Oracle® Fusion Middleware

Installing and Configuring Oracle GoldenGate for SQL Server

12*c* (12.3.0.1)

E81604-01

August 2017

Oracle Fusion Middleware Installing and Configuring Oracle GoldenGate for SQL Server, 12*c* (12.3.0.1)

E81604-01

Contents

Preface

1 System Requirements and Preinstallation Instructions

2 Installing Oracle GoldenGate

3 Preparing the System for Oracle GoldenGate

4 Preparing the Database for Oracle GoldenGate — Classic Capture

D Requirements Summary for Capture and Delivery of Databases in an AlwaysOn Availability Group

E CDC Capture Method Operational Considerations

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.

~~And~~ beginning with Oracle GoldenGate 12.3, there ~~will be~~ two ~~separate~~ data capture methods. The first, ~~which is referred to as~~ Classic Capture, is the transaction log based capture method. The second method, newly introduced ~~with Oracle GoldenGate 12.3~~, is the CDC Capture method.

The Classic Extract binary is available at My Oracle Support, under Patches and Updates, and requires a Service Request in order to receive a password to download the binary. The CDC Extract binary is available on the Oracle Software Delivery Cloud.

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.

Accessible Access to Oracle Support

Oracle customers who 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 Information

The Oracle GoldenGate Product Documentation Libraries are found at

Oracle GoldenGate

Oracle GoldenGate Application Adapters

Oracle GoldenGate for Big Data

Oracle GoldenGate Plug-in for EMCC

Oracle GoldenGate Monitor

Oracle GoldenGate for HP NonStop (Guardian)

Oracle GoldenGate Veridata

Oracle GoldenGate Studio

Additional Oracle GoldenGate information, including best practices, articles, and solutions, is found at:

Oracle GoldenGate A-Team Chronicles

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 *`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` `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

Topics:

- Verifying Certification and System Requirements
- Operating System Requirements
- SQL Server Requirements
- Supported SQL Server Data Types
- Non-Supported SQL Server Data Types and Features
- Supported Objects and Operations for SQL Server
- Non-Supported Objects and Operations for SQL Server

Verifying Certification and System Requirements

Make sure that you are installing your product on a supported hardware and operating system configuration. 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.

Operating System Requirements

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

- Memory Requirements
- Disk Requirements
- Temporary Disk Requirements
- Network
- Operating System Privileges
- Other Programs and Settings

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.

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.~~.~~
- 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. ~~S~~ee 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.~~.~~
- ~~(CDC Capture)~~ When ~~enabling~~ a database for CDC Capture, CDC staging tables are created for each table enabled with supplemental logging and stored within the database. Data from DML operations ~~are~~ inserted into the CDC staging tables and ~~are~~ retained in the database until either the SQL Server CDC Cleanup job~~,~~ or the Oracle GoldenGate Cleanup job~~,~~ removes that data based on the retention period set~~, so~~ plan accordingly to expect more disk usage from the database files.

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.

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. ~~S~~ee the *Administering Oracle GoldenGate* ~~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.

Operating System Privileges

Assign operating system privileges according to the following instructions.

- Manager
- Extract and Replicat

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.
- (Classic Extract) ~~For a source capture installation of Oracle GoldenGate, Manager requires~~ Read permissions on the SQL Server database transaction log files and transaction log backups.

- (Classic Extract) ~~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 operating system level privileges.

Extract and Replicat

See Database User for Oracle GoldenGate Processes.

Other Programs and Settings

~~O~~bserve the following ~~other~~ program and settings information for Oracle GoldenGate for SQL Server:

- To install capture on a remote Windows server, for a Classic Extract in archived log mode or for a remote CDC Extract, 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. ~~Make certain it is the SP1 version of this package.~~ This package installs runtime components of Visual C++ Libraries that are required for Oracle GoldenGate processes. to download this package, go to https://www.microsoft.com/en-us/download/details.aspx?id=13523.
- (Classic Extract) To capture from a source SQL Server 2008/2008R2/2012/2014 Standard Edition database, the SQL Server Replication features must be installed and a Distributor must be configured.
- 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, a Classic Extract running off the database server in an Archived Log Mode, or a remote CDC Extract, 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`

  For SQL Server 2012, 2014, and 2016, install the Microsoft SQL Server 2012 Native Client, which is part of the Microsoft SQL Server 2012 Feature Pack:`https://www.microsoft.com/en-us/download/details.aspx?id=29065`
- Oracle GoldenGate fully supports virtual machine environments created with any virtualization software on any platform. When ~~installing~~ Oracle GoldenGate into a virtual machine environment, select a build that matches the database and the operating system of the virtual machine, not the host system.

SQL Server Requirements

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

- Instance Requirements
- Database Requirements
- Table Requirements
- Database Connection
- Database User for Oracle GoldenGate Processes
- Encrypting and Storing User Credentials

Instance Requirements

~~To~~ configure Oracle GoldenGate for SQL Server instance~~, the following must be true~~:

- Classic Extract supports SQL Server Enterprise and Standard editions of versions 2008, 2008R2, 2012, and 2014.
- CDC Extract supports SQL Server Enterprise editions of versions 2008, 2008R2, 2012, 2014, and 2016.
- (CDC Extract) – For SQL Server 2012, 2014, and 2016, Microsoft has identified and fixed an issue where some `UPDATE` operations may be written incorrectly to a CDC staging table as an `INSERT` followed by a `DELETE`, rather than a `DELETE/INSERT` pair. This may cause downstream replication issues, such as a primary key violation~~, therefore it is recommended to~~ apply the Microsoft fix for this issue: https://support.microsoft.com/en-us/help/3030352
- Oracle GoldenGate Delivery supports SQL Server Enterprise and Standard editions of versions 2008, 2008R2, 2012, 2014, and 2016.
- The SQL Server server name (`@@SERVERNAME`) ~~should~~ not be NULL.
- (CDC Extract) ~~For the source SQL Server instance,~~ the SQL Server Agent must be running in order for the CDC Capture Job to load change data to the CDC tables~~, in order for Oracle GoldenGate to capture transactional data~~.
- (CDC Extract) ~~For~~ SQL Server 2016, ~~prior to enabling~~ supplemental logging, ensure that you have ~~patched the SQL Server instance based on~~ the following bug fix from Microsoft: https://support.microsoft.com/en-us/help/3166120/fix-could-not-find-stored-procedure-sys.sp-cdc-parse-captured-column-list-error-in-sql-server-2016 If ~~the instance is not correctly patched with the Microsoft fix~~, issuing `ADD TRANDATA` against a table for the database may incorrectly report that supplemental logging succeeded, when in fact it may not have~~; therefore~~ no records will be captured for ~~that~~ table.
- (Classic Extract) 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~~.
- 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`” and adjust as needed.

Database Requirements

~~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.
- (CDC Extract) The source database can be configured with Transparent Data Encryption (TDE).
- (CDC Extract) ~~S~~ource database can be set to any recovery model that supports ~~Microsoft SQL Server's~~ ~~C~~hange ~~D~~ata ~~C~~apture feature.
- (Classic Extract) Source databases must be set to the Full recovery model.
- (Classic Extract) After the source database is set to full recovery, a full database backup ~~must be taken~~. ~~T~~his 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~~.
- (Classic Extract) If a source database was restored from a backup ~~of a database from~~ another instance, ~~the database must have a new backup taken on the new instance, once restored~~.
- If the source database was created by restoring a backup from a different instance ~~to the current instance~~, the database owner SID ~~must be synced~~ with ~~a~~ SID ~~that exists~~ on the new instance. Alternatively `sp_changedbowner` ~~can be used~~ to set the restored database to a current login ~~for that instance~~.
- (Classic Extract) 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.
- AlwaysOn - Extract supports capturing from the ~~P~~rimary database, or a read-only, ~~S~~ynchronous mode ~~S~~econdary database~~s~~. Asynchronous mode ~~S~~econdary databases are not supported for capture.

Limitations:

- (CDC Extract) ~~The source database must not be configured with SQL Server Transactional Replication. The~~ CDC Extract ~~cannot~~ work in conjunction with a database ~~enabled with~~ SQL Server ~~T~~ransactional ~~R~~eplication.
- Oracle GoldenGate does not support capture or delivery of system databases.
- Oracle GoldenGate does not support capture from ~~C~~ontained 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 (~~I~~n-~~M~~emory ~~O~~ptimization) 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~~/~~2016 provide~~s~~ the option of ~~D~~elayed ~~T~~ransaction ~~D~~urability for transactions~~, h~~owever, transactions against a table configured with supplemental logging (`TRANDATA`) by Oracle GoldenGate will always be ~~F~~ully ~~D~~urable.
- (Classic Extract) The source database must not be configured with ~~T~~ransparent ~~D~~ata ~~E~~ncryption (TDE).
- Capture from ~~A~~synchronous mode ~~S~~econdary databases of an Always On Availability ~~G~~roup are not supported.

Table Requirements

Tables to be included for capture and delivery must meet the following requirements ~~and must only include data types listed in Supported SQL Server Data Types~~:

- Oracle GoldenGate supports capture of transactional DML from user tables, and delivery to user tables and writeable views.
- DDL operations are not supported.
- Source tables for an Enterprise Edition database do not require a primary key.
- (Classic Extract) Source tables for a Standard Edition database do require a primary key.
- Oracle GoldenGate supports the maximum table names and column lengths~~ that are permitted~~ for tables that are tracked by SQL Server Change Data Capture for Enterprise Edition, and~~ that are permitted~~ for tables that are ~~A~~rticles of a ~~P~~ublication for SQL Server ~~T~~ransactional ~~R~~eplication.
- The sum of all column lengths for a table to be captured from ~~cannot~~ exceed the length that is ~~allowed by SQL Server~~ for enabling Change Data Capture for ~~that~~ table. If the sum of all column lengths exceeds ~~that~~ allowed by the SQL Server procedure `sys.sp.cdc_enable_table`, then `ADD TRANDATA` cannot be enabled for that table. The maximum allowable record length decreases as more columns are present, so there is an inverse relationship between maximum record length and the number of columns in the table.

Database Connection

Oracle GoldenGate uses ODBC and 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 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 Configuring a Database Connection.

Limitations of Support

- For SQL Server 2008~~/~~2008R2, use the SQL Server Native Client 10.0 driver. The older SQL Server driver (`SQLSRV32.DLL`) does not support newer SQL Server data types.
- For SQL Server 2012~~/~~2014~~/~~2016, use the SQL Server Native Client 11.0 driver.
- ~~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.
- For SQL Server 2014~~/~~2016, only the SQL Server Native Client 11.0 driver is supported. The ODBC Driver 11/13 for SQL Server is not supported.
- An AlwaysOn Availability Group Listener connection is not supported.

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.

- User that Enables Supplemental Logging
- Extract (Capture) and Replicat (Apply) Users

User that Enables Supplemental Logging

A database user ~~is required to~~ issue the `ADD TRANDATA` command to enable supplemental logging on the source database in ~~the~~ Oracle GoldenGate configuration. A database login command (`DBLOGIN`) is issued from GGSCI before ~~issuing~~ `ADD TRANDATA`.

- The database user that enables `TRANDATA` must ~~be a user that has~~ `sysadmin` rights.

(CDC Extract) Extract can run with *dbowner* permissions~~,~~ however ~~when issuing~~ `ADD HEARTBEATTABLE` ~~or~~ `DELETE HEARTBEATTABLE` or `INFO HEARTBEATTABLE`~~, the user will need *sysadmin* rights.~~

Extract (Capture) and Replicat (Apply) Users

The Oracle GoldenGate Extract processes capture data from a source SQL Server database for initial loads, from the transaction log and transaction log backups for a Classic Extract, and from the CDC tables for a CDC Extract. The Replicat process applies the data 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 here.

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

- 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.
- To use SQL Server authentication, create a dedicated SQL Server login for Extract and Replicat and assign the privileges listed here.

Classic Extract connecting via SQL Server Authentication	CDC Extract connecting via SQL Server Authentication	Replicat connecting via SQL Server Authentication
Member of the SQL Server fixed server role `System Administrators`.	~~At least~~ a member of the `db_owner` fixed database role of the source database.	~~At least~~ a member of the `db_owner` fixed database role of the target database.

For more information about these parameters, see *Administering Oracle GoldenGate for Windows and UNIX*.

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*.

Supported SQL Server Data Types

The following data types are supported for capture and delivery, unless specifically noted ~~otherwise~~:

- Binary Data Types
  - `(binary, varbinary, varbinary (max))`
  - `(Classic Extract – varbinary (max)`with `FILETREAM)`
- Character Data Types
  - `(char, nchar, nvarchar, nvarchar (max), varchar, varchar (max))`
- Date and Time Data Types
  - `(date, datetime2, datetime, datetimeoffset, smalldatetime, time)`
- Numeric Data Types
  - `(bigint, bit, decimal, float, int, money, numeric, real, smallint, smallmoney, tinyint)`
- LOBs
  - `(image, ntext, text)`
- Other Data Types
  - `(timestamp, uniqueidentifier, hierarchyid, geography, geometry, sql_variant (Delivery only), XML)`

Limitations:

- 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 or unique constraint, or a substitute key specified with a `KEYCOLS` clause in the `TABLE` or `MAP` statements. ~~S~~ee Assigning Row Identifiers ~~for further information~~.
- 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.
- (Classic Extract) `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.
- 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.

Limitations on Computed Columns:

- (Classic Extract) 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.
- (CDC Extract) Computed columns, either persisted or non-persisted, are not supported by Microsoft's Change Data Capture~~, therefore~~ no data ~~will be~~ written to the trail for columns that contain computed columns. 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 data base 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.

Non-Supported SQL Server Data Types and Features

- The `SQL_Variant` data type is not supported for capture.
- Sparse columns and ~~C~~olumn ~~S~~ets are not supported for capture, even when their base column data type is supported.
- (CDC Extract) The `FILESTREAM` feature is not supported.
- Tables that contain unsupported data types may cause Extract~~ to Abend~~. ~~As a workaround, you must~~ remove `TRANDATA` from ~~those~~ tables and remove them from the Extract’s `TABLE` statement, or use the Extract’s `TABLEEXCLUDE` parameter for the table.

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.
- (Classic Extract) 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 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 ~~T~~ransactional ~~R~~eplication (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 for partitioned tables if the table has the same physical layout across all partitions.

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 ~~T~~ransactional ~~R~~eplication, such as `TRUNCATE` statements. ~~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 ~~T~~ransactional ~~R~~eplication (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. ~~T~~hese 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.
- (Classic Extract) 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 Preparing the Database for Oracle GoldenGate — Classic Capture.
- (CDC Extract) Due to a limitation ~~with SQL Server's Change Data Capture~~, column level collations that are different from the database collation~~,~~ may cause incorrect data to be written to the CDC tables for character data~~, and~~ Extract will capture them as they are written to the CDC tables.

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.
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*.

Topics:

- Where to Install Oracle GoldenGate
- Understanding and Obtaining the Oracle GoldenGate Distribution
- Preparing to Install Oracle GoldenGate Within a Cluster
- Installing Oracle GoldenGate
- Integrating Oracle GoldenGate into a Cluster

Where to Install Oracle GoldenGate

Oracle GoldenGate capture for SQL Server supports real-time mode and archived log mode. For Classic Extract, real-time capture requires that Oracle GoldenGate ~~must be~~ installed on the source database server. For Classic Extract in archived log mode, Oracle GoldenGate may be installed on the source database server or a remote Windows server. CDC Extract supports real-time capture from the local 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.

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*.

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 the Oracle Technology Network: `http://www.oracle.com/technetwork/middleware/fusion-middleware/downloads/index.html`
2. Find the Oracle GoldenGate 12*c* (12.*x*) release that you want to install.
3. Download the ZIP file appropriate for your system.

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.

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

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`
- ~~T~~he 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~~ inherit~~ed~~ 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 Integrating Oracle GoldenGate into a Cluster after you install Oracle GoldenGate.

Installing Oracle GoldenGate

Follow these steps to install Oracle GoldenGate on a Windows system or in the appropriate location in a cluster. These instructions apply to all versions of Microsoft SQL Server. Additional database preparation is required before running the Oracle GoldenGate processes. See Preparing the System for Oracle GoldenGate.

- Installing Oracle GoldenGate into a Windows Cluster
- Installing the Oracle GoldenGate Files
- Specifying a Custom Manager Name
- Installing Manager as a Windows Service

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.

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~~.~~

~~5. CREATE SUBDIRS~~

6. Issue the ~~following~~ command to exit GGSCI.

~~7. EXIT~~

Specifying a Custom Manager Name

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

- You want to use a name for Manager other than the default of `GGSMGR`.
- ~~There will be m~~ultiple 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~~.~~

~~4.~~ ~~`MGRSERVNAME` *`name`*~~

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

Installing Manager as a Windows Service

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

Installing Manager as a service is required on a Windows ~~C~~luster, 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:

~~4.~~ 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.

Table 2-1 (Cont.) `install` Options

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

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

~~5.~~ 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.

Integrating Oracle GoldenGate into a Cluster

If you install~~ed~~ Oracle GoldenGate in a cluster, take the following steps to integrate Oracle GoldenGate with~~in~~ the cluster solution.

- General Requirements in a Cluster
- Adding Oracle GoldenGate as a Windows Cluster Resource

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.

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.

Topics:

- Configuring a Database Connection
- Preparing Tables for Processing
- Globalization Support

Configuring a Database Connection

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

- Configuring an Extract Database Connection
- Configuring a Replicat Database Connection
- Configuring an ODBC Connection

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 Configuring an ODBC Connection for instructions.

When using Transport Layer Security (TLS) 1.2 in your database environment, include the following parameter in Extract to force it to use a supported connection protocol: DBOPTIONS DRIVER SQLNCLI11.

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.

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

- Using ODBC or Default OLE DB
- Using OLE DB with USEREPLICATIONUSER

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 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 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.

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 Configuring an ODBC Connection.

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. ~~S~~elect **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. ~~C~~lose the confirmation box and the Create a New Data Source box.
15. Repeat this procedure on each SQL Server source and target system.

Preparing Tables for Processing

The table attributes in the following sections must be addressed in your Oracle GoldenGate environment.

- Disabling Triggers and Cascade Constraints on the Target
- Assigning Row Identifiers
- Improving IDENTITY Replication with Array Processing

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 ha~~d~~ an insert trigger on TableA that insert~~ed~~ 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 `IDEN~~T~~TITY` 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~~.

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 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.

- How Oracle GoldenGate Determines the Kind of Row Identifier to Use
- Using KEYCOLS to Specify a Custom Key

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.

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*.

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.

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*.

4

Preparing the Database for Oracle GoldenGate — Classic Capture

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

Topics:

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 ~~F~~ull 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**.

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 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 file is not recommended.
- Mixing compressed and uncompressed log backups to the same device or file is not supported.

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 transactions. ~~This is more information than what SQL Server logs by default~~

`ADD TRANDATA` must be issued for all tables that are to 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 ~~T~~ransactional ~~P~~ublication is created, as well as an ~~A~~rticle for each table enabled with `TRANDATA`.

Enterprise Edition~~:~~

- Oracle GoldenGatedoes 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 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 ~~T~~ransactional ~~P~~ublication is created for the database, and an ~~A~~rticle ~~per~~ table is created ~~within~~ ~~that~~ database's ~~P~~ublication. The ~~A~~rticles 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 ~~T~~ransactional ~~P~~ublication, 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 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~~

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 database login and ~~`PASSWORD`~~ *`password`* is the password~~ that is required if the data source connects via 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.*
```

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:

- Oracle GoldenGate Manages the Secondary Truncation Point
- SQL Server Manages the Secondary Truncation Point
- ~~Oracle GoldenGate Manages the Secondary Truncation Point~~
- ~~SQL Server Manages the Secondary Truncation Point~~

Oracle GoldenGate Manages the Secondary Truncation Point

T~~here 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 SQL Server Change Data Capture (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 occur~~red~~ 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 occur~~red~~ 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.

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 SQL Server CDC that is configured by applications other than Oracle GoldenGate,~~ SQL Server will manage the secondary truncation point.

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*.

5

Preparing the Database for Oracle GoldenGate – CDC Capture

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

Topics:

- Enabling Supplemental Logging
- Retaining the CDC Table History Data
- Enabling Bi-Directional Loop Detection

Enabling Supplemental Logging

~~With the CDC Extract, the method of capturing change data is via SQL Server Change Data Capture tables, so it is imperative that you follow the procedures and requirements below, so that change data is correctly logged, maintained, and captured by Extract.~~

~~You will~~ enable supplemental logging with the `ADD TRANDATA` command~~ so that~~ Extract can capture the information that is required to reconstruct transactions.

`ADD TRANDATA` must be issued for all tables that are to be captured by Oracle GoldenGate~~, and to do so requires that a~~ valid schema be used in order to create the necessary Oracle GoldenGate tables and stored procedures.

Enabling supplemental logging for a CDC Extract does the following:

- Enables the database for SQL Server Change Data Capture.
  - `EXECUTE sys.sp_cdc_enable_db`
- Creates a Change Data Capture table for each base table enabled with supplemental logging, and creates a trigger for each CDC table. The CDC table exists as part of the system tables ~~with~~in the database and ~~has a naming convention like the following:~~ `cdc.OracleGG_basetableobjectid_CT`.
- Creates a tracking table ~~of~~ naming convention `<schema>.OracleGGTranTables`. This table ~~is used to store~~ transaction indicators for the CDC tables~~,~~ and is populated when the trigger for a CDC table is fired. The table ~~will be~~ owned by the schema listed in the `GLOBALS` file’s~~,~~ `GGSCHEMA` parameter.
- Creates a unique fetch stored procedure for each CDC table, as well as several other stored procedures that are required for Extract to function. These stored procedures will be owned by the schema listed in the `GLOBALS` file’s~~,~~ `GGSCHEMA` parameter.
- Also, as part of enabling CDC for tables, SQL Server creates two jobs per database:

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

- The CDC Capture job ~~is the job that~~ reads the SQL Server transaction log and ~~populates~~ the data into the CDC tables~~, and it is from those CDC tables that the Extract will capture the transactions. So it is of extreme importance that~~ the `cdc.dbname_capture` job ~~be running~~ at all times. This ~~too~~ requires that SQL Server Agent ~~be set to run~~ at all times and ~~enabled to run automatically~~ when SQL Server starts.
- ~~Additional tuning information of~~ the CDC Capture job ~~can be found in the~~ CDC Capture Method Operational Considerations ~~appendix of this document~~.
- ~~The CDC Cleanup job that is created by Microsoft, does not have any dependencies on whether or not the Oracle GoldenGate Extract has captured data in the CDC tables or not, therefore extra steps should be followed to disable or delete the CDC Cleanup job immediately after `TRANDATA` is enabled. These steps will be listed under~~ Retaining the CDC Table History Data ~~section in this document~~.

To enable ~~S~~upplemental ~~L~~ogging~~:~~

The steps below require a database user who is a member of the SQL Server System Administrators (`sysadmin`) role.

1. In the source Oracle GoldenGate installation, ensure that a `GLOBALS` (all caapsand no extension) file has been created with the parameter `GGSCHEMA <schemaname>`. Ensure that the schema name used has been created in the source database. This schema will be used by all ~~subsequent~~ Oracle GoldenGate components created in the database~~, therefore it is recommended to~~ create a unique schema that is ~~solely~~ used by Oracle GoldenGate, such as `‘ogg’`.

2. On the source system, run `GGSCI`.

3. 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 database login and ~~`PASSWORD`~~ *`password`* is the password that is required if the data source connects via 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.

5. 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.*
```

Retaining the CDC Table History Data

When ~~enabling~~ supplemental logging, data that ~~is required by Extract~~ to reconstruct transactions ~~are~~ stored in a series of SQL Server CDC system tables, as ~~well Oracle GoldenGate objects that are used~~ to track operations within a transaction. ~~And as part of enabling supplemental logging~~, SQL Server ~~will create~~ its own Change Data Capture Cleanup job that runs nightly by default, and purges data older than 72 hours. The SQL Server CDC Cleanup job ~~is unaware that an Extract(s) may~~ still require data from these CDC system tables, and ~~can~~ remove ~~that~~ data before ~~the~~ Extract ~~has a chance to~~ capture it.

If data that Extract needs during processing has been deleted from the CDC system tables, then one of the following corrective actions ~~might~~ be ~~required:~~

- Alter Extract to capture from a later ~~point in~~ time ~~for~~ which CDC 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 remedy this situation, Oracle GoldenGate for CDC Extract includes a .bat ~~binary~~ that ~~will create~~ an Oracle GoldenGate Cleanup job and associated stored procedures and tables. Each CDC Extract~~, upon startup, will expect, by default that those~~ Oracle GoldenGate Cleanup job objects exist, and ~~the~~ Extract ~~will~~ shut down if they do not. ~~The~~ Extract ~~will~~ also shutdown if ~~the~~ SQL Server CDC Cleanup job exists alongside the Oracle GoldenGate Cleanup job. The default ~~checks by Extract for~~ the Oracle GoldenGate CDC Cleanup objects can be overwritten by using the `TRANLOGOPTIONS NOMANAGECDCCLEANUP` ~~in the Extract, but this would only be~~ recommended in development and testing situations.

~~Follow the steps below,~~ immediately after ~~enabling~~ supplemental logging and ~~prior to starting the~~ Extract, ~~in order~~ to create the Oracle GoldenGate CDC Cleanup job and associated objects. ~~You can re-run these steps if ever needed to re-enable this feature, should any of the objects get manually deleted.~~

~~To create~~ the Oracle GoldenGate CDC Cleanup job and objects~~:~~

The steps below ~~require~~ a SQL Server authenticated database user who is a member of the SQL Server System Administrators (`sysadmin`) role. ~~Windows authentication is not supported for the .bat binary.~~

1. Manually delete the database’s SQL Server `cdc.`*`dbname`*`_cleanup` job from SQL Server Agent. Alternatively, you can drop it from the source database ~~with~~ the following command

```
EXECUTE sys.sp_cdc_drop_job 'cleanup'
```

2. On the source system, open a command ~~prompt~~ and ~~change to~~ the Oracle GoldenGate installation folder.

3. Run the `ogg_cdc_cleanup_setup.bat` file, providing the following variable values:

```
ogg_cdc_cleanup_setup.bat createJob <userid> <password> <databasename> <servername
\instancename> <schema>
```

In the above, `<userid>` `<password>` ~~is~~ a valid SQL Server ~~login~~ and password for a `sysadmin` user. `<databasename>` `<servername\instancename>` are the source database name and instance name. If only server name is listed, then the default instance will be connected to `<schema>` is the schema name listed in the `GLOBALS` file~~, with the `GGSCHEMA` parameter~~. This schema ~~should~~ be the same for all Oracle GoldenGate objects, including supplemental logging, checkpoint tables, heartbeat tables, and the Oracle GoldenGate CDC Cleanup job.

Example: `ogg_cdc_cleanup_setup.bat createJob ggsuser ggspword db1 server1\inst1 ogg`

The Oracle GoldenGate CDC Cleanup job ~~when created, is scheduled to run~~ every ten minutes, with a default retention period of ~~seventy two~~ hours. ~~T~~he job ~~will~~ not purge data for ~~an~~ Extract's recovery checkpoint ~~however~~, regardless of the retention period.

~~Additional information of~~ the Oracle GoldenGate CDC Cleanup job ~~can be found in the~~ CDC Capture Method Operational Considerations ~~appendix of this document~~.

Enabling Bi-Directional Loop Detection

Loop detection is a requirement for bi-directional implementations of Oracle GoldenGate, so that an Extract for one source database does not recapture transactions sent by a Replicat ~~from~~ another source database.

~~W~~ith the CDC Extract capture method, ~~by default,~~ any transaction committed by ~~a~~ Replicat into a database where ~~an~~ Extract is configured~~, will~~ recapture ~~that~~ transaction from ~~the~~ Replicat as long as supplemental logging is enabled for ~~those~~ tables that the Replicat is delivering to.

In order to ignore recapturing transactions that are applied by ~~a~~ Replicat, ~~you must~~ use the `TRANLOGOPTIONS FILTERTABLE` parameter for the CDC Extract. The table used as the filtering table~~, will be~~ the Oracle GoldenGate checkpoint table that you ~~must~~ create for the Replicat.

~~To create~~ a Filter Table and enable for Supplemental Logging~~:~~

The steps below ~~require~~ 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 database login and `PASSWORD` *`password`* is the password that is required if the data source connects via 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. Create the Oracle GoldenGate checkpoint table ~~that will~~ be used by the Replicat that ~~will~~ deliver data to the source database. ~~It is recommended to~~ use the same schema name as ~~used~~ in the `GGSCHEMA` parameter of the `GLOBALS` file.

```
ADD CHECKPOINTTABLE <schema>.<checkpointtablename>
```

Example: `ADD TRANDATA` *`ogg.ggchkpt`*

5. Add the Replicat with the checkpoint table information.

Example: `ADD REPLICAT reptgt1, EXTTRAIL ./dirdat/e2,checkpointtable` *`ogg.ggchkpt`*

6. Configure the Extract ~~with~~ the `IGNOREREPLICATES` (on by default) and `FILTERTABLE` parameters, using the Replicat's checkpoint table for the filtering table.

```
TRANLOGOPTIONS IGNOREREPLICATES
```

```
TRANLOGOPTIONS FILTERTABLE ogg.ggchkpt
```

6

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.
Topics:

- Stopping Processes
- Removing Oracle GoldenGate Database Objects
- Uninstalling Oracle GoldenGate

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 *
   ```

Removing Oracle GoldenGate Database Objects

Follow these instructions to remove supplemental logging and any Oracle GoldenGate CDC Cleanup objects from the source database 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~~ if not already done~~.

   ```
   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:

   - **1.** 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 Standar Edition:

     ```
     EXEC sp_removedbreplication @dbname='sourcedbname', @type='tran' EXEC
     sys.sp_cdc_disable_db
     ```

   - **2.** 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 any heartbeat table entries by running the `DELETE HEARTBEATTABLE` command.

8. For a CDC Capture configuration, remove the Oracle GoldenGate CDC Cleanup job and objects if they were created.

   - **1.** Open a command ~~prompt~~ and ~~change to~~ the Oracle GoldenGate installation folder.
   - **2.** Run the `ogg_cdc_cleanup_setup.bat` file, providing the following variable values:

     ```
     ogg_cdc_cleanup_setup.bat dropJob <userid> <password> <databasename>
     <servername\instancename> <schema>
     ```

     In ~~the above~~, `<userid>` `<password>` ~~is~~ a valid SQL Server ~~login~~ and password for a `sysadmin` user. `<databasename>` `<servername\instancename>` are the source database name and instance name. If only server name is listed, then the default instance will be connected to. `<schema>` is the schema name listed in the `GLOBALS`file~~, with the `GGSCHEMA` parameter~~. This schema ~~should~~ be the same for all Oracle GoldenGate objects, including supplemental logging, checkpoint tables, heartbeat tables, and the Oracle GoldenGate CDC Cleanup job.

     Example: `ogg_cdc_cleanup_setup.bat dropJob ggsuser ggspword db1 server1\inst1 ogg`

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
   ```

Uninstalling Oracle GoldenGate

Follow these instructions to remove the Oracle GoldenGate installed files.

- Removing Oracle GoldenGate Windows Components
- Removing the Oracle GoldenGate Files

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~~.

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.

Topics:

- Oracle GoldenGate Programs and Utilities
- Oracle GoldenGate Subdirectories
- Other Oracle GoldenGate Files
- Oracle GoldenGate Checkpoint Table

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.
`extract`	Performs capture from database tables or transaction logs or receives transaction data from a vendor access module.

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

Program	Description
`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`
`ggsmgr.proc`
`ggsmgrst.jcl`
`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.
`ogg_cdc_cleanup_setup`	~~Windows batch file to install~~ the Oracle GoldenGate CDC Cleanup job for CDC Capture.
`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.

Oracle GoldenGate Subdirectories

This ~~S~~ection 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.
`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.
	File name format is *`group_name`*`+`*`sequence_number`*`.`*`ext`* where *`sequence_number`* is a sequential number appended to aged files and *`ext`* is either `cpe` for Extract checkpoint files or `cpr` for Replicat checkpoint files.
	Do not edit these files.
	Examples:
	`ext1.cpe`
	`rep1.cpr`

`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.
	File name format is a user-defined two-character prefix followed by either a six-digit sequence number (trail files) or the user-defined name of the associated Extract process group (extract files).
	Do not edit these files.
	Examples:
	`rt000001`
	`finance`

Table A-2 (Cont.) Oracle GoldenGate Installed Subdirectories

Directory	Description
`dirdef`	The default location for data definition~~s~~ files created by the `DEFGEN` utility to contain source or target data definitions used in a heterogeneous synchronization environment. ~~W~~ritten in external ASCII. File name format is a user-defined name specified in the `DEFGEN` parameter file. These files may be edited to add definitions for newly created tables. If you are unsure of how to edit a definitions file, contact Oracle GoldenGate technical support. Example: `defs.dat`
`dirdmp`	Contains ~~trace, or dump, files~~ that support the internal activity logging mechanism.
`dirjar`	Contains the Java executable files that support Oracle GoldenGate Monitor.

`dirpcs`	Default location for status files. File name format is *`group.extension`* where *`group`* is the name of the group and *`extension`* is either `pce` (Extract), `pcr` (Replicat), or `pcm` (Manager).
	These files are only created while a process is running. The file shows the program name, the process name, the port number, and the process ID.
	Do not edit these files.
	Examples:
	`mgr.pcm`
	`ext.pce`

`dirprm`	The default location for Oracle GoldenGate parameter files created by Oracle GoldenGate users to store ~~run-time~~ parameters for Oracle GoldenGate process groups or utilities. Written in external ASCII format. File name format is *`group name/user-defined name`*`.prm` or `mgr.prm` .
	These files may be edited to change Oracle GoldenGate parameter values after stopping the process. They can be edited directly from a text editor or by using the `EDIT PARAMS` command in GGSCI.
	Example~~s~~:
	`defgen.prm`
	`finance.prm`

`dirrec`	Not used by Oracle GoldenGate.

Table A-2 (Cont.) Oracle GoldenGate Installed Subdirectories

Directory	Description

`dirrpt`	The default location for process report files created by Extract, Replicat, and Manager processes to report statistical information relating to a processing run. Written in external ASCII format.
	File name format is *`group name+sequence number.rpt`* where `sequence number` is a sequential number appended to aged files.
	Do not edit these files.
	Examples:
	`fin2.rpt`
	`mgr4.rpt`

`dirsql`	U~~sed by the `triggen` utility to store SQL scripts before `triggen` was deprecated. Currently u~~sed 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.

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.

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

Component	Description
`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_nopurgeRecyclebin.sql`	Empty script file for use by Oracle GoldenGate support staff.
`ddl_ora11.sql` `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` `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` `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` `ddl_trace_on.sql`	Scripts that control whether DDL tracing is on or off.
`ddl_tracelevel.sql`	Script that sets the level of tracing for the DDL support feature. (Oracle installations)

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

Component	Description
debug files	Debug text files that may be present if tracing was turned on.
demo_*db_scriptname*.sql 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 icuinxx.dll 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 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.

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

Component	Description
`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)
`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~~.
`ogg_create_cdc_cleanup_job.sql`	The tables and procedure template used by the CDC Cleanup utility.
`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` `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` `stop.prm`	DB2 z/OS `paramlib` members to start and stop the Manager process.
`startmgr` `stopmgr`	DB2 z/OS Unix System Services scripts to start the Manager process from GGSCI.

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

Component	Description
`startmgrcom` `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.

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*.

B

Requirements Summary for Classic Extract in Archived Log Only ~~(ALO)~~ Mode

Oracle GoldenGate for SQL Server ~~includes a 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.

~~**Topics:**~~

- Windows OS Requirements
- SQL Server Instance Requirements
- Transaction Log Backups
- ODBC Connection
- Supplemental Logging
- Operational Requirements and Considerations

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.~~

SQL Server Instance Requirements

- Valid for Enterprise and Standard Editions

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~~.

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.

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.

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 Classic Extract for SQL Server Standard Edition Capture

Classic Extract for Oracle GoldenGate for SQL Server is designed to capture DML from both SQL Server Standard Edition and SQL Server Enterprise Edition.
Topics:

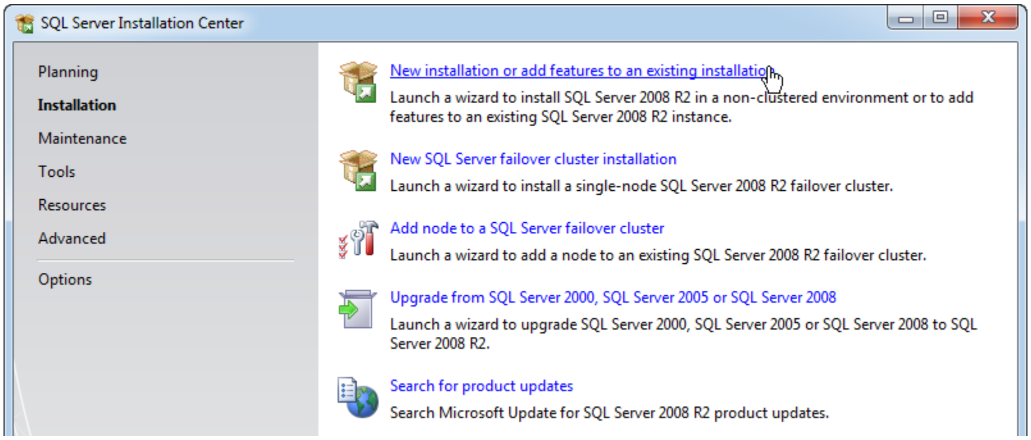
Overview

Oracle GoldenGate for SQL Server includes ~~a~~ Classic Capture support for SQL Server Standard Edition. Oracle GoldenGate utilizes certain SQL Server ~~R~~eplication components in order to enable supplemental logging. 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.

SQL Server Instance Requirements

SQL Server ~~R~~eplication 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


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.

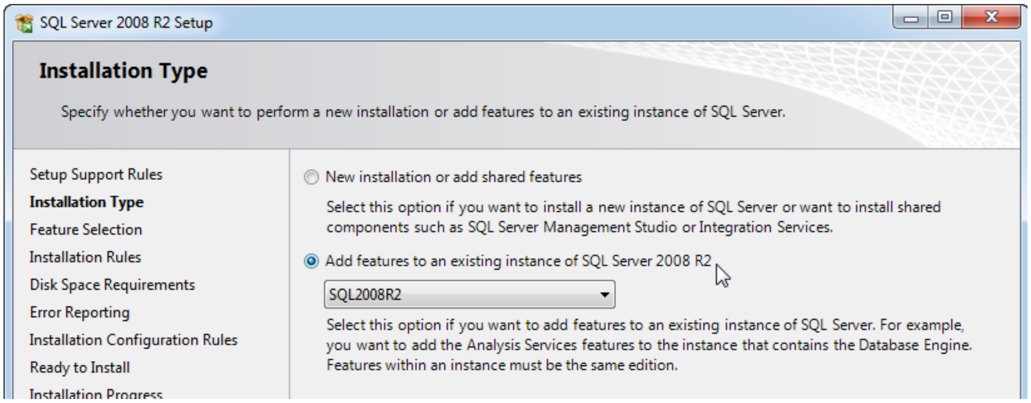
New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2008 R2 failover cluster.

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



SQL Server 2008 R2 Setup

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

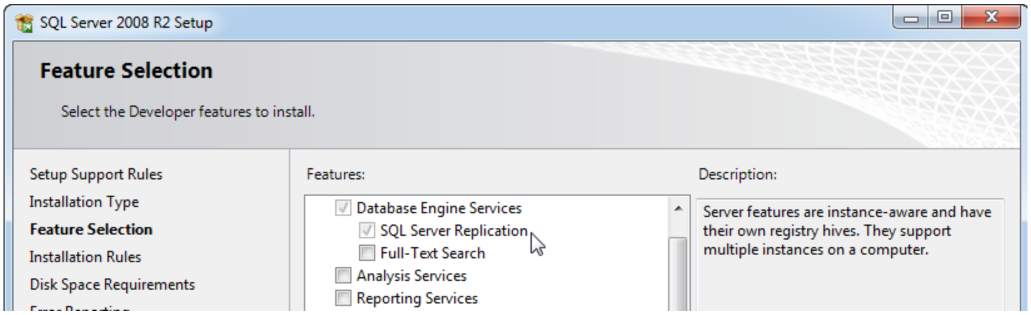
Select this option if you want to install a new instance of SQL Server or want to install shared components such as SQL Server Management Studio or Integration Services.

● 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



SQL Server 2008 R2 Setup

Feature Selection

Select the Developer features to install.

Features:

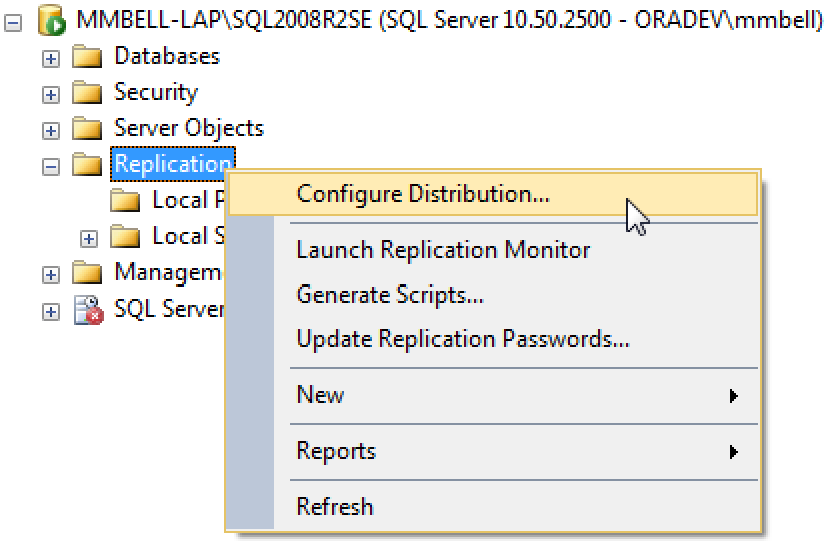
☑ Database Engine Services
☑ SQL Server Replication
□ Full-Text Search
□ Analysis Services
□ Reporting Services

Description:

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

~~Once~~ the SQL Server ~~R~~eplication 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

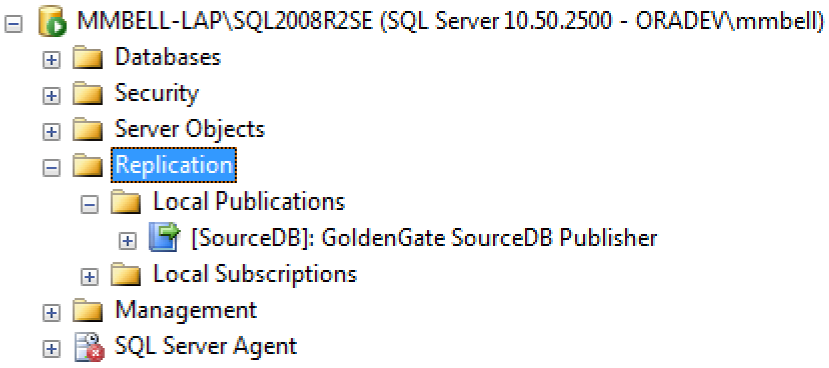
The ~~D~~istributor can be local or ~~a~~ remote ~~Distributor~~, and can be ~~one~~ that has already been configured for an existing SQL Server ~~R~~eplication implementation. Oracle GoldenGate does not require the distribution database to store change data, but ~~it~~ must be configured in order to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

Table Requirements

When ~~enabling~~ supplemental logging for tables ~~with~~in SQL Server Standard Edition, the tables must contain a ~~P~~rimary ~~K~~ey.

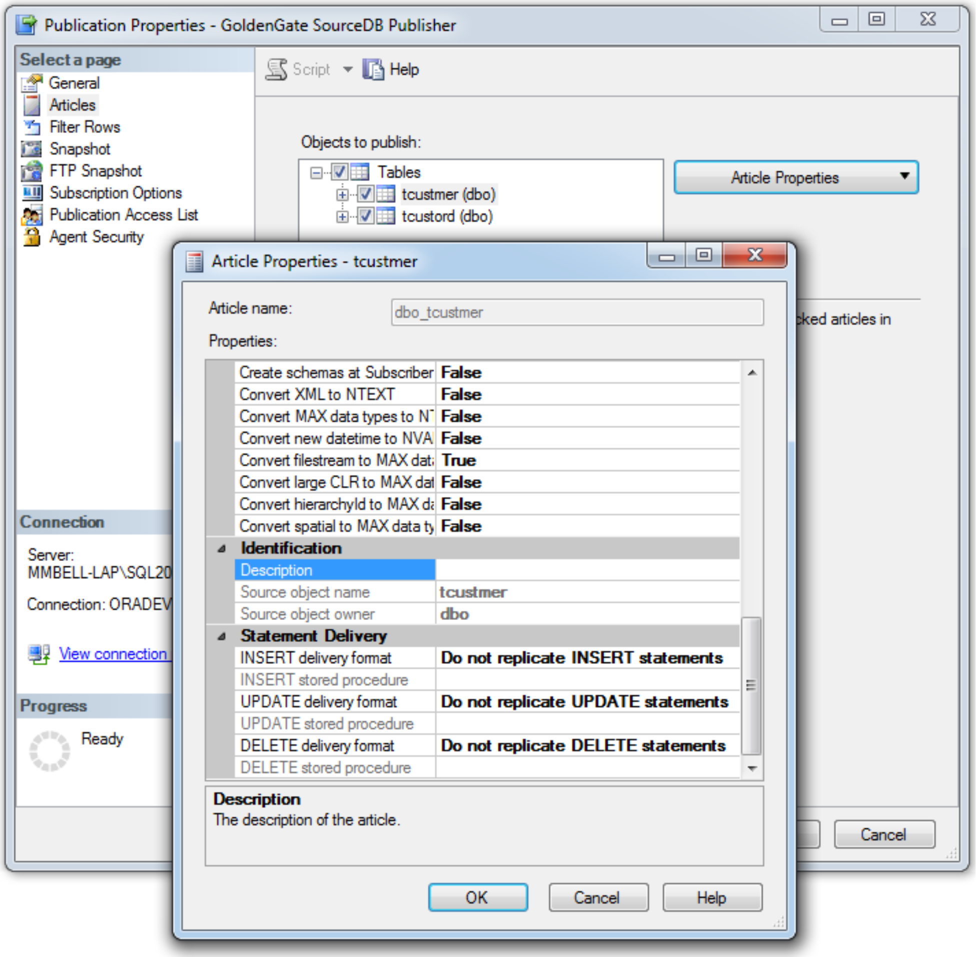
Supplemental Logging

- When ~~enabling~~ supplemental logging for tables ~~with~~in SQL Server Standard Edition, a new SQL Server ~~R~~eplication ~~P~~ublication~~ is created~~, and the tables enabled for supplemental logging ~~will be~~ added to the ~~P~~ublication as ~~A~~rticles.
- If there are no previous ~~P~~ublications 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 ~~P~~ublication ~~has~~ the naming convention~~ of:~~ ~~"~~[*DatabaseName*]: GoldenGate *DatabaseName* Publisher~~"~~.
- **Figure C-5 Manually stopping the Reader job**



- The ~~A~~rticle properties of ~~the~~ tables ~~to be~~ configured with supplemental logging ~~will~~ not log data changes to the distribution database~~, but the creation of the~~ ~~P~~ublication with ~~A~~rticles ~~is the requirement to enabling~~ supplemental logging.

Figure C-6   Publication Properties



Publication Properties - GoldenGate SourceDB Publisher

Select a page: General, Articles, Filter Rows, Snapshot, FTP Snapshot, Subscription Options, Publication Access List, Agent Security

Connection — Server: MMBELL-LAP\SQL20 — Connection: ORADEV — View connection

Progress — Ready

Objects to publish:

- ☑ Tables
  - ☑ tcustmer (dbo)
  - ☑ tcustord (dbo)

Article Properties - tcustmer

Article name: dbo_tcustmer

Properties:

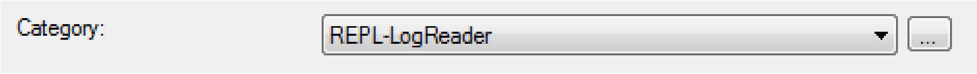
Create schemas at Subscriber	False
Convert XML to NTEXT	False
Convert MAX data types to N	False
Convert new datetime to NVA	False
Convert filestream to MAX dat	True
Convert large CLR to MAX dat	False
Convert hierarchyId to MAX d	False
Convert spatial to MAX data ty	False
Identification	
Description	
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.

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 ~~with~~in Management Studio ~~then s~~elect **Stop Job**, follow the prompts to stop the job, ~~then~~ right-click again, and then click **Disable**.

D

Requirements Summary for Capture and Delivery of Databases in an AlwaysOn Availability Group

Oracle GoldenGate 12.3 for SQL Server includes ~~a new feature of C~~apture support ~~of~~ the ~~P~~rimary and read-only, synchronous mode Secondary databases of an AlwaysOn Availability Group~~, and D~~elivery to the Primary database.

Topics:

- ODBC Connection
- Supplemental Logging
- Operational Requirements and Considerations

ODBC Connection

- Create a System DSN that connects to the Primary (Capture/Delivery) or Secondary (Capture only) replica database as per normal instructions.
- Oracle GoldenGate does not support connections to an AlwaysOn Listener.

Supplemental Logging

- ~~Supplemental logging must be enabled by normal means (ADD TRANDATA) via GGSCI against~~ the Primary database tables to be included for capture, and not against a Secondary replica database.
  - If Oracle GoldenGate is installed on a ~~S~~econdary replica node, you ~~will need to~~ create a separate DSN to the Primary database~~,~~ in order to connect via `DBLOGIN` and run `ADD TRANDATA`.
  - The ~~login used with~~ `DBLOGIN` must ~~have~~ `sysadmin` ~~membership of~~ the ~~P~~rimary replica instance.
  - (CDC Extract) When ~~enabling~~ supplemental logging on the ~~P~~rimary database, the SQL Server Change Data Capture job ~~does~~ not automatically ~~get~~ created~~,~~ ~~t~~herefore, if a ~~S~~econdary database becomes a ~~P~~rimary after failover, ~~then~~ you must manually create the SQL Server CDC capture job on the new ~~P~~rimary ~~by running~~ the following against the new ~~P~~rimary:

    ```
    EXECUTE sys.sp_cdc_add_job 'capture'
    ```

    ~~Consult~~ the Microsoft documentation on how to enable the CDC Capture job for AlwaysOn ~~S~~econdary ~~R~~eplicas ~~for more information~~: https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/replicate-track-change-data-capture-always-on-availability

Operational Requirements and Considerations

- Set the middle tier server's date, time, and time zone to the same as the primary source database server.
- (Classic Extract) ~~Due to~~ the varying nature of where transaction log backups may be taken in an AlwaysOn availability group, Extract must use `TRANLOGOPTIONS ACTIVESECONDARYTRUNCATIONPOINT`.
  - `ACTIVESECONDARYTRUNCATIONPOINT` is not compatible with multiple Extract~~s~~ against the same database ~~or~~ with SQL Server native transactional replication or Change Data Capture.

E

CDC Capture Method Operational Considerations

This section ~~contains~~ information about ~~relevant~~ CDC Capture options, new features, and recommend settings.

Topics:

- Tuning the SQL Server Change Data Capture Job
- New Parameters and Parameters not Valid for CDC Capture
- Details of the Oracle GoldenGate CDC Cleanup Process
- Changing from Classic Extract to a CDC Extract

Tuning the SQL Server Change Data Capture Job

The SQL Server CDC Capture job collects data from the SQL Server transaction log and loads it into CDC staging tables ~~with~~in the database.

~~As part of the job that is created, there are several available tuning parameters that can be used, and information on how to best tune the job can be found in the following article:~~https://technet.microsoft.com/en-us/library/dd266396(v=sql.100).aspx

~~As a general recommendation~~, you should change the SQL Server CDC Capture ~~J~~ob polling interval from the default of 5 seconds~~,~~ to 1 second.

To change the default polling interval ~~of the CDC Capture job~~, ~~execute~~ the following queries against the database:

```
EXEC [sys].[sp_cdc_change_job]

@job_type = N'capture' ,

@pollinginterval = 1,

GO

–stops cdc job

EXEC [sys].[sp_cdc_stop_job]

@job_type = N'capture'

GO

–restarts cdc job for new polling interval to take affect

EXEC [sys].[sp_cdc_start_job]

@job_type = N'capture'
```

New Parameters and Parameters not Valid for CDC Capture

This section describes new parameters added for the CDC Capture method as well as parameters that are not available for ~~the~~ CDC Capture. For more information ~~about supported and unsupported parameters for the CDC Capture method, review~~ *Reference for Oracle GoldenGate fpr Windows and UNIX*.

TRANLOGOPTIONS LOB_CHUNK_SIZE

~~A~~ new ~~Extract parameter~~ LOB_CHUNK_SIZE ~~is add for the CDC Capture method to support large objects~~. ~~If you have huge LOB data sizes, then you can adjust the~~ LOB_CHUNK_SIZE from the default of 4000 bytes~~, to a higher value up~~ to 65535 bytes~~, so that the fetch size is increased~~, reducing the trips needed to fetch the entire LOB.

Example: `TRANLOGOPTIONS LOB_CHUNK_SIZE 8000`

TRANLOGOPTIONS MANAGECDCCLEANUP/NOMANAGECDCCLEANUP

~~A~~ new ~~Extract parameter~~ MANAGECDCCLEANUP/NOMANAGECDCCLEANUP ~~has been added for the CDC Capture method to require or override the requirement, to maintain CDC staging data in the CDC tables, in order for Extract to capture without data loss~~. The default value ~~is~~ MANAGECDCCLEANUP ~~and~~ does not have to be explicitly ~~listed in the Extract, but does require the user to~~ create the Oracle GoldenGate CDC Cleanup job in order for Extract to run with the default. MANAGECDCCLEANUP ~~should be used~~ for all production environments~~, where~~ NOMANAGECDCCLEANUP ~~may be used~~ for temporary and testing implementations ~~as needed~~

Example: `TRANLOGOPTIONS MANAGECDCCLEANUP`

TRANLOGOPTIONS EXLUDEUSER/EXCLUDETRANS

The SQL Server CDC Capture job does not capture user information or transaction names associated with a transaction~~, and as~~ this information is not logged in the CDC staging tables, Extract ~~has no method of excluding~~ DML from a specific user ~~or~~ DML ~~of~~ a specific transaction name. ~~The~~ EXCLUDEUSER and EXCLUDETRANS parameters are ~~therefore~~ not valid for the CDC Capture process.

TRANLOGOPTIONS
MANAGESECONDARYTRUNCATIONPOINT/NOMANAGESECONDARYTRUNCATIONPOINT/
ACTIVESECONDARYTRUNCATIONPOINT

~~The~~ SQL Server CDC Capture job ~~is the only process that~~ captures data from the transaction log~~, when using the Oracle GoldenGate CDC Capture method, therefore~~ secondary truncation point ~~management is not handled by the Extract~~, and for the CDC Capture Extract~~, these parameters are not valid~~.

Details of the Oracle GoldenGate CDC Cleanup Process

The Oracle GoldenGate CDC Cleanup job is required for a CDC Extract by default, since Extract defaults to `TRANLOGOPTIONS MANAGECDCCLEANUP`, and is installed from a Windows batch file (`ogg_cdc_cleanup_setup.bat`), which uses `sqlcmd` to connect to the source SQL Server database and create the necessary objects and job.

~~There should be one job for each database enabled for CDC Capture, and you must create the job and objects following the steps mentioned~~ in ~~the~~ Preparing the Database for Oracle GoldenGate – CDC Capture~~ section of this document~~.

~~There are other options to the utility as well, and those will be discussed below.~~

The ~~steps below~~ require a SQL Server authenticated database user who is a member of the SQL Server System Administrators (`sysadmin`) role. Windows authentication is not supported for the .bat binary.

Removing an Extract from the Database

When ~~the~~ Oracle GoldenGate CDC Cleanup object tables exist, each CDC Extract that is started against that database ~~will create~~ an entry in the `OracleGGExtractCheckpoint` table. This entry tracks ~~a particular~~ Extract's point in time recovery checkpoint, which is used as the cutoff LSN for ~~the~~ Oracle GoldenGate CDC cleanup tasks. If ~~there are~~ multiple Extracts running, ~~each logging~~ more recent recovery checkpoints in the table, but one Extract has been removed from the system without removing its entry in~~to~~ the `OracleGGExtractCheckpoint` table, ~~then~~ no data ~~will be~~ purged newer than ~~that~~ deleted Extract's old recovery checkpoint for all of the CDC staging tables. ~~So when deleting~~ an Extract from the database, follow the steps below to remove the Extract from the `OracleGGExtractCheckpoint` table~~ if more than one Extract is running against the database.~~

1. Ensure that the Extract has been deleted ~~via~~ `GGSCI: DELETE EXTRACT <extname>`

2. On the source system, open a command ~~prompt~~ and ~~change to~~ the Oracle GoldenGate installation folder.

3. Run the `ogg_cdc_cleanup_setup.bat` file, providing the following variable values:

```
ogg_cdc_cleanup_setup.bat deleteExtCheckpoint <userid> <password> <databasename>
<servername\instancename> <schema>
```

~~Example~~: ~~`ogg_cdc_cleanup_setup.bat deleteExtCheckoint ggsuser ggspword db1 server1\inst1 ogg`~~

4. ~~You will be prompted to~~ enter the name of the Extract that is to be removed, and ~~once entered~~, press the Enter/Return key to continue.

Modifying the Oracle GoldenGate CDC Cleanup Job

~~The default schedule and retention period for the~~ Oracle GoldenGate CDC Cleanup job ~~of a database is to run~~ every 10 minutes, with a data retention ~~policy~~ of 72 hours (~~listed as~~ 4320 minutes)~~, and may be manually adjusted to a lower or higher~~ schedule and retention period depending on job resource consumption, repositioning needs or space utilization in the database.

To adjust ~~the job execution frequency, manually~~ modify the schedule for the "`OracleGGCleanup_`*`dbname`*`_Job`" job ~~with~~in SQL Server Agent.

~~If you need t~~o adjust the retention period, ~~you must manually~~ edit the job step for the "`OracleGGCleanup_`*`dbname`*`_Job`" job ~~with~~in SQL Server Agent. The job step passes a parameter to the cleanup stored procedure~~, and y~~ou can modify the value for `@retention_minutes` to adjust the data retention policy as needed.

Deleting the Oracle GoldenGate CDC Cleanup Job

~~If you no longer require the Oracle GoldenGate CDC Cleanup job and associated objects and need to remove them,~~ perform the following steps:

1. Open a command ~~prompt~~ and ~~change to~~ the Oracle GoldenGate installation folder.

2. Run the `ogg_cdc_cleanup_setup.bat` file, providing the following variable values:

```
ogg_cdc_cleanup_setup.bat dropJob <userid> <password> <databasename> <servername
\instancename> <schema>
```

Example: `ogg_cdc_cleanup_setup.bat dropJob ggsuser ggspword db1 server1\inst1 ogg`

Changing from Classic Extract to a CDC Extract

~~If you plan~~ to change from using a Classic Extract ~~for an existing database,~~ to a CDC Extract, ~~then you must~~ remove the supplemental logging ~~that was implemented using the Classic Extract installation method,~~ and re-enable supplemental logging using the CDC Extract installation binaries~~, as~~ the calls to enable TRANDATA ~~are different~~ between the two versions, and the implementation of TRANDATA for Classic Extract ~~is not supported by the CDC Extract~~.

Follow these ~~general~~ guidelines to remove~~,~~ and re-enable supplemental logging. ~~Special consideration and planning should be involved if~~ migrating from Classic to CDC Extract in a production system~~, and~~ the information ~~provide here~~ does not cover all requirements ~~and is only offered as general requirements regarding supplemental logging:~~

1. Ensure that the Classic Extract has processed all remaining data in the logs ~~and can be gracefully stopped~~.

2. D~~o one of the following, d~~epending on how Extract was running in relation to other replication or CDC components:

- ~~1.~~ 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.~~

  ```
  EXEC sys.sp_cdc_disable_db
  ```

- ~~2.~~ If Extract ~~was~~ running *concurrently* with SQL Server transactional replication or a non-Oracle CDC configuration on the same database, run GGSCI from the Classic Extract's installation folder, login to the source database with the `DBLOGIN`, 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
  DELETE TRANDATA owner.*
  ```

3. Delete ~~any~~ heartbeat table entries ~~if one was installed.~~

```
DELETE HEARTBEATTABLE
```

4. Using the Oracle GoldenGate CDC Extract installation binaries, follow the steps listed in Preparing the Database for Oracle GoldenGate – CDC Capture to re-enable

supplemental logging and other necessary components, and ~~re-add~~ the heartbeat table.