforms, supported 1
ports, required by Oracle GoldenGate 4
pre-installation instructions 1
privileges
- database 6
- operating system 4

R

recovery model, setting 24
removing
- Oracle GoldenGate 30
- supplemental logging 30

replication components
- configuring 33
- requirements for 5

replication publication, creating 26
replication user, Replicat as 19
replication, disabling 30
rows, size supported 7

S

secondary truncation point, managing 27

spaces

in folder names 13

in object and column names 11

SQL Server Authentication, using 6

SQL Server requirements 4

SQL_Variant data type 7

storage requirements 3

supplemental logging

disabling 30

enabling 25

Supported editions of SQL Server 1

sys.sp_extended_logging stored procedure 25

sysobjects table 25

system requirements 1

system, configuring 18

T

tables

preparing for processing 21

supported kinds 7

TCP/IP, configuring 3

TEXT data type 7

TextCopy utility 8

TIMESTAMP data type 7

trails, storage for 3

TRANLOGOPTIONS parameter 26, 27

transaction log

backing up 24

backup, retaining 25

transaction retention, configuring 34

triggers, disabling on target 21

truncation point, managing 27

U

uninstalling Oracle GoldenGate 30

UPDATETEXT statement 8

USER Windows service option 15

USERID parameter 6

V

VAMSERV program 37

VARCHAR (MAX) data type 7

virtual machine, support for 4

Visual C ++ 2005 SP1 Redistributable Package 4

W

Windows Authentication, using 6

Windows, installing on 13

WRITETEXT statement 8