# Oracle® Fusion Middleware

Installing and Configuring Oracle GoldenGate for SQL Server

12*c* (12.2.0.1)

**E66360-04**

February 2017

This document describes how to install and set up Oracle GoldenGate for SQL Server.

Oracle Fusion Middleware Installing and Configuring Oracle GoldenGate for SQL Server , 12*c* (12.2.0.1)

E66360-04

# Contents

## 3 Preparing the System for Oracle GoldenGate

## 4 Preparing the Transaction Logs for Oracle GoldenGate

## 5 Uninstalling Oracle GoldenGate

## A Oracle GoldenGate Installed Components

## B Requirements Summary for Extract/Capture in Archived Log Only (ALO) Mode

## C Oracle GoldenGate 12.2 for SQL Server - Standard Edition Capture

# Preface

With Oracle GoldenGate for Microsoft SQL Server, you can capture transactional data from user tables of supported SQL Server versions and replicate the data to a SQL Server database or other supported Oracle GoldenGate targets, such as an Oracle Database or Big Data target. Oracle GoldenGate for SQL Server supports data filtering, mapping, and transformations unless noted otherwise in this documentation.

This guide helps you get started with installing Oracle GoldenGate for SQL Server and performing the initial setup. Refer to the other Oracle GoldenGate documentation listed in this Preface for additional information to configure, run, and manage your Oracle GoldenGate environment.

## Audience

This guide is intended for installers, database administrators, and system administrators who are installing, configuring and running Oracle GoldenGate.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

## Related Documents

The Oracle GoldenGate documentation set includes the following components:

### Windows, UNIX, and Linux Platforms

- *Installing and Configuring Oracle GoldenGate for DB2 for i*
- *Installing and Configuring Oracle GoldenGate for DB2 LUW*
- *Installing and Configuring Oracle GoldenGate for DB2 z/OS*
- *Installing and Configuring Oracle GoldenGate for Informix*
- *Installing and Configuring Oracle GoldenGate for MySQL*

- *Installing and Configuring Oracle GoldenGate for NonStop SQL/MX*
- *Installing and Configuring Oracle GoldenGate for SQL Server*
- *Installing and Configuring Oracle GoldenGate for Oracle TimesTen*
- *Installing and Configuring Oracle GoldenGate for Oracle Database*
- *Installing and Configuring Oracle GoldenGate for Sybase*
- *Installing and Configuring Oracle GoldenGate for Teradata*
- *Administering Oracle GoldenGate for Windows and UNIX*
- *Reference for Oracle GoldenGate for Windows and UNIX*
- *Logdump Reference for Oracle GoldenGate*
- *Upgrading Oracle GoldenGate for Windows and UNIX*
- *Error Messages Reference for Oracle GoldenGate for Windows and UNIX*

## Conventions

The following text conventions are used in this document:

| Convention | Meaning |
|---|---|
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, such as "From the File menu, select **Save**." Boldface also is used for terms defined in text or in the glossary. |
| *italic*<br>`*italic*` | Italic type indicates placeholder variables for which you supply particular values, such as in the parameter statement: `TABLE` *`table_name`*. Italic type also is used for book titles and emphasis. |
| `monospace`<br>`MONOSPACE` | Monospace type indicates code components such as user exits and scripts; the names of files and database objects; URL paths; and input and output text that appears on the screen. Uppercase monospace type is generally used to represent the names of Oracle GoldenGate parameters, commands, and user-configurable functions, as well as SQL commands and keywords. |
| UPPERCASE | Uppercase in the regular text font indicates the name of a utility unless the name is intended to be a specific case. |
| { } | Braces within syntax enclose a set of options that are separated by pipe symbols, one of which must be selected, for example: `{`*`option1`*` \| `*`option2`*` \| `*`option3`*`}`. |
| [ ] | Brackets within syntax indicate an optional element. For example in this syntax, the `SAVE` clause is optional: `CLEANUP REPLICAT` *`group_name`* `[, SAVE `*`count`*`]`. Multiple options within an optional element are separated by a pipe symbol, for example: `[`*`option1`*` \| `*`option2`*`]`. |

# 1 System Requirements and Preinstallation Instructions

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

This chapter includes the following sections:

- Section 1.1, "Verifying Certification and System Requirements"
- Section 1.2, "Operating System Requirements"
- Section 1.3, "SQL Server Requirements"
- Section 1.4, "Supported SQL Server Data Types"
- Section 1.5, "Non-Supported SQL Server Data Types"
- Section 1.6, "Supported Objects and Operations for SQL Server"
- Section 1.7, "Non-Supported Objects and Operations for SQL Server"

## 1.1 Verifying Certification and System Requirements

Make sure that you are installing your product on a supported hardware and operating system configurations. For more information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

Oracle has tested and verified the performance of your product on all certified systems and environments; whenever new certifications occur, they are added to the proper certification document right away. New certifications can occur at any time, and for this reason the certification documents are kept outside of the documentation libraries and are available on Oracle Technology Network.

## 1.2 Operating System Requirements

This section describes the operating system requirements of Oracle GoldenGate. These requirements fall into one of the following categories:

- Section 1.2.1, "Memory Requirements"
- Section 1.2.2, "Disk Requirements"
- Section 1.2.3, "Temporary Disk Requirements"
- Section 1.2.4, "Network"
- Section 1.2.5, "Operating System Privileges"
- Section 1.2.6, "Other Programs and Settings"

### 1.2.1 Memory Requirements

The amount of memory that is required for Oracle GoldenGate depends on the amount of data being processed, the number of Oracle GoldenGate processes running, the amount of RAM available to Oracle GoldenGate, and the amount of disk space that is available to Oracle GoldenGate for storing pages of RAM temporarily on disk when the operating system needs to free up RAM (typically when a low watermark is reached). This temporary storage of RAM to disk is commonly known as *swapping* or *paging* (herein referred to as *swapping*). Depending on the platform, the term *swap space* can be a swap partition, a swap file, a page file (Windows) or a shared memory segment (IBM i platforms).

Modern servers have sufficient RAM combined with sufficient swap space and memory management systems to run Oracle GoldenGate. However, increasing the amount of RAM available to Oracle GoldenGate may significantly improve its performance, as well as that of the system in general.

Typical Oracle GoldenGate installations provide RAM in multiples of gigabytes to prevent excessive swapping of RAM pages to disk. The more contention there is for RAM the more swap space that is used.

Excessive swapping to disk causes performance issues for the Extract process in particular, because it must store data from each open transaction until a commit record is received. If Oracle GoldenGate runs on the same system as the database, the amount of RAM that is available becomes critical to the performance of both.

RAM and swap usage are controlled by the operating system, not the Oracle GoldenGate processes. The Oracle GoldenGate cache manager takes advantage of the memory management functions of the operating system to ensure that the Oracle GoldenGate processes work in a sustained and efficient manner. In most cases, users need not change the default Oracle GoldenGate memory management configuration.

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

### 1.2.2 Disk Requirements

Assign free disk space according to the following instructions:

- To determine the size of the Oracle GoldenGate download file, view the Size column before downloading your selected build from Oracle Software Delivery Cloud. The value shown is the size of the files in compressed form. The size of the expanded Oracle GoldenGate installation directory will be significantly larger on disk. For more information, see Section 2.2, "Understanding and Obtaining the Oracle GoldenGate Distribution."
- To install Oracle GoldenGate into a cluster environment, install the Oracle GoldenGate binaries and files as the Oracle user on a shared file system that is available to all cluster nodes. See Section 2.3, "Preparing to Install Oracle GoldenGate Within a Cluster" for more information.
- An additional 1 GB of disk space on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. You may need more or less than this amount, because the space that is consumed by the trails depends on the volume of data that will be processed. See the guidelines for sizing trails in *Administering Oracle GoldenGate for Windows and UNIX*.

### 1.2.3 Temporary Disk Requirements

By default, Oracle GoldenGate maintains data that it swaps to disk in the `dirtmp` sub-directory of the Oracle GoldenGate installation directory. The cache manager assumes that all of the free space on the file system is available. This directory can fill up quickly if there is a large transaction volume with large transaction sizes. To prevent I/O contention and possible disk-related Extract failures, dedicate a disk to this directory. You can assign a name and size to this directory with the `CACHEDIRECTORY` option of the `CACHEMGR` parameter. The `CACHESIZE` option of `CACHEMGR` sets a soft limit for the amount of virtual memory (cache size) that is available for caching transaction data. See *Reference for Oracle GoldenGate for Windows and UNIX* for the default values of these options and detailed explanations, in case system adjustments need to be made.

### 1.2.4 Network

Configure networking according to the following instructions:

- Configure the system to use TCP/IP services, including DNS. Oracle GoldenGate supports IPv4 and IPv6 and can operate in a system that supports one or both of these protocols.
- Configure the network with the host names or IP addresses of all systems that will be hosting Oracle GoldenGate processes and to which Oracle GoldenGate will be connecting. Host names are easier to use.
- Oracle GoldenGate requires some unreserved and unrestricted TCP/IP ports, the number of which depends on the number and types of processes in your configuration. See the *Administering Oracle GoldenGate for Windows and UNIX* for details on how to configure the Manager process to handle the required ports.
- Keep a record of the ports that you assigned to Oracle GoldenGate. You will specify them with parameters when configuring the Manager process.
- Configure your firewalls to accept connections through the Oracle GoldenGate ports.

### 1.2.5 Operating System Privileges

Assign operating system privileges according to the following instructions.

#### 1.2.5.1 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 permissions over the files and folders within the Oracle GoldenGate directories.
- Full control permissions over the trail files, if stored in a location other than the Oracle GoldenGate directory.
- For a source capture installation of Oracle GoldenGate, Manager requires Read permissions on the SQL Server database transaction log files and transaction log backups.
- If running a source capture in Archived Log Mode from a middle tier Windows server, Manager requires Read permissions to the network share where the transaction log backups are written, and Read permissions on the transaction log backups.

- Membership in the server's 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 the Manager's privileges.

#### 1.2.5.2 Extract and Replicat

See Section 1.3.4, "Database User for Oracle GoldenGate Processes."

### 1.2.6 Other Programs and Settings

Observe the following other program and settings information for Oracle GoldenGate for SQL Server:

- To install capture on a remote Windows server, for archived log mode, set the remote server's time and time zone to that of the database server.
- Before installing Oracle GoldenGate on a Windows system, install and configure the Microsoft Visual C ++ 2010 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`.
- To capture from a source SQL Server Standard Edition database, the SQL Server Replication features must be installed.
- SQL Server Client Tools Connectivity features must be installed on the server where Oracle GoldenGate is to be installed. This feature is normally installed by default when installing an instance of SQL Server, but for a Windows server that is to be used for a remote Replicat or for an Extract running off the database server in an Archived Log Mode, the required client connectivity drivers can be obtained through the SQL Server installation media or from the following links:

  Microsoft SQL Server 2008 SP4 Feature Pack:
  `https://www.microsoft.com/en-us/download/details.aspx?id=44277`

  Microsoft SQL Server 2008 R2 SP3 Feature Pack:
  `https://www.microsoft.com/en-us/download/details.aspx?id=44272`

  Microsoft SQL Server 2012 Feature Pack:
  `https://www.microsoft.com/en-us/download/details.aspx?id=29065`

  Microsoft SQL Server 2014 Feature Pack:
  `https://www.microsoft.com/en-us/download/details.aspx?id=42295`

- 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.

## 1.3 SQL Server Requirements

To operate with Microsoft SQL Server databases, Oracle GoldenGate requires the following setup in the database instance.

- Section 1.3.1, "Instance Configuration"
- Section 1.3.2, "Database Configuration"
- Section 1.3.3, "Database Connection"

- Section 1.3.4, "Database User for Oracle GoldenGate Processes"
- Section 1.3.5, "Encrypting and Storing User Credentials"

## 1.3.1 Instance Configuration

To configure an instance, the following must be true:

- The SQL Server server name (`@@SERVERNAME`) should not be NULL.
- The SQL Server instance must be either Standard or Enterprise Edition.
- To capture from a source SQL Server Standard Edition database, the SQL Server Replication features must be installed, a Distributor configured, and a distribution database created. See Section 4.4, "Enabling Supplemental Logging" for more information.

## 1.3.2 Database Configuration

The database should be configured according to the following requirements:

- Only user databases are supported for capture and delivery.
- The database must be set to the compatibility level of the SQL Server instance version.
- Source databases must be set to the Full recovery model.
- 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.
- The log chain on the source database must not be broken at any time while Oracle GoldenGate is installed and running. 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. For more information, see the Microsoft SQL Server documentation on Log Chains.

**Limitations:**

- Oracle GoldenGate does not support system databases or capture from Contained databases.
- Source database names should not exceed 121 characters due to a limitation in the SQL Server stored procedures that are used to enable supplemental logging. If configuring the Oracle GoldenGate heartbeat functionality, the SQL Server database name should not exceed 107 characters.
- Capture from SQL Server 2014 databases enabled with In-Memory OLTP (In-Memory Optimization) is not supported. When you add a Memory Optimized Data file group to your database, Oracle GoldenGate is not allowed to enable supplemental logging for any table in the database, and conversely, if supplemental logging has been enabled for any table in the database prior to the creation of a Memory Optimized Data file group, SQL Server will not allow a Memory Optimized Data file group to be created.
- SQL Server 2014 provides the option of Delayed Transaction Durability for transactions, however, transactions against a table configured with supplemental logging (`TRANDATA`) by Oracle GoldenGate will always be Fully Durable.

- The source database must not be configured with TDE (Transparent Data Encryption).

## 1.3.3 Database Connection

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

- ODBC: The Extract process uses ODBC to connect to a source SQL Server database to obtain metadata and perform other process queries. The Replicat process uses ODBC to connect to a target SQL Server database to obtain metadata, but can optionally use it for its delivery of transactions as well. ODBC must be properly configured. For more information, see Section 3.1.3, "Configuring an ODBC Connection."
- 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 DML). For more information about Replicat connection options, see Section 3.1, "Configuring a Database Connection."

**Limitations of Support**

- Use at least the SQL Server Native Client 10.0 driver. The older SQL Server driver (`SQLSRV32.DLL`) does not support newer SQL Server data types.
- Using the SQL Server Native Client 11 OLE DB driver to connect to a SQL Server 2012 or a SQL Server 2014 instance in OLEDB mode may lead to a memory leak issue (Microsoft article 2881661). Microsoft has provided a fix in SQL Server 2012 SP1 CU7 (Microsoft article 2894115) and SQL Server 2014 CU1 (Microsoft article 2931693). To avoid a possible memory leak, you may choose one of the following options:
  - For SQL Server 2012, upgrade the SQL Native Client 11.0 driver to the SP1 CU7 level.
  - For SQL Server 2014, a possible memory leak still may exist after installing SQL Server 2014 CU1 on a new Windows system. This does not occur when you upgrade from SQL Server 2012 SP1 CU7 to SQL Server 2014 CU1.
  - Use ODBC mode
- For SQL Server 2014, only the SQL Server Native Client 11.0 driver is supported. The ODBC Driver 11 for SQL Server is not supported.

## 1.3.4 Database User for Oracle GoldenGate Processes

The following database users and privileges are required for Oracle GoldenGate to capture from, and apply to, a Microsoft SQL Server database.

### 1.3.4.1 User that Enables Supplemental Logging

A database user is required to issue the `ADD TRANDATA` command to enable supplemental logging on the source database tables in the Oracle GoldenGate configuration. A database login command (`DBLOGIN`) is issued from GGSCI before issuing `ADD TRANDATA`. This user must be a member of the SQL Server System Administrators (`sysadmin`) role. For more information, see Section 4.4, "Enabling Supplemental Logging."

### 1.3.4.2 Extract (Capture) and Replicat (Apply) Users

The Oracle GoldenGate Extract process captures data from a source SQL Server database for initial loads and from the transaction log for change data capture, and the Replicat process applies it to a target SQL Server database. These 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 1–1 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 1–2. 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 or, alternatively, use the Oracle GoldenGate credential store and specify a user alias with the `USERIDALIAS` parameter. For more information about these parameters, see *Administering Oracle GoldenGate for Windows and UNIX*.

***Table 1–1 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<br>(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<br>(target sdystem) | `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 1–2 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. |

## 1.3.5 Encrypting and Storing User Credentials

If using SQL Server authentication rather than Windows authentication, you will at times during installation and setup of Oracle GoldenGate need to log into the database by using the `DBLOGIN` command in the GGSCI command interface. An example is when you add supplemental logging with the `ADD TRANDATA` command.

Encrypting the login password is a recommended security measure. However, using a secure password in the standard `DBLOGIN` command requires encrypting it through the previous use of the `ENCRYPT PASSWORD` command. To avoid having to encrypt the password each time that you issue `DBLOGIN`, and also to protect the user ID from exposure, you can create an Oracle GoldenGate credential store before performing any setup and configuration.

The credential store enables you to simply supply an alias for the login credential whenever you log in with `DBLOGIN`. It also makes the work of specifying login credentials for the Extract and Replicat processes easier and more secure when configuring the parameter files. You can create basic entries in the credential store now and then use the management commands to expand it later as needed. For more information, see *Administering Oracle GoldenGate for Windows and UNIX*.

## 1.4 Supported SQL Server Data Types

Oracle GoldenGate supports most SQL Server data types except those listed under Section 1.5, "Non-Supported SQL Server Data Types."

### Limitations of Support

- Oracle GoldenGate does not support capture from tables that have sparse columns or column sets.
- Oracle GoldenGate does not support the filtering, column mapping, or manipulation of large objects larger than 4K. Full Oracle GoldenGate functionality can be used for objects that are 4K or smaller.
- 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 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 Section 3.2.2, "Assigning Row Identifiers."
- 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, SQL Server configuration. Transformation, filtering, and other types of manipulation are not supported for multi-byte character data.
- If capture of data for `TEXT`, `NTEXT`, `IMAGE`, or `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` and adjust as needed.
- Oracle GoldenGate supports UDT and UDA data of up to 2 GB in size. All UDTs except SQL_Variant are supported.
- Common Language Runtime (CLR), including SQL Server built-in CLR data types (such as, geometry, geography, and hierarchy id), are supported. CLR data types are only supported in a like-to-like SQL Server configuration. Transformation, filtering, and other types of manipulation are not supported for CLR data.
- 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.
- Oracle GoldenGate supports timestamp data from `0001/01/03:00:00:00` to `9999/12/31:23:59:59`. If a timestamp is converted from GMT to local time, these limits also apply to the resulting timestamp. Depending on the time zone, conversion may add or subtract hours, which can cause the timestamp to exceed the lower or upper supported limit.
- `VARBINARY (MAX)` column with the `FILESTREAM` attribute is supported up to a size of 4 GB. Extract uses standard Win32 file functions to read the `FILESTREAM` file.

## 1.5 Non-Supported SQL Server Data Types

- `SQL_Variant` data type is not supported for capture.

## 1.6 Supported Objects and Operations for SQL Server

The following objects and operations are supported:

- Oracle GoldenGate supports capture of transactional DML from user tables, and delivery to user tables and writeable views.
- Capture from tables of a SQL Server Standard Edition database requires a primary key on the tables. Tables for an Enterprise Edition database do not require a primary key.
- Oracle GoldenGate supports the capture and delivery 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.
- Oracle GoldenGate supports the maximum sizes that are permitted for tables that are tracked by CDC (for Enterprise Edition) and Transactional Replication (for Standard Edition).
- Oracle GoldenGate supports capture from tables enabled with `PAGE` and `ROW` compression. For partitioned tables that use compression, all partitions must be enabled with compression.
- Oracle GoldenGate supports capture from 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. To replicate data for non-persisted computed columns, you can use the `FETCHCOLS` or `FETCHMODCOLS` option of the `TABLE` parameter to fetch the column data from the table. Keep in mind that there can be discrepancies caused by differences in data values between when the column was changed in the database and when Extract fetches the data for the transaction record that is being processed.
- Replicat does not apply DML to any computed column, even if the data for that column is in the trail, because the database does not permit DML on that type of column. Data from a source persisted computed column, or from a fetched non-persisted column, can be applied to a target column that is not a computed column.
- In an initial load, all of the data is selected directly from the source tables, not the transaction log. Therefore, in an initial load, data values for all columns, including non-persisted computed columns, gets written to the trail or sent to the target, depending on the method that is being used. As when applying change data, however, Replicat does not apply initial load data to computed columns, because the database does not permit DML on that type of column.
- Oracle GoldenGate 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.

## 1.7 Non-Supported Objects and Operations for SQL Server

The following objects and operations are not supported:

- For source databases, operations that are not supported by SQL Server Change Data Capture or Transactional Replication. Refer to SQL Server Books Online for a complete list of the operations that are limited by enabling SQL Server Change Data Capture (for Enterprise Edition) and Transactional Replication (for Standard Edition).
- Extraction or replication of DDL (data definition language) operations.
- Capture 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.
- Partitioned tables that have more than one physical layout across partitions.
- Partition switching.
- Oracle GoldenGate does not support non-native SQL Server transaction log backups, such as those offered by third-party vendors. However, if using the `TRANLOGOPTIONS` parameter with the `ACTIVESECONDARYTRUNCATIONPOINT` option, Extract does not need to read from any transaction log backups, so any log backup utility may be used. For more information, see Chapter 4, "Preparing the Transaction Logs for Oracle GoldenGate."

# 2 Installing Oracle GoldenGate

This chapter contains the procedures for installing Oracle GoldenGate for Microsoft SQL Server. Installing Oracle GoldenGate installs all of the components that are required to run and manage the processing (excluding any components required from other vendors, such as drivers or libraries) and it installs the Oracle GoldenGate utilities.

This chapter contains the following sections:

- Section 2.1, "Where to Install Oracle GoldenGate"
- Section 2.2, "Understanding and Obtaining the Oracle GoldenGate Distribution"
- Section 2.3, "Preparing to Install Oracle GoldenGate Within a Cluster"
- Section 2.4, "Installing Oracle GoldenGate"
- Section 2.5, "Integrating Oracle GoldenGate into a Cluster"

These instructions are for installing Oracle GoldenGate for the first time. Additionally, they are for downloading the base release of a new version of Oracle GoldenGate.

To download and install subsequent patches to the base release, go to the Patches and Updates tab of My Oracle Support at:

`http://support.oracle.com`

To upgrade Oracle GoldenGate from one version to another, follow the upgrade instructions in *Upgrading Oracle GoldenGate for Windows and UNIX*.

## 2.1 Where to Install Oracle GoldenGate

Oracle GoldenGate capture for SQL Server supports real-time mode and archived log mode. For real-time capture, Oracle GoldenGate must be installed on the source database server. For archived log mode, Oracle GoldenGate may be installed on the source database server or a remote Windows server.

To apply to a Microsoft SQL Server database, you can install Oracle GoldenGate on the database server or a remote Windows server.

## 2.2 Understanding and Obtaining the Oracle GoldenGate Distribution

For complete information about how to obtain Oracle Fusion Middleware software, see "Understanding and Obtaining Product Distributions" in *Planning an Installation of Oracle Fusion Middleware*.

To download the Oracle WebLogic Server and Coherence software for development or evaluation, see the following location on the Oracle Technology Network (OTN):

http://www.oracle.com/technetwork/middleware/fusion-middleware/downloads/index.html

For more information about locating and downloading Oracle Fusion Middleware products, see the *Oracle Fusion Middleware Download, Installation, and Configuration Readme Files* on OTN.

To obtain Oracle GoldenGate follow these steps:

1. Go to Oracle Technology Network.
2. Find the Oracle GoldenGate 12*c* (12.2.0.1) release and download the ZIP file onto your system.

## 2.3 Preparing to Install Oracle GoldenGate Within a Cluster

This topic covers the installation requirements that apply when Oracle GoldenGate will be installed in a cluster environment solution that has the ability to automate failover.

### 2.3.1 Deciding Where to Install Oracle GoldenGate Binaries and Files in the Cluster

You will need to install at least some Oracle GoldenGate objects on shared storage. Select cluster-aware shared storage that is independent of, but available to, all nodes of the cluster.

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

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

- The Oracle GoldenGate installation must have the same location path on every node
- At minimum, install the following directories on the shared storage to support Oracle GoldenGate recovery requirements.
  - `dirchk`
  - `dirdat`
- The parameter files in the `dirprm` directory, if not placed on the shared drive, must be identical on all nodes. To resolve environment settings that must be different from one node to the other, you can set environment settings so they are inherited from the local Manager process or reference a node-specific Oracle GoldenGate macro file. Because this scenario can be difficult to enforce, the inherent concerns can be avoided by storing the parameter files on the shared drive.

  See also Section 2.5, "Integrating Oracle GoldenGate into a Cluster" after you install Oracle GoldenGate.

## 2.4 Installing Oracle GoldenGate

Follow these steps to install Oracle GoldenGate on a Windows system or in the appropriate location in a cluster. See Section 2.3, "Preparing to Install Oracle GoldenGate Within a Cluster" for more information.

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

### 2.4.1 Installing Oracle GoldenGate into a Windows Cluster

To install Oracle GoldenGate into a Windows cluster, do the following:

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.

### 2.4.2 Installing the Oracle GoldenGate Files

To install the Oracle GoldenGate files, do the following:

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
   ```

### 2.4.3 Specifying a Custom Manager Name

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

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

To specify a custom Manager name, use this procedure:

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.

## 2.4.4 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, do the following:

1. (Recommended) Log on as the system administrator.
2. Click **Start** then **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 2–1 install Options***

| Option | Description |
|---|---|
| ADDEVENTS | Adds Oracle GoldenGate events to the Windows Event Manager. |
| 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]<br>The service is installed to start at system boot time (see AUTOSTART). To start it after installation, either reboot the system or start the service manually from the Services applet of the Control Panel. |
| 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. |

*Table 2–1 (Cont.) `install` Options*

| Option | Description |
|---|---|
| `USER` *`name`* | Specifies a domain user account that executes Manager. For *`name`*, include the domain name, a backward slash, and the user name, for example `HEADQT\GGSMGR`.<br>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. 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.

## 2.5 Integrating Oracle GoldenGate into a Cluster

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

### 2.5.1 General Requirements in a Cluster

The requirements for integrating Oracle GoldenGate into a cluster are:

1. Register the Oracle GoldenGate Manager process (and only Manager) as a cluster-managed resource as you would any other application. Manager must be the only Oracle GoldenGate process that the cluster-management software starts and stops, because it is the parent process that manages all other processes.
2. If the cluster uses a virtual IP address, you may need to obtain an available fixed IP address for the Manager process. The VIP must be an available IP address on the public subnet and cannot be determined through DHCP. In the parameter files of the Extract data pumps, specify the VIP of the remote Manager as the input value of the `RMTHOST` parameter. Other Oracle GoldenGate products that access Manager also should use the VIP.
3. When you configure Manager, add the `AUTOSTART` and `AUTORESTART` parameters so that Manager starts the replication processes automatically. You can, when needed, control Extract, Replicat, and other Oracle GoldenGate processes from within the Oracle GoldenGate user interfaces.
4. Mount the shared drive on one node only. This prevents processes from being started on another node. Use the same mount point on all nodes.
5. Configure Oracle GoldenGate as directed in this documentation.

### 2.5.2 Adding Oracle GoldenGate as a Windows Cluster Resource

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

- In the cluster administrator, add the Manager process to the group that contains the database instance to which Oracle GoldenGate will connect.
- Make sure all nodes on which Oracle GoldenGate will run are selected as possible owners of the resource.
- Make certain the Manager Windows service has the following dependencies (configurable from the Services control panel):
    - The 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

# 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 delivery. Some steps apply to just a source system, some just to a target, and some to both.

This chapter is comprised of the following sections:

## 3.1 Configuring a Database Connection

This section contains instructions for setting up the Extract and Replicat connections to a SQL Server database.

### 3.1.1 Configuring an Extract Database Connection

Extract connects to a source SQL Server database through an ODBC (Open Database Connectivity) connection. To create this connection, you set up a data source name (DSN) through the Data Sources (ODBC) control panel. See Section 3.1.3, "Configuring an ODBC Connection" for instructions.

### 3.1.2 Configuring a Replicat Database Connection

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, so you will still need to configure a target ODBC connection.

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

#### 3.1.2.1 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 either ODBC or default OLE DB, follow these steps:

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 Section 3.2.1, "Disabling Triggers and Cascade Constraints on the Target."
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 Section 3.1.3, "Configuring an ODBC Connection."

> **Note:** OLE DB uses the ODBC connection settings to derive connection information for OLE DB together with information on which driver to use.

#### 3.1.2.2 Using OLE DB with `USEREPLICATIONUSER`

If Replicat connects as the SQL Server replication user through OLE DB with the `USEREPLICATIONUSER` option, 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. For SQL Server bidirectional configurations, stagger the seed and increment values like the example in Step 3 of the previous section.

- 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 `USEREPLICATIONUSER`, follow these steps:

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 for bidirectional configurations.
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.
5. Configure an ODBC data source. See Section 3.1.3, "Configuring an ODBC Connection."

### 3.1.3 Configuring an ODBC Connection

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

**Note:** Even when using OLEDB as the apply connection method, Replicat will always use ODBC to query the target database for metadata. Therefore Replicat always requires a DSN to exist.

**To create a SQL Server DSN**

1. Select **Control Panel** then **Administrative Tools** then **Data Sources (ODBC)** to run the ODBC client.
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 for the edition.
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 Integrated Windows Authentication** 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 on each SQL Server source and target system.

## 3.2 Preparing Tables for Processing

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

### 3.2.1 Disabling Triggers and Cascade Constraints on the Target

In an environment where SQL Server is the target, you have to consider triggers and cascade constraints that may repeat an operation that occurred on the source. For example, if the source had an insert trigger on TableA that inserted a record into TableB, and Oracle GoldenGate was configured to capture and deliver both TableA and TableB, the insert trigger on the target table, TableA, must be disabled. Otherwise, the Replicat will insert into TableA, the trigger will fire and insert into TableB, then the Replicat will also try to insert into TableB, and then abend.

The only times you do not need to disable triggers or cascade constraints on the target tables when also replicating those actions from the source is when you are using both the OLE DB connection for the Replicat (which is the default so no parameter needs to be configured) and when you use the `DBOPTIONS USEREPLICATIONUSER` parameter in the Replicat. Note that the constraint, or trigger, or the `IDENTTITY` property must have `NOT FOR REPLICATION` enabled.

The following scenario are applicable for disabling triggers and constraints on the target:

- Uni-directional replication where all tables on the source are replicated; disable the target cascade constraints and triggers.

The following scenarios are applicable to enable triggers and constraints on the target tables:

- Uni-directional replication where tables effected by a trigger or cascade operation are not replicated and the only application that loads these tables is using a trigger or cascade operation; do *not* disable target cascade constraints or triggers.
- Uni-directional or -bi-directional replication where all tables on the source are replicated; set the target table cascade constraints and triggers to enable `NOT FOR REPLICATION` and use the `DBOPTIONS USEREPLICATIONUSER` parameter in the Replicats.

## 3.2.2 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. Source tables can have any kind of key listed in Section 3.2.2.1, "How Oracle GoldenGate Determines the Kind of Row Identifier to Use", except for tables of a SQL Server Standard Edition instance, which require a primary key. 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.

### 3.2.2.1 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 (required for tables of a Standard Edition instance).
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.

#### 3.2.2.2 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 *Reference for Oracle GoldenGate for Windows and UNIX*.

### 3.2.3 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.

## 3.3 Globalization Support

Oracle GoldenGate provides globalization support that enables the processing of data in its native language encoding. The Oracle GoldenGate apply process (Replicat) supports the conversion of data from one character set to another when the data is contained in character column types. For additional support information and limitations, see *Administering Oracle GoldenGate for Windows and UNIX*.

# 4 Preparing the Transaction Logs for Oracle GoldenGate

This section contains information that helps you configure database settings and supplemental logging to support capture of source transaction data by Oracle GoldenGate.

This chapter includes the following sections:

## 4.1 Setting the Database to Full Recovery Model

Oracle GoldenGate requires a SQL Server source database to be set to the Full recovery model and a Full database backup exist.

To verify or set the recovery model:

1. Connect to the SQL Server instance with SQL Server Management Studio for SQL Server.
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 it is not already set.
6. If the database was in **Simple** recovery or never had a **Full** database backup, take a **Full** database backup before positioning Extract.
7. Click **OK**.

## 4.2 Backing Up the Transaction Log

Oracle GoldenGate Capture for SQL Server can run in two modes, real-time capture mode and archived log only mode.

For archived log only mode, Extract only reads from transaction log backups.

For real-time capture mode, depending on how you configure the management of the secondary truncation point (see Section 4.5, "Managing the Secondary Truncation Point"), the Extract process may occasionally require information from a log backup. This happens when the required log records are no longer available in the online log and have been flushed to log backups.

In either mode, Oracle GoldenGate Capture for SQL Server requires the log backup files meet the following conditions:

- The log backup must be a native SQL Server log backup made by issuing the `BACKUP LOG` command (or the corresponding GUI command). Third-party log backups are not supported.
- The log backup can be compressed using native SQL Server compression features.
- The log backup must be made to a DISK device. Valid examples include:

```
BACKUP LOG dbname TO DISK = "c:\folder\logbackup.trn"
BACKUP LOG dbname TO DISK = "\\server\share\logbackup.trn"
```

- Do not overwrite existing log backups.
- Striped log backups are not supported.
- Appending log backups to the same disk device (or file) is not recommended. Mixing compressed and uncompressed log backups to the same device or file is not supported.

## 4.3 Retaining the Log Backups and Backup History

Retain enough log backups and backup history (in the msdb database) 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 SHOWCH` command. For more information, see *Reference for Oracle GoldenGate for Windows and UNIX*.

## 4.4 Enabling Supplemental Logging

These instructions apply to source 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 captured by Oracle GoldenGate.

`ADD TRANDATA` utilizes different features of SQL Server depending on the edition of SQL Server.

For Enterprise Edition, Change Data Capture (CDC) for SQL Server is utilized and creates a minimal Change Data Capture object on the specified table.

For Standard Edition, a SQL Server Transactional Publication is created, as well as an Article for each table enabled with `TRANDATA`.

### Enterprise Edition:

- 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:

  ```
  cdc.dbname_capture
  cdc.dbname_cleanup
  ```

- It is important to review the information on `TRANLOGOPTIONS` options in Section 4.5, "Managing the Secondary Truncation Point" for a complete understanding of this required parameter and the interaction it has with the secondary truncation point and these SQL Server jobs.

### Standard Edition:

- The first requirement to enable `TRANDATA` for tables of a SQL Server Standard Edition database, is that the SQL Server Replication features must be installed, a Distributor must be configured, and a distribution database must be created.
- An Oracle GoldenGate Transactional Publication is created for the database, and an Article per table is created within that database's Publication. The Articles are set to not replicate the data to a distribution database, but are needed to enable supplemental logging for SQL Server Standard Edition.
- As part of creating a Transactional Publication, a SQL Server Log Reader Agent job is created for the database. The Log Reader Agent job has a naming convention of *`instancename-dbname-#`* and job category of REPL-LogReader.
- It is important to review the information on TRANLOGOPTIONS options in Section 4.5, "Managing the Secondary Truncation Point" for a complete understanding of this required parameter and the interaction it has with the secondary truncation point and these SQL Server jobs.

### To enable Supplemental Logging:

This procedure requires a database user who is a member of the SQL Server System Administrators (`sysadmin`) role.

1. On the source system, run GGSCI.
2. Issue the following command to log into the database.

   ```
   DBLOGIN SOURCEDB DSN[, {USERID user, PASSWORD password | USERIDALIAS alias}]
   ```

   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. Alternatively, `USERIDALIAS` *`alias`* is the alias for the credentials if they are stored in a credentials store. If using `DBLOGIN` with a DSN that is using Integrated Windows authentication, the connection to the database for the GGSCI session will be that of the user running GGSCI. In order to issue `ADD TRANDATA` or

DELETE TRANDATA, this user must be a member of the SQL Server sysadmin 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.

```
ADD TRANDATA owner.table
ADD TRANDATA owner.*
```

## 4.5 Managing the Secondary Truncation Point

When you enable supplemental logging with the ADD TRANDATA command for at least one table in a source SQL Server 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 how the secondary truncation point is managed. TRANLOGOPTIONS is a required parameter for a SQL Server Extract and has three available options for managing the secondary truncation point, one of which must be chosen:

- Section 4.5.1, "Oracle GoldenGate Manages the Secondary Truncation Point"
- Section 4.5.2, "SQL Server Manages the Secondary Truncation Point"

### 4.5.1 Oracle GoldenGate Manages the Secondary Truncation Point

There are two TRANLOGOPTIONS options that instruct the Oracle GoldenGate Extract process to manage the secondary truncation point, ACTIVESECONDARYTRUNCATIONPOINT and MANAGESECONDARYTRUNCATIONPOINT.You can use either of these options if Extract *will not* be running concurrently (for the same source database) with SQL Server transactional replication and/or CDC that is configured by applications other than Oracle GoldenGate, and each has different characteristics that are described in this section.

**Note:** Using TRANLOGOPTIONS ACTIVESECONDARYTRUNCATIONPOINT or MANAGESECONDARYTRUNCATIONPOINT for Extract when either SQL Server transactional replication and/or CDC configured by applications other than Oracle GoldenGate are running at the same time causes the SQL Server Log Reader Agent or CDC capture job to fail.

**ACTIVESECONDARYTRUNCATIONPOINT has the following characteristics:**

- Extract sets the secondary truncation point in the transaction log, only for those transactions that it has captured.
- ACTIVESECONDARYTRUNCATIONPOINT prevents the truncation of the transactions until Extract has captured them from the online transaction log.
- Because no online data can be truncated prior to Extract capturing it, Extract is never required to read from log backups. The advantage of this method is that you can use third-party transaction log backup software without concern that Extract may need to access the log backups to retrieve unprocessed transaction data.
- Using ACTIVESECONDARYTRUNCATIONPOINT may cause larger log files if Extract has any significant lag. This happens because transaction log backups *will not* be able to immediately truncate the transaction log. Truncation occurs only after Extract has finished capturing the data. Thus, you may need to adjust your storage parameters to accommodate any increased log size.

- Using this option requires that only one Extract is allowed to capture from the source database.
- This option is not valid for an Extract running in archived log mode.
- For SQL Server Enterprise Edition, when the Extract is started, it will stop and delete the CDC capture job for the database thus preventing data from being loaded to any CDC tables. If any transactions for those tables occurred between the time that supplemental logging was added and Extract was started, the corresponding CDC tables may have data in them. This data is not used by Oracle GoldenGate and can be manually deleted or truncated from those CDC tables. Additionally, the CDC cleanup job for the source database can be stopped and disabled, or deleted because it no longer functions once the CDC capture job has been removed.
- For SQL Server Standard Edition, you must manually stop and disable the SQL Server Log Reader Agent job for the source database, prior to positioning the Extract.

**MANAGESECONDARYTRUNCATIONPOINT has the following characteristics:**

- Extract frequently sets the truncation point in the transaction log at preset interval, by setting a high water mark for the truncation point at that point in time in the log.
- As subsequent transaction log backups are taken, transactions before the truncation point are eligible to be removed from the online transaction log, which frees space in the online log. If Extract has not captured a transaction from the online log before it is flushed to a log backup, it will switch to the log backup and read the transaction from there. This ability requires the logs to be backed up only with a native SQL Server backup utility. Third-party backup software cannot be used in this scenario.
- Using this option does allow for multiple Extracts to capture from the same source database, and each Extract should be configured with this option if there are multiple Extracts per a source database.
- For SQL Server Enterprise Edition, when the Extract is started, it will stop and delete the CDC capture job for the database thus preventing data from being loaded to any CDC tables. If any transactions for those tables occurred between the time that supplemental logging was added and Extract was started, the corresponding CDC tables may have data in them. This data is not used by Oracle GoldenGate and can be manually deleted or truncated from those CDC tables. Additionally, the CDC cleanup job for the source database can be stopped and disabled, or deleted because it no longer functions once the CDC capture job has been removed.
- For SQL Server Standard Edition, you must manually stop and disable the SQL Server Log Reader Agent job for the source database.

With either of these options, if Extract is stopped or down for a longer period of time than the log backup frequency, the secondary truncation point will not move and transactions will not be removed from the transaction log after subsequent log backups, potentially causing log file growth and a full transaction log. Should this occur, contact Oracle Support for instructions and ramifications of manually moving the secondary truncation point.

### 4.5.2 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 transactional replication and/or CDC that is configured by applications other than Oracle GoldenGate. SQL Server will manage the secondary truncation point.

# 5 Uninstalling Oracle GoldenGate

This chapter contains the procedures for uninstalling Oracle GoldenGate for Microsoft SQL Server. It 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.

This chapter contains the following sections:

- Section 5.1, "Stopping Processes"
- Section 5.2, "Removing Oracle GoldenGate Database Objects"
- Section 5.3, "Uninstalling Oracle GoldenGate"

## 5.1 Stopping Processes

This procedure stops the Extract and Replication processes. Leave Manager running until directed to stop it.

1. Log on as the system administrator or as a user with permission to issue Oracle GoldenGate commands and delete files and directories from the operating system.
2. Run GGSCI from the Oracle GoldenGate directory.
3. Stop all Oracle GoldenGate processes.

   ```
   STOP ER *
   ```

4. Stop the Manager process.

   ```
   STOP MANAGER
   ```

## 5.2 Removing Oracle GoldenGate Database Objects

Follow these instructions to remove supplemental logging from the source objects in the Oracle GoldenGate capture configuration and to remove the Replicat checkpoint table in the apply configuration. Specific steps and commands may not apply to your configuration.

### On a Source System:

1. Log on as the system administrator or as a user with permission to issue Oracle GoldenGate commands and delete files and directories from the operating system.
2. Run GGSCI from the Oracle GoldenGate directory.
3. Stop all Oracle GoldenGate processes.

```
STOP ER *
```

4. Stop the Manager process.

   ```
   STOP MANAGER
   ```

5. Issue the following command to log into the source database.

   ```
   DBLOGIN SOURCEDB DSN[, {USERID user, PASSWORD password | USERIDALIAS alias}]
   ```

   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. Alternatively, USERIDALIAS *alias* is the alias for the credentials if they are stored in a credentials store. If using DBLOGIN with a DSN that is using Integrated Windows authentication, the connection to the database for the GGSCI session will be that of the user running GGSCI. In order to issue DELETE TRANDATA or DELETE TRANDATA, this user must be a member of the SQL Server sysadmin server role.

6. Do one of the following, depending on how Extract was running in relation to other replication or CDC components:

   - If Extract was *not* running concurrently with SQL Server transactional replication or a non-Oracle CDC configuration on the same database, open a query session in Management Studio and issue the following statement against the source database to disable and delete any CDC or replication components, and to clear the secondary truncation point.

     For SQL Server Enterprise Edition:

     ```
     EXEC sys.sp_cdc_disable_db
     ```

     For SQL Server Standard Edition:

     ```
     EXEC sp_removedbreplication @dbname='sourcedbname', @type='tran'
     ```

   - If Extract was running *concurrently* with SQL Server transactional replication or a non-Oracle CDC configuration on the same database, run GGSCI and then issue the following command for each table that is in the Extract configuration. You can use a wildcard to specify multiple table names.

     ```
     DELETE TRANDATA owner.table
     ```

7. Remove the heartbeat table by running the DELETE HEARTBEATTABLE command.

### On a Target System:

1. Stop Replicat.

   ```
   STOP REPLICAT group
   ```

2. Issue the following command to log into the target database.

   ```
   DBLOGIN SOURCEDB DSN[, {USERID user, PASSWORD password | USERIDALIAS alias}]
   ```

   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. Alternatively, USERIDALIAS *alias* is the alias for the credentials if they are stored in a credentials store. If using DBLOGIN with a DSN that is using Integrated Windows authentication, the connection to the database for the GGSCI session will be that of the user running GGSCI. In order to issue DELETE CHECKPOINTTABLE, this user must be a member of the SQL Server sysadmin server role.

3. Remove the Replicat checkpoint tables and heartbeat by running the DELETE CHECKPOINTTABLE and DELETE HEARTBEATTABLE commands.

```
DELETE CHECKPOINTTABLE schema.table
DELETE HEARTBEATTABLE
```

## 5.3 Uninstalling Oracle GoldenGate

Follow these instructions to remove the Oracle GoldenGate installed files.

### 5.3.1 Removing Oracle GoldenGate Windows Components

This procedure does the following: removes Oracle GoldenGate as a Windows cluster resource from a source or target Windows system, stops Oracle GoldenGate events from being reported to the Windows Event Manager, and removes the Manager service. Perform these steps on source and target systems.

1. 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. (Cluster) Working from the node in the cluster that owns the cluster group that contains the Manager resource, run GGSCI and make certain that all Extract and Replicat processes are stopped. Stop any that are running.

```
STATUS ER *
STOP ER *
```

3. (Cluster) Use the Cluster Administrator tool to take the Manager resource offline.
4. (Cluster) Right click the resource and select **Delete** to remove it.
5. Click **Start** then **Run**, and then type cmd in the Run dialog box to open the command console.
6. Change directories to the Oracle GoldenGate installation directory.
7. Run the install utility with the following syntax.

```
install deleteevents deleteservice
```

8. (Cluster) Move the cluster group to the next node in the cluster, and repeat from step 5.

### 5.3.2 Removing the Oracle GoldenGate Files

Perform these steps on all systems to remove the Oracle GoldenGate installation directory.

1. In GGSCI, verify that all processes are stopped. Stop any that are running.

```
STATUS MANAGER
STATUS ER *
```

```
STOP MANAGER
STOP ER *
```

2. Exit GGSCI.

   ```
   EXIT
   ```

3. Remove the Oracle GoldenGate installation directory.

# A

# 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.

This appendix contains the following sections:

- Section A.1, "Oracle GoldenGate Programs and Utilities"
- Section A.2, "Oracle GoldenGate Subdirectories"
- Section A.3, "Other Oracle GoldenGate Files"
- Section A.4, "Oracle GoldenGate Checkpoint Table"
- Section A.5, "Oracle GoldenGate HeartBeat Table"

## A.1 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 A–1 Oracle GoldenGate Installed Programs and Utilities***

| Program | Description |
|---|---|
| `convchk` | Converts checkpoint files to a newer version. |
| `convprm` | Converts parameter files that do not use SQL-92 rules for quoted names and literals to updated parameter files that use SQL-92 rules. SQL-92 format for quoted object names and literals was introduced as the default with version 12c of Oracle GoldenGate. |
| `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. |

*Table A–1 (Cont.) Oracle GoldenGate Installed Programs and Utilities*

| Program | Description |
|---|---|
| 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. |
| ggcmd | Associated program of GGSCI. Launches and monitors external applications, such as the JAgent of Oracle GoldenGate Monitor. Integrates those applications into the GGSCI environment. |
| ggsci | User interface to Oracle GoldenGate for issuing commands and managing parameter files. |
| ggsmgr.jcl<br>ggsmgr.proc<br>ggsmgrst.jcl<br>ggsmgrst.proc | Start the Oracle GoldenGate Manager process from a batch job or the operator console on a DB2 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. |
| oggerr | Manages Oracle GoldenGate error messages. |
| 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. |

## A.2 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 A–2 Oracle GoldenGate Installed 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 A–2 (Cont.) Oracle GoldenGate Installed Subdirectories*

| Directory | Description |
|---|---|
| `dirdb` | Contains the data store that is used to persist information that is gathered from an Oracle GoldenGate instance for use by the Oracle GoldenGate Monitor application or within Oracle Enterprise Manager. |
| `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.<br>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.<br>Do not edit these files.<br>Examples:<br>`ext1.cpe`<br>`rep1.cpr` |
| `dircrd` | Contains credential store files. |
| `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.<br>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).<br>Do not edit these files.<br>Examples:<br>`rt000001`<br>`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.<br>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.<br>Example:<br>`defs.dat` |
| `dirdmp` | Contains trace, or dump, files that support the internal activity logging mechanism. |
| `dirjar` | Contains the Java executable files that support Oracle GoldenGate Monitor. |

***Table A–2 (Cont.) Oracle GoldenGate Installed Subdirectories***

| 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).<br>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.<br>Do not edit these files.<br>Examples:<br>`mgr.pcm`<br>`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` .<br>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.<br>Examples:<br>`defgen.prm`<br>`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.<br>File name format is *`group name+sequence number.rpt`* where `sequence number` is a sequential number appended to aged files.<br>Do not edit these files.<br>Examples:<br>`fin2.rpt`<br>`mgr4.rpt` |
| `dirsql` | Used by the `triggen` utility to store SQL scripts before `triggen` was deprecated. Currently used to store training scripts and any user-created SQL scripts that support Oracle GoldenGate. |
| `dirtmp` | The default location for storing transaction data when the size exceeds the memory size that is allocated for the cache manager. Do not edit these files. |
| `dirwlt` | Contains Oracle GoldenGate wallet files. |
| `UserExitExamples` | Contains sample files to help with the creation of user exits. |

## A.3 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 A–3 Other Oracle GoldenGate Installed 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` utility. |
| `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_ddl2file.sql` | Script that saves DDL from the marker table to a file. |
| `ddl_disable.sql` | Script that disables the Oracle GoldenGate DDL trigger. (Oracle installations) |
| `ddl_enable.sql` | Script that enables the Oracle GoldenGate DDL trigger. (Oracle installations) |
| `ddl_filter.sql` | Script that supports filtering of DDL by Oracle GoldenGate. This script runs programmatically; do not run it manually. |
| `ddl_`<br>`nopurgeRecyclebin.sql` | Empty script file for use by Oracle GoldenGate support staff. |
| `ddl_ora11.sql`<br>`ddl_ora12.sql` | Scripts that run programmatically as part of Oracle GoldenGate DDL support; do not run these scripts. |
| `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 recycle bin in support of the DDL replication feature. |
| `ddl_remove.sql` | Script that removes the DDL extraction trigger and package. (Oracle installations) |
| `ddl_session.sql`<br>`ddl_session1.sql` | Supports the installation of the Oracle DDL objects. This script runs programmatically; do not run it manually. |
| `ddl_setup.sql` | Script that installs the Oracle GoldenGate DDL extraction and replication objects. (Oracle installations) |
| `ddl_status.sql` | Script that verifies whether or not each object created by the Oracle GoldenGate DDL support feature exists and is functioning properly. (Oracle installations) |
| `ddl_staymetadata_off.sql`<br>`ddl_staymetadata_on.sql` | Scripts that control whether the Oracle DDL trigger collects metadata. This script runs programmatically; do not run it manually. |
| `ddl_trace_off.sql`<br>`ddl_trace_on.sql` | Scripts that control whether DDL tracing is on or off. |

***Table A–3 (Cont.) Other Oracle GoldenGate Installed Files***

| 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_scriptname*.sql<br>demo_*more_db_scriptname*.sql | Scripts that create and populate demonstration tables for use with tutorials and basic testing. |
| .dmp files | Dump files created by Oracle GoldenGate processes for tracing purposes. |
| ENCKEYS | User-created file that stores encryption keys. Written in external ASCII format. |
| exitdemo.c | User exit example. |
| exitdemo_utf16.c | User exit example that demonstrates how to use UTF-16 encoded data in the callback structures for information exchanged between the user exit and the process. |
| freeBSD.txt | License agreement for FreeBSD. |
| ggmessage.dat | Data file that contains error, informational, and warning messages that are returned by the Oracle GoldenGate processes. The version of this file is checked upon process startup and must be identical to that of the process in order for the process to operate. |
| ggserr.log | File that logs processing events, messages, errors, and warnings generated by Oracle GoldenGate. |
| ggsmsg.dll | Windows dynamic link library used by the install program. |
| GLOBALS | User-created file that stores parameters applying to the Oracle GoldenGate instance as a whole. |
| help.txt | Help file for the GGSCI command interface. |
| icudtxx.dll<br>icuinxx.dll<br>icuucxx.dll | Windows shared libraries for International Components for Unicode, where *xx* is the currently used version. |
| jagent.bat | Windows batch file for the Java Agent for Oracle GoldenGate Monitor. |
| jagent.log<br>jagentjni.log | Log files for the Oracle GoldenGate Monitor Agent. |
| jagent.sh | UNIX shell script for the Java Agent for Oracle GoldenGate Monitor |
| LGPL.txt | Lesser General Public License statement. Applies to free libraries from the Free Software Foundation. |
| libodbc.so | ODBC file for Ingres 2.6 on Unix. |
| libodbc.txt | License agreement for libodbc.so. |
| 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 A–3 (Cont.) Other Oracle GoldenGate Installed Files*

| 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) |
| notices.txt | Third-party software license file. |
| odbcinst.ini | Ingres 2.6 on Unix ODBC configuration file. |
| params.sql | Script that contains configurable parameters for DDL support. (Oracle installations) |
| pthread-win32.txt | License agreement for pthread-VC.dll . |
| pthread-VC.dll | POSIX threads library for Microsoft Windows. |
| prvtclkm.plb | Supports the replication of Oracle encrypted data. |
| pw_agent_util.bat<br>pw_agent_util.sh | Script files that support the Oracle GoldenGate Monitor Agent. |
| role_setup.sql | Script that creates the database role necessary for Oracle GoldenGate DDL support. (Oracle installations) |
| sampleodbc.ini | Sample ODBC file for Ingres 2.6 on UNIX. |
| sqlldr.tpl | Template for use with Replicat when creating a control file for the Oracle SQL*Loader bulk-load utility. |
| start.prm<br>stop.prm | DB2 z/OS paramlib members to start and stop the Manager process. |
| startmgr<br>stopmgr | DB2 z/OS Unix System Services scripts to start the Manager process from GGSCI. |
| startmgrcom<br>stopmgrcom | DB2 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. |

## A.4 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 an abbreviation of the type of database that the script supports). For a description of this table, see *Administering Oracle GoldenGate for Windows and UNIX*.

## A.5 Oracle GoldenGate HeartBeat Table

B

# Requirements Summary for Extract/Capture in Archived Log Only (ALO) Mode

Oracle GoldenGate 12.2 for SQL Server includes a new feature of capturing DML from only the SQL Server transaction log backups GoldenGate can run on the database server in an ALO configuration, or optionally, GoldenGate can be installed and run on a middle tier Windows server.

It should be pointed out that when using an ALO mode configuration, replication will have an induced lag which will be based on the log backup interval as well as the time it takes to complete writing out each log backup during that interval, and the time that it takes the Extract to fully process the log backup file.

## B.1 Windows OS Requirements

- Optional requirements if installing and running GoldenGate from a middle tier Windows server:
    - Install SQL Server Client Tools on the middle tier server in order to configure primary database ODBC connection
        * Microsoft SQL Server 2008 SP4 Feature Pack: `https://www.microsoft.com/en-us/download/details.aspx?id=44277`
        * Microsoft SQL Server 2008 R2 SP3 Feature Pack: `https://www.microsoft.com/en-us/download/details.aspx?id=44272`
        * Microsoft SQL Server 2012 Feature Pack: `https://www.microsoft.com/en-us/download/details.aspx?id=29065`
        * Microsoft SQL Server 2014 Feature Pack: `https://www.microsoft.com/en-us/download/details.aspx?id=42295`
    - Set the middle tier server's date, time, and time zone to the same as the primary source database server.
    - Create a network share of the folder that contains the source database transaction log backups. For example, if SQL Server writes log backups to `D:\SQLBackups`, then create a share on this folder that can be accessed by the Extract running on the middle tier Windows server.
- Oracle GoldenGate Manager must run as an account with READ permissions to the log backup folder, the log backups, and the network share if configuring for remote ALO mode capture.
    - The default of Local System Account will work if 'Everybody' has share and folder access (not very secure).

  - Oracle recommends that you use a Windows account to run the Manager service and control share and folder access to that account.

## B.2 SQL Server Instance Requirements

- Valid for Enterprise and Standard Editions

## B.3 Transaction Log Backups

- Set up SQL Server transaction log backups at a tolerable interval, every 15 minutes for example, if this is still within the recovery SLA for the database.
  - The log backup frequency will be minimum data lag for replication as well.
- Set the transaction log backups to write to the folder that is shared on the network if installing GoldenGate on a middle tier Windows server.

## B.4 ODBC Connection

- For an Extract in ALO mode running on the database server, create a System DSN for the database as per normal instructions.
- For an Extract in ALO mode running on a middle tier Windows server, create a system DSN on that server that connects back to the primary database. This connection will be used by Extract to retrieve table metadata, manage the secondary truncation point, and other tasks.

## B.5 Supplemental Logging

- Supplemental logging must be enabled by normal means (`ADD TRANDATA`) via `GGSCI` against the source database tables to be included for capture.
  - Tables to be captured from ALO mode only are still required to have Supplemental Logging enabled.

## B.6 Operational Requirements and Considerations

- Extract must use either `TRANLOGOPTIONS MANAGESECONDARYTRUNCATIONPOINT` or `NOMANAGESECONDARYTRUNCATIONPOINT`.
  - `ACTIVESECONDARYTRUNCATIONPOINT` is not compatible with ALO mode and will cause Extract to Abend if it is enabled.
- Extract must also use the `TRANLOGOPTIONS ARCHIVEDLOGONLY` parameter, which instructs the Extract to capture DML from the transaction log backups only.
- For remote ALO capture, use `TRANLOGOPTIONS ALTARCHIVELOGDEST` as well, listing the path of the network share location (in double quotes) where the SQL Server transaction log backups are located. Extract will query the msdb database to find the full path of the SQL Server log backup that is needed, and substitute the value of the path from the query result with the path listed in `ALTARCHIVELOGDEST`.

# C

# Oracle GoldenGate 12.2 for SQL Server - Standard Edition Capture

The Oracle GoldenGate 12.2 for SQL Server is designed to capture DML from both SQL Server Standard Edition and SQL Server Enterprise Edition.

This chapter contains the following sections:

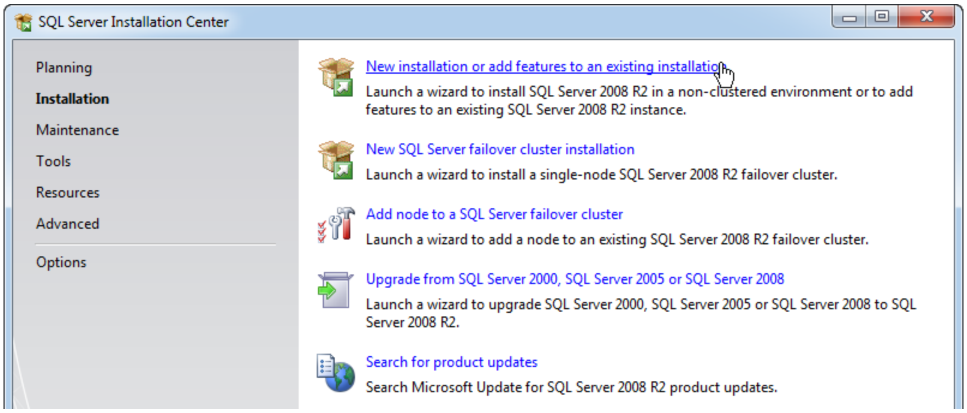
## C.1 Overview

Oracle GoldenGate 12.2 for SQL Server includes a new feature of capturing DML from SQL Server Standard Edition. In the previous GoldenGate 12.1 release, GoldenGate supported capture of DML only from SQL Server Enterprise Edition, due to the method of enabling Supplemental Logging, which is done via a feature only available to Enterprise Editions of SQL Server. Fortunately, beginning with Oracle GoldenGate 12.2, GoldenGate is able to enable Supplemental Logging for Standard Edition instances via a different mechanism. That mechanism to enable Supplemental Logging is done via certain SQL Server Replication components. These SQL Server Replication components are required to be installed and configured in order to enable Supplemental Logging, and the instructions and limitations are outlined in the following sections.

## C.2 SQL Server Instance Requirements

SQL Server Replication components must be installed. This is normally done with initial installation of SQL Server, as an option to check, but in the event that the Replication components are not already installed, you can run the SQL Server installer program again and add that feature to the existing instance.

*Figure C–1 SQL Server Installation Center*



New installation or add features to an existing installation

Launch a wizard to install SQL Server 2008 R2 in a non-clustered environment or to add features to an existing SQL Server 2008 R2 instance.

New SQL Server failover cluster installation

Launch a wizard to install a single-node SQL Server 2008 R2 failover cluster.

Add node to a SQL Server failover cluster

Launch a wizard to add a node to an existing SQL Server 2008 R2 failover cluster.

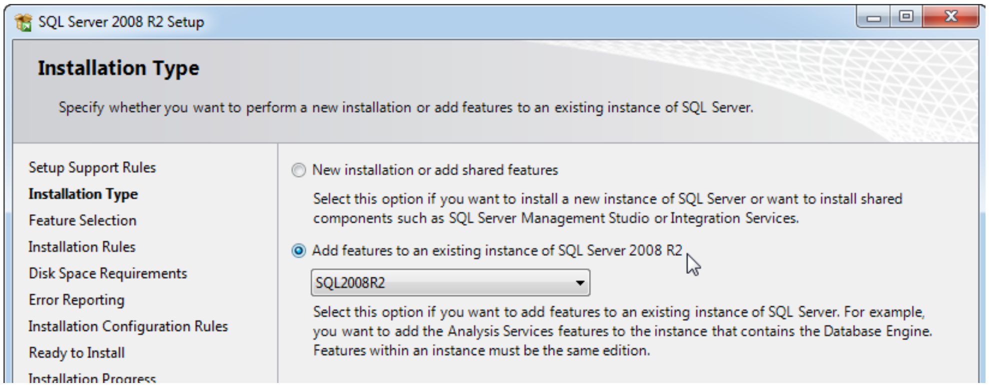
Upgrade from SQL Server 2000, SQL Server 2005 or SQL Server 2008

Launch a wizard to upgrade SQL Server 2000, SQL Server 2005 or SQL Server 2008 to SQL Server 2008 R2.

Search for product updates

Search Microsoft Update for SQL Server 2008 R2 product updates.

*Figure C–2 Selecting Installation Type*



**Installation Type**

Specify whether you want to perform a new installation or add features to an existing instance of SQL Server.

○ New installation or add shared features

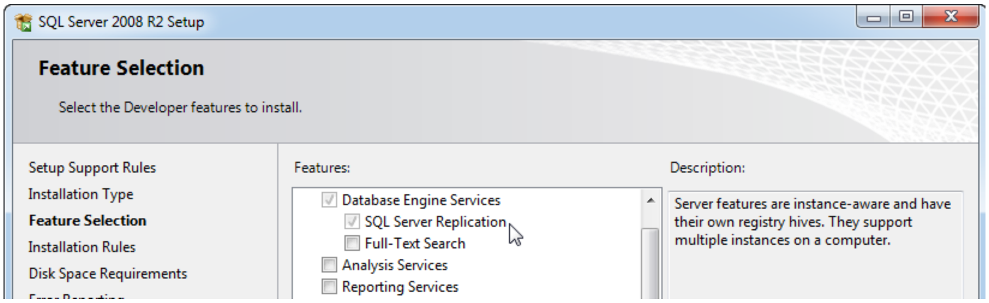
Select this option if you want to install a new instance of SQL Server or want to install shared components such as SQL Server Management Studio or Integration Services.

● Add features to an existing instance of SQL Server 2008 R2

SQL2008R2

Select this option if you want to add features to an existing instance of SQL Server. For example, you want to add the Analysis Services features to the instance that contains the Database Engine. Features within an instance must be the same edition.

*Figure C–3 Feature Selection*



**Feature Selection**

Select the Developer features to install.

Features:

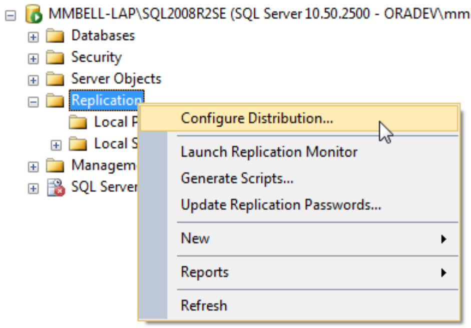
- ☑ Database Engine Services
  - ☑ SQL Server Replication
  - □ Full-Text Search
- □ Analysis Services
- □ Reporting Services

Description:

Server features are instance-aware and have their own registry hives. They support multiple instances on a computer.

Once the SQL Server Replication components are installed, a Distributor must be configured, along with a distribution database. These steps must be done manually prior to attempting to enable Supplemental Logging for any table in the database.

***Figure C–4 Configuring Distribution***


MMBELL-LAP\SQL2008R2SE (SQL Server 10.50.2500 - ORADEV\mmbell)
Databases
Security
Server Objects
Replication
Local P
Local S
Managem
SQL Server
Configure Distribution...
Launch Replication Monitor
Generate Scripts...
Update Replication Passwords...
New
Reports
Refresh

The Distributor can be local or a remote Distributor, and can be one that has already been configured for an existing SQL Server Replication implementation. Oracle GoldenGate does not require the distribution database to store change data, but it must be configured in order to enable supplemental logging

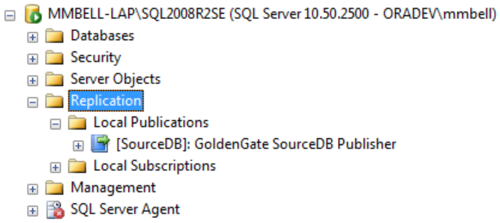
## C.3 Table Requirements

Tables to be enabled with supplemental logging for SQL Server Standard Edition require that a primary key exists on the table.

## C.4 Supplemental Logging

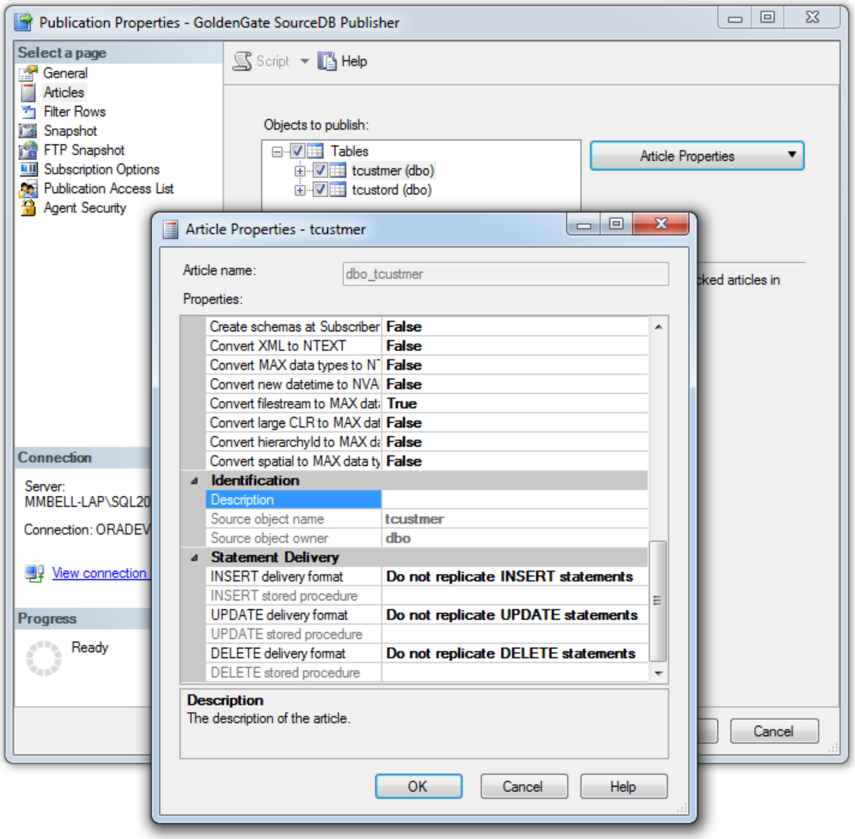
- When enabling supplemental logging for tables within SQL Server Standard Edition, a new SQL Server Replication Publication is created, and the tables enabled for supplemental logging will be added to the Publication as Articles.
- If there are no previous Publications for the database, a new Log Reader job will be created under SQL Server Agent, and depending on the secondary truncation point management method used by Extract, the job may need to be manually stopped and disabled. The created Publication has the naming convention of: "[*DatabaseName*]: GoldenGate *DatabaseName* Publisher".

***Figure C–5 Manually stopping the Reader job***



- The Article properties of the tables to be configured with supplemental logging will not log data changes to the distribution database, but the creation of the Publication with Articles is the requirement to enabling supplemental logging.

***Figure C–6 Publication Properties***



Server:
MMBELL-LAP\SQL20

Connection: ORADEV

Article name: dbo_tcustmer

Properties:

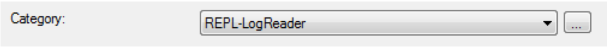
| | |
|---|---|
| Create schemas at Subscriber | False |
| Convert XML to NTEXT | False |
| Convert MAX data types to N | False |
| Convert new datetime to NVA | False |
| Convert filestream to MAX dat | True |
| Convert large CLR to MAX dat | False |
| Convert hierarchyId to MAX d | False |
| Convert spatial to MAX data ty | False |
| **Identification** | |
| Description | |
| Source object name | tcustmer |
| Source object owner | dbo |
| **Statement Delivery** | |
| INSERT delivery format | Do not replicate INSERT statements |
| INSERT stored procedure | |
| UPDATE delivery format | Do not replicate UPDATE statements |
| UPDATE stored procedure | |
| DELETE delivery format | Do not replicate DELETE statements |
| DELETE stored procedure | |

**Description**
The description of the article.

## C.5 Operational Requirements and Considerations

- If Oracle GoldenGate Capture will not be used in conjunction with other SQL Server Transactional Publications for the source database, it is recommended to let the Extract manage the secondary truncation point by one of the two available options listed in See Reference.
- Allowing Oracle GoldenGate to manage the secondary truncation point requires that you manually stop and disable the database Log Reader job SQL Server Agent.
- The database Log Reader job will have a similar naming convention as "*Server*\*Instance*-*DatabaseName*-1" and will be of the job category "REPL-LogReader" when its properties are viewed.

***Figure C–7 REPL-LogReader Category***



Category: REPL-LogReader

- To stop and disable the job, right-click the job name under SQL Server Agent within Management Studio then select **Stop Job**, follow the prompts to stop the job, then right-click again, and then click **Disable**.