

Oracle® GoldenGate
SQL Server Installation and Setup Guide
11g Release 1 (11.1.1)
E17805-01

August 2010

ORACLE®

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

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

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

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

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

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

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

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

Contents

.....

Chapter 1	System requirements and preinstallation instructions	3
	Overview of Oracle GoldenGate for SQL Server.....	3
	Supported platforms.....	3
	Operating system requirements.....	3
	Database requirements.....	6
	Supported data types.....	8
	Non-supported data types	9
	Supported objects and operations.....	9
	Non-supported objects and operations.....	10
	Supported and non-supported object names and case.....	10
	Object names and owners	11
	Case sensitivity.....	11
	Supported characters.....	11
	Non-supported characters.....	12
Chapter 2	Installing Oracle GoldenGate	14
	Installation overview.....	14
	Upgrades	14
	New installations.....	14
	Downloading Oracle GoldenGate.....	14
	Installing Oracle GoldenGate on Windows and Windows Cluster	15
	Installing Oracle GoldenGate into a Windows Cluster	15
	Installing the Oracle GoldenGate files.....	15
	Specifying a custom Manager name	16
	Installing Manager as a Windows service	16
	Adding Oracle GoldenGate as a Windows cluster resource.....	18
	Configuring Manager and other processes	19
Chapter 3	Preparing the system for Oracle GoldenGate	20
	Choosing Replicat database connection options.....	20
	Using ODBC or default OLE DB	20
	Using OLE DB with NOT FOR REPLICATION.....	20

.....

Configuring an ODBC connection.....	21
Preparing tables for processing	22
Assigning row identifiers	23
Constraining updates when a table has no key.....	24
Disabling triggers and cascade delete constraints.....	24
Preparing IDENTITY columns for replication.....	24
Suppressing DDL change warnings	25
Configuring the replication components (SQL Server 2005 without CU6 or higher)...	25
What the replication components do for Oracle GoldenGate.....	25
Installing the replication components	26
Configuring and cleaning up the Distributor database.....	26
Enabling extended logging	27
Maintaining the secondary truncation point (SQL Server 2005)	29
SQL Server replication maintains the truncation point.....	29
Oracle GoldenGate maintains the truncation point	29
Managing the CDC capture jobs (SQL Server 2008)	30
Oracle GoldenGate maintains the secondary truncation point	30
SQL Server maintains the secondary truncation point	31
Specifying the location of the backups	31
Sizing and retaining the logs.....	31
Configuring SQL Server logging	32
Chapter 4 Uninstalling Oracle GoldenGate	33
Uninstalling Oracle GoldenGate from Windows (non-cluster)	33
Uninstalling Oracle GoldenGate from Windows Cluster.....	34
Appendix 1 Oracle GoldenGate installed components	35
Oracle GoldenGate Programs and Utilities.....	35
Oracle GoldenGate subdirectories	37
Other Oracle GoldenGate files	39
Oracle GoldenGate checkpoint table	42
Index.....	43

CHAPTER 1

System requirements and preinstallation instructions

.....

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 platforms

SQL Server 2000, 2005, and 2008 on Microsoft-supported operating systems as a source or target.

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

An email and password are required to enter this site.

Operating system requirements

Memory requirements

The amount of memory that is required for Oracle GoldenGate depends on the number of concurrent processes that will be running. At minimum, there is a primary Extract process that captures source data, a secondary Extract data-pump process that transfers data across the network, and one or more Replicat processes that apply the replicated data to the target.

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

The Oracle GoldenGate cache manager takes advantage of the memory management functions of the operating system to ensure that Oracle GoldenGate processes work in a

.....

sustained and efficient manner. Within its cache, it makes use of modern virtual memory techniques by:

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

The actual amount of physical memory that is used by any Oracle GoldenGate process is controlled by the operating system, not the Oracle GoldenGate program.

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

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

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

Disk requirements

Assign the following free disk space:

- 50-150 MB, depending on the database and platform. This includes space for the compressed download file and space for the uncompressed files. You can delete the download file after the installation is complete.
- 40 MB for the working directories and binaries for each instance of Oracle GoldenGate that you are installing on the system. For example, to install two builds of Oracle GoldenGate into two separate directories, allocate 80 MB of space.
- To install Oracle GoldenGate into a cluster environment, install the Oracle GoldenGate binaries and files on a shared file system that is available to all cluster nodes.
- Additional disk space on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. The space that is consumed by the trails varies, depending on the volume of data that will be processed. A good starting point is 1 GB. See the following guidelines.

Storage for Oracle GoldenGate trails

To prevent trail activity from interfering with business applications, assign a separate disk or file system to contain the trail files. These files are created during processing to store all

of the data that is captured by Oracle GoldenGate. The default size is 10 megabytes, but can be changed during the configuration process. Trail files accumulate but can be purged according to rules set with the PURGEOLDEXTRACTS parameter.

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

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

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

To estimate required trail space

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

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

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

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

TCP/IP

- Configure the system to use TCP/IP services, including DNS.
- Configure the network with the host names or IP addresses of all systems that will be hosting Oracle GoldenGate processes and to which Oracle GoldenGate will be connecting. Host names are easier to use.
- Oracle GoldenGate requires the following unreserved and unrestricted TCP/IP ports:
 - One port for communication between the Manager process and other Oracle GoldenGate processes.
 - A range of ports for local Oracle GoldenGate communications: can be the default range starting at port 7840 or a customized range of up to 256 other ports.
- Keep a record of the ports you assigned to Oracle GoldenGate. You will specify them with parameters when configuring the Manager process.
- Configure your firewalls to accept connections through the Oracle GoldenGate ports.

Operating system permissions

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.
- select user rights assignments on the system.

The programs that capture and replicate data (Extract and Replicat) run under the Manager account; therefore run Manager as a member of the local Administrators group (on all nodes in a cluster) if possible.

Extract and Replicat: See “Database user”.

Microsoft-supplied components

- 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>.

Virtual machines

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

Database requirements**Database connection**

Oracle GoldenGate can use either ODBC or ODBC with 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 page 21.
- **OLEDB:** 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). The correct driver must be installed on the system where Replicat is running (see “Database drivers”). The file name of the OLE DB driver is SQLNCLI<version>.DLL (known as the Native Client). For more information about Replicat connection options, see page 20.

Database drivers

The following drivers are required:

SQL Server 2000:

- OLE DB: SQL Server driver or Native Client 9 or 10 (can be downloaded from the Microsoft website).
- ODBC: SQL Server driver or Native Client 9 or 10.

SQL Server 2005:

- OLE DB: SQL Server driver or Native Client 9 or 10. You can use SQL Server driver, but it will not support the new SQL Server 2005 data types.
- ODBC: SQL Server driver or Native Client 9 or 10. You can use SQL Server driver, but it will not support the new SQL Server 2005 data types.

SQL Server 2008:

- OLE DB: Native Client 10 (installed with SQL Server 2008).
- ODBC: SQL Server driver, or Native Client 10. You can use SQL Server driver or Native Client 9, but it will not support the new SQL Server 2008 data types.

Database configuration

- The SQL Server database must be set to the full recovery model, and at least one full database backup must be done before Oracle GoldenGate processes are started for the first time. For more information, see page 32.
- For SQL Server 2008, 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 page 27.
- You can configure a SQL Server 2005 database for replication in one of these ways:
 - Run Oracle GoldenGate replication without having to install the SQL Server 2005 replication components. To use this configuration, obtain from Microsoft and install the Cumulative Update Package 6 (CU6) for SQL Server 2005 Service Pack 2, or install Service Pack 3, which contains the CU6.
 - Run Oracle GoldenGate in conjunction with the SQL Server 2005 replication components. This configuration can be used if the CU6 patch is not installed. Oracle GoldenGate will fall back on this configuration automatically if CU6 components are not detected. This manual contains instructions for installing and configuring the replication components to support Oracle GoldenGate, including the creation of a Distributor database.

Database user

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 permissions 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 permissions listed in Table 2. If using SQL Server authentication, you will need to specify the user and password with the USERID parameter in the Extract or Replicat parameter file.

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

Oracle GoldenGate Process	Manager permissions if using Local System account	Manager permission 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 a member of the db_owner fixed database role of the target database.

Table 2 Required SQL Server permissions 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 data types

Oracle GoldenGate supports most SQL Server 2000, 2005, and 2008 data types except those listed in “Non-supported 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 23.
- 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 with the 'max text repl size parameter, extend the size. Use sp_configure to view the current value of max text repl size.
- IDENTITY is supported as follows:
 - Fully in a uni-directional configuration when the increment values are configured properly. See “Preparing IDENTITY columns for replication” on page 24.
 - Fully, or with limitations, in a bi-directional configuration, depending on how the Replicat connection is defined. Full support requires OLE DB to be used with NOT FOR REPLICATION enabled for the IDENTITY columns and with Replicat operating as the replication user. See “Using OLE DB with NOT FOR REPLICATION” on page 20.
 - IDENTITY ranges are supported only when using OLE DB with NOT FOR REPLICATION set, and with Replicat operating as the replication user.
- 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.

Non-supported 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

Supported objects and operations

- 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 2005, Oracle GoldenGate supports the maximum number of columns per table that is supported by the database. Oracle GoldenGate supports the maximum column size that is supported by the database. 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.

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

- (SQL Server 2005) Operations that are not supported by SQL Server replication. All of the SQL Server 2005 tables that are in the Oracle GoldenGate configuration will be marked for SQL Server replication when you perform the installation procedures (see page 27). *Refer to SQL Server Books Online* for a complete list of the operations that are limited by enabling SQL Server Replication.
- (SQL Server 2008) CDC-restricted items. Any limitations that apply to Change Data Capture will apply to the tables that are replicated by Oracle GoldenGate.
- Extraction or replication of DDL (data definition language) operations.
- Extraction from, or replication to, views. The underlying tables can be extracted and replicated.
- Operations by the TextCopy utility, WriteText TSQL function, or UpdateText. These programs 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 install Oracle GoldenGate on a system where this log backup technology is in use.
- SQL Server 2008 MERGE operations.
- SQL Server 2008 compressed log backups.
- TDE (Transparent Data Encryption)

Supported and non-supported object names and case

The following will help you verify whether the name of a supported object type 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 names¹

Character	Description
~	Tilde
< >	Greater-than and less-than symbols
/	Forward slash
\	Backward slash

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

Character	Description
!	Exclamation point
@	At symbol
#	Pound symbol
\$	Dollar symbol
%	Percent symbol
^	Caret symbol
()	Open and close parentheses
_	Underscore
-	Dash
+	Plus sign
=	Equal symbol
	Pipe
[]	Begin and end brackets
{}	Begin and end curly brackets (braces)

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

Non-supported characters

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

Table 4 Non-supported characters in object and column names¹

Character	Description
&	Ampersand
*	Asterisk
?	Question mark

Table 4 Non-supported characters in object and column names¹

Character	Description
:	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

.....

Installing Oracle GoldenGate

This chapter contains instructions for installing Oracle GoldenGate for use with SQL Server.

Installation overview

These instructions are for installing Oracle GoldenGate for the first time. Installing Oracle GoldenGate installs all of the components required to run and manage the processing (exclusive of any components required from other vendors, such as drivers or libraries) and it installs the Oracle GoldenGate utilities. The installation process takes a short amount of time.

Upgrades

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

New installations

To install Oracle GoldenGate for the first time, the following steps are required:

- Downloading Oracle GoldenGate
- Installing the Oracle GoldenGate software

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

Downloading Oracle GoldenGate

1. Navigate to <http://edelivery.oracle.com>.
2. On the **Welcome** page:
 - Select your language.
 - Click **Continue**.

.....

3. On the **Export Validation** page:
 - Enter your identification information.
 - Accept the **Trial License Agreement** (even if you have a permanent license).
 - Accept the **Export Restrictions**.
 - Click **Continue**.
4. On the **Media Pack Search** page:
 - Select the **Oracle Fusion Middleware** Product Pack.
 - Select the platform on which you will be installing the software.
 - Click **Go**.
5. In the **Results List**:
 - Select the Oracle GoldenGate Media Pack that you want.
 - Click **Continue**.
6. On the **Download** page:
 - Click **Download** for each component that you want. Follow the automatic download process to transfer the mediapack.zip file to your system.

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

Installing Oracle GoldenGate on Windows and Windows Cluster

These instructions apply to all versions of SQL Server. Additional database preparation will be 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. For the Oracle GoldenGate installation location, choose a drive that is a resource within the same cluster group that contains the database instance.
3. Ensure that this group is owned by the cluster node that you are logging into.
4. Install Oracle GoldenGate according to the following instructions.

Installing the Oracle GoldenGate files

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

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

```
CREATE SUBDIRS
```

5. Issue the following command to exit GGSCI.

```
EXIT
```

Specifying a custom Manager name

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

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

To specify a custom Manager name

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

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

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

Installing Manager as a Windows service

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

To install Manager as a Windows service

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

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

Where: <option> is one of the following:

Table 5 INSTALL options

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

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

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

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

Adding Oracle GoldenGate as a Windows cluster resource

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

1. In the Cluster Administrator, select **File>New>Resource**.
2. In the New Resource dialog box, provide a descriptive name for the Oracle GoldenGate Manager (need not be its actual name). For Resource Type, select Generic Service. For Group, select the group that contains the database instance to which Oracle GoldenGate will connect.
3. Click **Next**.
4. In the Possible Owners dialog box, select the nodes on which Oracle GoldenGate will run.
5. Click **Next**.
6. In the GGS Manager Service Properties dialog box, click the Dependencies tab, and add the following to the Resource dependencies list:
 - The database resource group
 - The disk resource containing the Oracle GoldenGate directory
 - The disk resource containing the database transaction log files
 - The disk resource containing the database transaction log backup files
7. Click **Apply**, then **OK**.
8. In the Generic Service Parameters dialog box, type either the default Manager service name of GGS_MGR or, if applicable, the custom name specified in the GLOBALS file.
9. Click **Next**.
10. Click **Finish** to exit the wizard.
11. In the Cluster Administrator tree, right-click the Manager resource and select Properties.
12. Click the Advanced tab, and deselect Affect the Group. This is a recommendation, but you can configure it as needed for your environment.
13. Click **Apply**.
14. Bring the cluster resource online to verify that it was installed correctly.
15. Take the resource offline again.
16. Move the group to the next node in the cluster. When the group has been successfully moved to the second node, the Manager resource should still be offline.
17. Log onto the second node.
18. Install Oracle GoldenGate Manager as a service on this node by running the **install** program as you did on the previous node. If you created a custom name for Manager in the GLOBALS file, that name will be used.
19. Bring the resource online to verify that it is running correctly on this node.
20. Repeat from step 16 for each additional node in the cluster.

Configuring Manager and other processes

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

CHAPTER 3

Preparing the system for Oracle GoldenGate

.....

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.

Evaluate the advantages and disadvantages of these methods before selecting one to use.

Using ODBC or default OLE DB

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

- To keep IDENTITY columns identical on source and target when using ODBC or default OLE DB, Replicat must create special operations in its transaction to ensure that the seeds are incremented on the target. These steps may reduce delivery performance.
- Steps must be taken to adjust or disable triggers and constraints on the target tables to eliminate the potential for redundant operations. See “Preparing tables for processing” on page 22.

To use either ODBC or ODBC with OLE DB, you must also configure the ODBC data source. See “Configuring an ODBC connection” on page 21.

NOTE OLE DB uses the ODBC connection settings to derive connection information for OLE DB together with information on what 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 will fire on the target for other users. Any tables that are affected by trigger operations should be included in the replication configuration for the target tables to be updated properly.
- Foreign key constraints are not enforced on the target for Replicat transactions. CASCADE updates and deletes are not performed. Referenced tables should be included in the replication configuration with their referencing table so that cascaded operations are captured and replicated. If referenced tables are omitted, there will be 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.
- CHECK constraints are not enforced on the target for Replicat transactions. You should consider any data integrity issues that this can cause, even though the constraints will be enforced on the source before data is captured and replicated.

NOTE Normal IDENTITY, trigger, and constraint behavior will be 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:
 - Foreign Key Constraints
 - Check Constraints
 - IDENTITY columns
 - Triggers (requires textual changes to the definition)
2. In the Replicat parameter file, include the DBOPTIONS parameter with the USEREPLICATIONUSER option.

NOTE In an active-passive bi-directional configuration, the passive database must have NOT FOR REPLICATION set on the appropriate columns, and DBOPTIONS USEREPLICATIONUSER must be specified in the Replicat parameter file. In an active-active environment, both source and target must have those flags enabled.

Configuring an ODBC connection

Extract uses ODBC (Open Database Connectivity) to connect to the source database. By default, Replicat uses OLE DB with ODBC to connect to the target database, but can be configured to use ODBC exclusively if the DBOPTIONS parameter is set to USEODBC.

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

Establish a system data source name (DSN) on each source and target system where Oracle GoldenGate will interface with a SQL Server database. A DSN stores information about how to connect to the database.

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%\Windows\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 ODBC 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.
 - **Which SQL Server do you want to connect to:** Select the SQL Server instance name.
 6. Click **Next**.
 7. For login authentication, select **With Windows NT authentication** if the Oracle GoldenGate process will use a Windows authentication, or select **With SQL Server authentication using a login** if the process will use database credentials. Supply login information as appropriate.
 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 the next system where SQL Server is installed.

Preparing tables for processing

Some action might be required on your part to prepare your tables for capture and delivery by Oracle GoldenGate.

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 prior to Cumulative Update 6:** 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 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, then 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.

Regardless of the SQL Server version, do not use SQL Server native replication concurrently with Oracle GoldenGate replication if Oracle GoldenGate will be replicating tables that do not have a primary key. Oracle GoldenGate will enable those tables for replication, but tables without keys are not supported by SQL Server replication and will cause it to fail.

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 total length of data in rows for target tables without a primary key to be below 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.

How to specify your own key for Oracle GoldenGate to use

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

Constraining updates when a table has no key

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

Disabling triggers and cascade delete constraints

NOTE If you configure `Replicat` to connect to the database through ODBC or the default OLE DB connection (where `Replicat` does not operate as the replication user) follow these instructions. Skip this section 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.

Preparing IDENTITY columns for replication

To configure `IDENTITY` columns correctly for support by Oracle GoldenGate, follow these instructions.

Using IDENTITY columns in a bidirectional configuration

NOTE These instructions only apply if you configured Replicat to use the default connection of OLE DB and Replicat does *not* operate as the replication user, or if Replicat will use the ODBC driver.

For a bidirectional SQL Server configuration, define 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.

Replicating to an IDENTITY column

Because only one table in a session can have IDENTITY_INSERT set to ON, Replicat might need to continuously toggle IDENTITY_INSERT off and on when processing different tables that contain IDENTITY columns. 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.

Suppressing DDL change warnings

Extract logs a warning for any DDL that it detects and, depending on the type of DDL, it may also abend. DDL replication is not supported for SQL Server. If you are willing to accept any data inconsistencies that result from DDL that does not cause Extract to abend, you can suppress the warning to prevent the messages from accumulating in the Oracle GoldenGate log. Use the TRANLOGOPTIONS parameter with the NODDLCHANGEWARNING option.

Configuring the replication components (SQL Server 2005 without CU6 or higher)

If Microsoft Cumulative Update Package 6 (CU6) for SQL Server 2005 Service Pack 2 is not installed on a SQL Server source system, the SQL Server 2005 replication components must be used with a Distributor database to support Oracle GoldenGate replication. Install the SQL Server replication components only if they are not installed and configured already. If replication components and a distributor database already exist, skip to page 27 to enable extended logging.

What the replication components do for Oracle GoldenGate

The replication components enable the following:

- Extract can capture logged SQL Server 2005 UPDATE operations.
- 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 Distributor can be used for all source databases. Oracle GoldenGate does not depend on the Distributor database, but instead reads the logs directly, so you can set transaction retention to zero.

Installing the replication components

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 Distributor, or select a remote Distributor.
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 Distributor database and finish the setup.

Configuring and cleaning up the Distributor database

After the Distributor database is 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 Distributor setups.

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 Distributor 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 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. For the following jobs, right click and then select **Disable**.
 - Agent history clean up: <Distributor database name>
 - Distribution clean up: <Distributor 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**.

Enabling extended logging

These instructions apply to new installations of Oracle GoldenGate for SQL Server 2000, SQL Server 2005, and SQL Server 2008. You will be enabling extended logging with the ADD TRANDATA command so that Extract can capture all of the information that is required to reconstruct SQL operations on the target. This is more information than what SQL Server logs by default. This procedure is required for all tables that will be replicated with Oracle GoldenGate.

ADD TRANDATA 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 extended 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 with the CU6 update package installed:** ADD TRANDATA calls the `sys.sp_extended_logging` stored procedure.
- **SQL Server 2005 without CU6:** 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. Because this job cannot run concurrently with the Extract process, this procedure includes steps for stopping and disabling it.

NOTE The ADD TRANDATA command for SQL Server 2005 automatically filters out tables with `dbo.MS` and `dbo.sys` in their names, because it is assumed that they are system tables which should not be replicated. If you have any 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.

- **SQL Server 2008:** ADD TRANDATA enables Change Data Capture (CDC) and sets up a minimal Change Data Capture on the specified table. Oracle GoldenGate does not use the captured data and will minimize it by using either the primary-key columns or the smallest fixed or variable-length columns. The SQL Server Agent must be running if the database is SQL Server 2008. When enabling CDC, you should be aware of the following:
 - As part of enabling CDC, SQL Server will create two jobs per database: these are named `dbname_capture` and `dbname_cleanup`. The `dbname_capture` job adjusts the secondary truncation point and allows the transaction log space to be reused after parts of it have been backed up. The `dbname_cleanup` job ages and deletes data captured by CDC. These jobs are important to the functioning of CDC and Extract. Please do not delete them.
 - The use of CDC involves the use of extra space in the database. Oracle GoldenGate will try to minimize the amount of data that is captured by CDC. The `dbname_cleanup` job deletes CDC data after three days. You can adjust this retention period by changing `dbname_cleanup` according to the Microsoft documentation.

To enable extended 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 data source.
- USERID <user>, PASSWORD <password> is the Extract login and password, if Extract will use SQL Server authentication. Can be the Extract user or a member of an account in the System Administrators or Server Administrators fixed server role.

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>
```

To disable the Log Reader Agent Job (SQL Server 2005 without CU6 or higher)

1. In SQL Server Management Studio, connect to the SQL Server 2005 instance.
2. Start SQL Server Agent.
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**.

Maintaining the secondary truncation point (SQL Server 2005)

If the Extract process is running against a SQL Server 2005 source, use the `TRANLOGOPTIONS` parameter to control whether Extract or SQL Server maintains the secondary truncation point. This is a required parameter and must contain one of the following options.

SQL Server replication maintains the truncation point

Use `TRANLOGOPTIONS` with the `NOMANAGESECONDARYTRUNCATIONPOINT` option if Extract and SQL Server 2005 replication or a third-party replication program will be running at the same time for the same database. The replication program will manage the secondary truncation point. Extract will read from the log backups if the online log gets truncated before Extract is finished with it. If a replication program will be maintaining the secondary truncation point, make certain that the backups are available to Extract until it has finished processing all of the data in them.

NOTE If you use `TRANLOGOPTIONS MANAGESECONDARYTRUNCATIONPOINT` when Extract and a replication program are running at the same time, it will cause the replication program to fail, because Extract is moving the truncation point. If the replication program tries to read a record from the log and that record is already flagged by Extract as "processed," the replication program cannot capture the data.

Oracle GoldenGate maintains the truncation point

Use `TRANLOGOPTIONS` with the `MANAGESECONDARYTRUNCATIONPOINT` option if Extract and SQL Server 2005 replication or a third-party replication program will not be running at the same time for the same database. Extract will manage the secondary truncation point by moving it forward at a defined interval. Extract does not rely on the secondary truncation point to retain the transaction data because it can read from the log backups if needed.

If Extract is stopped for a long time or is removed while managing the secondary truncation point for a SQL Server 2005 database, the log could grow too large. To release log space, you can truncate the log manually as needed or disable replication, depending on whether you are keeping or removing Extract.

If keeping Extract

If you intend to start Extract again, make certain to retain all of the log backups that contain data that still needs to be processed by Extract.

1. Use the INFO EXTRACT command to find out the log-read checkpoint, which shows the oldest log that Extract needs.
2. To truncate the log, use the following command as often as needed to free up space.

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

If removing Extract

If you intend to remove Extract, you can disable replication instead of manually truncating the logs.

1. Run the DELETE TRANDATA command for every table that Extract was capturing. This command stops the extended logging of the Article for the affected tables from the Publication, and then removes it.
2. Do either of the following:
 - Use the sp_dboption stored procedure as follows.

```
EXEC sp_dboption '<Database Name>', 'Publish', 'false'
```
 - In the SQL Server Management Studio, delete the publication for the database from which Extract was capturing.

Managing the CDC capture jobs (SQL Server 2008)

To capture from a SQL Server 2008 source, Oracle GoldenGate uses the SQL Server Change Data Capture to enable supplemental logging for replicated tables. Use the TRANLOGOPTIONS parameter with either the MANAGESECONDARYTRUNCATIONPOINT or the NOMANAGESECONDARYTRUNCATIONPOINT option to manage the change data that the capture jobs generate together with the CDC jobs.

Oracle GoldenGate maintains the secondary truncation point

When MANAGESECONDARYTRUNCATIONPOINT is used for at least one Extract group in the database, the Extract process will disable the CDC capture jobs and manage the secondary truncation point. This will prevent change data from being collected for all of the tables in the database, resulting in:

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

SQL Server maintains the secondary truncation point

If you need to use CDC for your applications, use `TRANLOGOPTIONS` with `NOMANAGESECONDARYTRUNCATIONPOINT` for all Extract groups in the database to prevent Extract from deleting the CDC jobs. When CDC jobs are not deleted, they can degrade the performance of SQL Server. CDC will require extra storage space to store the captured data in the CDC tables, and it will generate extra transaction log records. SQL Server will manage the secondary truncation point.

Specifying the location of the backups

The log backups can be in their default locations or in a non-default location. By default, the Extract process reads the log backups if the online logs do not contain the records that need to be processed. This can happen if Extract lags too far behind in the transaction stream, and the transactions get backed up and moved offline before they can be processed.

By default, Extract looks for the log backups in the SQL Server default location. If you store the backups in a different location from the default, you must specify that location to Extract by using `TRANLOGOPTIONS` with the `ALTARCHIVELOGDEST` option in the Extract parameter file. For syntax and usage, see the Oracle GoldenGate *Windows and UNIX Reference Guide*.

Sizing and retaining the logs

Retain enough log data 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 log that contains the start of the oldest uncommitted unit of work, and all logs thereafter.

If data that Extract needs during processing was not retained, either in online logs or 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*.

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

If the source SQL Server 2005 database does not have CU6 and the Extract process needs to be suspended for a longer time than the normal log backup frequency, re-enable and start the SQL Server Replication Log Reader job temporarily to manage the last distributed transaction. Stop and disable the job before you restart Extract.

Configuring SQL Server logging

To support Oracle GoldenGate on a SQL Server source system, do the following:

- Set the database to the full recovery model. Oracle GoldenGate requires that log truncation and non-logged bulk copy are turned off.
- Make at least one full database backup before starting Oracle GoldenGate processes for the first time.

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**.
6. Click **OK**.

To make a full backup of the database

1. Right click the database name, and then select **All Tasks > Backup Database**.
2. Select **Database - Complete**. This option makes a full database backup and ensures that no transaction information is lost when Oracle GoldenGate starts.
3. Under **Destination**, click **Add** to specify the backup file name and location.
4. Click **OK**. The backup file is added to the **Destination** list box in the **SQL Server Backup** dialog box.
5. Click **OK** to start the backup.

CHAPTER 4

Uninstalling Oracle GoldenGate

.....

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

NOTE If you uninstall Oracle GoldenGate after Extract has been managing the secondary truncation point, the logs could fill up just as they would when Extract is stopped. To prevent this, you can remove and disable replication, or you can manually truncate the logs by running the `sp_repldone` script.

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 `GGMSG.DLL` files from the Windows `SYSTEM32` folder.
10. Delete the Oracle GoldenGate installation folder.
11. Drop any Oracle GoldenGate-related objects from the database as needed.

Uninstalling Oracle GoldenGate from Windows Cluster

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

```
install deleteevents deleteservice
```

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

7. Delete the CATEGORY.DLL and GGSMSG.DLL files from the Windows SYSTEM32 folder.
8. Move the cluster group to the next node in the cluster, and repeat from step 4.
9. Delete the Oracle GoldenGate installation folder.
10. Drop any Oracle GoldenGate-related objects from the database as needed.

APPENDIX 1

Oracle GoldenGate installed components



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

Oracle GoldenGate Programs and Utilities

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

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

Table 6 Programs and utilities

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



Table 6 Programs and utilities (continued)

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

Oracle GoldenGate subdirectories

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

NOTE Some directories may not exist in all installations.

Table 7 Subdirectories

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

Table 7 Subdirectories (continued)

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

Other Oracle GoldenGate files

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

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

Table 8 Other files

Component	Description
bcpfmt.tpl	Template for use with Replicat when creating a run file for the Microsoft BCP/DTS bulk-load utility.
blowfish.txt	Blowfish encryption software license agreement.
category.dll	Windows dynamic link library used by the INSTALL program.
chkpt_<db>_create.sql	Script that creates a checkpoint table in the local database. A different script is installed for each database type.
db2cntl.tpl	Template for use with Replicat when creating a control file for the IBM LOADUTIL bulk-load utility.
ddl_cleartrace.sql	Script that removes the DDL trace file. (Oracle installations)
ddl_disable.sql	Script that disables the Oracle GoldenGate DDL trigger. (Oracle installations)
ddl_enable.sql	Script that enables the Oracle GoldenGate DDL trigger. (Oracle installations)
ddl_ora9.sql	A script that gets tablespace information from an Oracle 9 database.
ddl_ora10.sql	A script that disables the Oracle recyclebin and gets tablespace information from an Oracle 10 database.
ddl_pin.sql	Script that pins DDL tracing, the DDL package, and the DDL trigger for performance improvements. (Oracle installations)
ddl_remove.sql	Script that removes the DDL extraction trigger and package. (Oracle installations)
ddl_setup.sql	Script that installs the Oracle GoldenGate DDL extraction and replication objects. (Oracle installations)
ddl_status.sql	Script that verifies whether or not each object created by the Oracle GoldenGate DDL support feature exists and is functioning properly. (Oracle installations)

Table 8 Other files (continued)

Component	Description
ddl_tracelevel.sql	Script that sets the level of tracing for the DDL support feature. (Oracle installations)
debug files	Debug text files that may be present if tracing was turned on.
demo_<db>_create.sql	Script that creates demonstration tables in the database associated with the Oracle GoldenGate installation.
demo_<db>_insert.sql	Script that inserts initial test data into the demonstration tables.
demo_<db>_misc.sql	Script that simulates transaction activity on the demonstration tables.
ENCKEYS	User-created file that stores encryption keys. Written in external ASCII format.
exitdemo.c	User exit example.
ggmessage.dat	Data file that contains error, informational, and warning messages that are returned by the Oracle GoldenGate processes. The version of this file is checked upon process startup and must be identical to that of the process in order for the process to operate.
ggserr.log	File that logs processing events, messages, errors, and warnings generated by Oracle GoldenGate.
ggsmsg.dll	Windows dynamic link library used by the INSTALL program.
GLOBALS	User-created file that stores parameters applying to the Oracle GoldenGate instance as a whole.
help.txt	Help file for the GGSCI command interface.
LGPL.txt	Lesser General Public License statement. Applies to free libraries from the Free Software Foundation.
libxml2.dll	Windows dynamic link library containing the XML library for the Oracle GoldenGate XML procedures.
libxml2.txt	License agreement for libxml2.dll.
marker.hist	File created by Replicat if markers were passed from a NonStop source system.
marker_remove.sql	Script that removes the DDL marker table. (Oracle installations)

Table 8 Other files (continued)

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

Oracle GoldenGate checkpoint table

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

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

Table 9 Checkpoint table definitions

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

Index

.....

A

- ADD TRANDATA command** 27
- ADDEVENTS Windows service option** 17
- ADDSERVICE Windows service option** 17
- ALTARCHIVELOGDEST option, TRANLOGOPTIONS** 31
- AUTOSTART Windows service option** 17

B

- backup logs, see logs**
- BATCHSQL parameter** 25
- before images, logging** 27

C

- cascade deletes, disabling** 24
- case, supported** 10
- category.dll** 17
- Change Data Capture** 28
- Change Data Capture (CDC)**
 - capture jobs, maintaining 30
 - enabling 28
- characters**
 - multi-byte 8
 - supported in object names 11
- cluster, installing on** 4, 15, 18
- columns**
 - datatypes supported 8
 - number of supported 9
- components, Oracle GoldenGate** 35
- connections, to database** 7
- constraints, integrity** 24

D

- data source name (DSN), creating** 21
- data types, supported** 8
- database**
 - Distributor, SQL Server 2005 26
 - requirements 6
 - user, Oracle GoldenGate 7
- deletes, cascaded** 24
- disk requirements** 4
- Distributor database, configuring** 26

E

- extended logging, enabling** 27

F

- files, installed by Oracle GoldenGate** 35
- firewall, configuring** 5

G

- ggmessage.dat file** 40
- GGSMGR default Manager name** 16
- ggmsg.dll** 17
- GLOBALS file** 16

I

- IDENTITY columns** 24
- IMAGE data type** 9
- INFO EXTRACT command** 31
- installing Oracle GoldenGate** 14

K

key

- assigning 23
- name, supported characters 11
- tables without 24
- TIMESTAMP in 8

KEYCOLS option, TABLE or MAP 24

L

large objects, limitations on 8

libraries, Visual C++ 6

LIMITROWS option, DBOPTIONS 24

logging, configuring 32

logging, extended 27

logs

- configuring 22, 32
- extended 27
- non-native backups 10

M

Manager

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

MANUALSTART Windows service option 17

Memory requirements for Oracle GoldenGate 3

MGRSERVNAME parameter 16

Microsoft Visual C++ 2005 SP1 Redistributable Package 6

multi-byte characters 8

N

name

- non-supported characters in 12
- supported characters in 11

names, supported 10

NOT FOR REPLICATION, enabling 20

NTEXT data type 9

O

objects, supported 9

ODBC connection

- about 20
- configuring 21

ODBCAD32.EXE client 21, 22

OLE DB connections, using 20

online logs, see logs

operating system requirements 3

operations, supported 9

Oracle GoldenGate

- installed programs and files 35
- installing 14
- uninstalling 33

P

PASSWORD Windows service option 17

permissions, database 7

platforms, supported 3

ports, required by Oracle GoldenGate 5

R

recovery model, setting 32

replication user, Replicat as 20

rows, size supported 9

S

secondary truncation point, maintaining 29, 30

spaces

- in folder names 15
- in object and column names 13

SQL Server Authentication, using 7

SQL_Variant data type 9

T

tables

- preparing for processing 22
- supported kinds 9

TCP/IP, configuring 5

TEXT data type 9

TextCopy utility 10

TIMESTAMP data type 8

TRANLOGOPTIONS parameter 29

transaction retention, configuring 27
triggers, disabling on target 24

U

uninstalling Oracle GoldenGate 33
UpdateText 10
USER Windows service option 17
user, database 7
USERID parameter 8

V

VAMSERV program 36
VARCHAR (MAX) data type 9
virtual machine, support for 6
Visual C ++ 2005 SP1 Redistributable Package 6

W

Windows Authentication, using 7
WriteText TSQL 10