Oracle® GoldenGate

Microsoft SQL Server Installation and Setup Guide

Version 10.4

October 2009



Oracle® GoldenGate Microsoft SQL Server Installation and Setup Guide, version 10.4

Copyright © 1995, 2009 Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

Contents

Chapter 1 System requirements and preinstallation instructions . . . 3

Overview of GoldenGate for SQL Server . . . 3
Supported platforms . . . 3
Operating system requirements . . . 3
Database requirements . . . 5
Supported data types . . . 6
Non-supported data types . . . 7
Supported objects and operations . . . 7
Non-supported objects and operations . . . 8
Supported and non-supported object names and case . . . 8
Object names and owners . . . 8
Case sensitivity . . . 8
Supported characters . . . 9
Non-supported characters . . . 10

Chapter 2 Installing GoldenGate . . . 11

Installation overview . . . 11
Upgrades . . . 11
New installations . . . 11
Downloading GoldenGate . . . 11
Installing GoldenGate on Windows and Windows Cluster . . . 12
Installing the Microsoft redistributable package . . . 12
Installing GoldenGate into a Windows Cluster . . . 12
Installing the GoldenGate files . . . 12
Specifying a custom Manager name . . . 12
Installing Manager as a Windows service . . . 13
Adding GoldenGate as a Windows cluster resource . . . 14
Configuring Manager and other processes . . . 18

Chapter 3 Preparing the system for GoldenGate . . . 19

Preparing tables for processing . . . 19
Assigning row identifiers . . . 19

Constraining updates when a table has no key ... 20
Disabling triggers and cascade delete constraints ... 20
Preparing IDENTITY columns for replication ... 20
Configuring an ODBC connection ... 21
Creating a SQL Server DSN ... 21
Preventing multiple connections ... 25
Configuring SQL Server logging ... 25
Installing replication components (SQL Server 2005) ... 27
Configuring and cleaning up the Distributor database ... 32
Enabling extended logging ... 34
Maintaining the secondary truncation point (SQL Server 2005) ... 35
SQL Server replication maintains the truncation point ... 35
GoldenGate maintains the truncation point ... 35
Specifying the location of the online and backup logs ... 36
Specifying an alternate location for the online logs ... 36
Specifying an alternate location for the backup logs ... 36
Invoking third-party restore utilities ... 37
Sizing and retaining the logs ... 37

Chapter 4 Uninstalling GoldenGate ... 38

Uninstalling GoldenGate from Windows (non-cluster) ... 38
Uninstalling GoldenGate from Windows Cluster ... 39

Appendix 1 GoldenGate installed components ... 40

GoldenGate Programs and Utilities ... 40
GoldenGate subdirectories ... 42
Other GoldenGate files ... 44
GoldenGate checkpoint table ... 48

Index ... 49

CHAPTER 1

System requirements and preinstallation instructions

Overview of GoldenGate for SQL Server

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

The instructions in this chapter apply to all supported versions of SQL Server unless otherwise noted. Additional instructions may apply to SQL Server 2005 only.

Supported platforms

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

To find out which GoldenGate builds are available for a specific combination of database version and operating system, go to http://support.goldengate.com. A valid user name and password are required to enter this site.

Operating system requirements

Disk requirements

- Assign the following free disk space:
    - 50 MB for the GoldenGate installation files. This includes space for the compressed download file and space for the uncompressed files. You can delete the download file after the installation is complete.
    - 40 MB for the working directories and binaries for each instance of GoldenGate that you are installing on the system. For example, to install two builds of GoldenGate into two separate directories, allocate 80 MB of space.

- Additional disk space on any system that hosts GoldenGate trails, which contain the working data. The space that is consumed by the trails varies, depending on the volume of data that will be processed. A good starting point is 1 GB.
- To install GoldenGate into a cluster environment, install the GoldenGate binaries and files on a shared file system that is available to all cluster nodes.

TCP/IP

- Configure the system to use TCP/IP services, including DNS.
- Configure the network with the host names or IP addresses of all systems that will be hosting GoldenGate processes and to which GoldenGate will be connecting. Host names are easier to use.
- GoldenGate requires the following unreserved and unrestricted TCP/IP ports:
  - One port for communication between the Manager process and other GoldenGate processes.
  - A range of ports for local GoldenGate communications: can be the default range starting at port 7840 or a customized range of up to 256 other ports.
- Keep a record of the ports you assigned to GoldenGate. You will specify them with parameters when configuring the Manager process.
- Configure your firewalls to accept connections through the GoldenGate ports.
- If possible, grant unrestricted FTP access to GoldenGate for transfers of data, parameters, and reports between source and target systems. Otherwise, provide for another transfer method. A secure transfer method is also required to resolve support cases.
- If possible, provide a connection between your source and target systems and a site where files can be staged for transfer to and from the GoldenGate Software FTP Support Site (ftp://support.goldengate.com).

Operating system permissions

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

- full control over the files and folders within the GoldenGate directories.
- full control over the trail files, if stored in a location other than the GoldenGate directory.
- select user rights assignments on the system.

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

Extract and Replicat: See “Database user”.

Microsoft-supplied components

- Before installing GoldenGate on a Windows system, install and configure the Microsoft Visual C ++ 2005 SP1 Redistributable Package. **Make certain it is the SP1 version of this package, and make certain the get the right bit version for your server**. This package installs runtime components of Visual C++ Libraries. For more information, and to download this package, go to http://www.microsoft.com.

Virtual machines

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

Database requirements

Database configuration

- The GoldenGate Extract and Replicat processes use ODBC (Open Database Connectivity) to connect to the database. For more information, see page 21.
- The SQL Server database must be set to the full recovery model, and at least one full database backup must be done before GoldenGate processes are started for the first time. For more information, see page 25.
- You can configure a SQL Server 2005 database for replication in one of these ways:
  - Run GoldenGate replication without having to install the SQL Server 2005 replication components. To use this configuration, obtain from Microsoft and install the Cumulative Update Package 6 (CU6) for SQL Server 2005 Service Pack 2 (or later).
  - Run GoldenGate in conjunction with the SQL Server 2005 replication components. This configuration can be used if the CU6 patch is not installed. GoldenGate will fall back on this configuration automatically if CU6 components are not detected. This manual contains instructions for installing and configuring the replication components to support GoldenGate, including the creation of a Distributor database.

    NOTE If you already are using GoldenGate with native SQL Server replication but want to use the CU6 configuration, see the GoldenGate upgrade instructions on the GoldenGate support site.

Database user

- The GoldenGate processes can use either Windows Authentication or SQL Server Authentication to connect to a database.
- To use Windows authentication, the Extract and Replicat processes inherit the login credentials of the Manager process, as identified by the Log On account specified in the Properties of the Manager service. This account must have the permissions listed in Table 1 on the source and target systems.

- To use SQL Server authentication, create a dedicated SQL Server login for Extract and Replicat and assign the permissions listed in Table 2. If using SQL Server authentication, you will need to specify the user and password with the USERID parameter in the Extract or Replicat parameter file.

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

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

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

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

Supported data types

- Most SQL Server data types except those listed in "Non-supported data types"

Limitations of support

- When the size of a large object exceeds 4K, GoldenGate stores the data in segments within the GoldenGate trail. The first 4K is stored in the base segment, and the rest is stored in a series of 2K segments. GoldenGate does not support filtering, column mapping, or manipulation for large objects of this size. Full GoldenGate functionality can be used for objects that are 4K or smaller.
- GoldenGate treats XML data as a large object (LOB), as does SQL Server when the XML does not fit into a row. There is no size limitation. SQL Server 2008 extended XML enhancements (such as lax validation, DATETIME, union functionality) are not supported.
- A system-assigned TIMESTAMP column or a non-materialized computed column cannot be part of a key. A table containing a TIMESTAMP column must have a key, which can be a primary key, a unique constraint, or a substitute key specified with a KEYCOLS clause in the TABLE or MAP statement (see "Assigning row identifiers" on page 19).
- GoldenGate supports multi-byte character data types and multi-byte data stored in character columns. Multi-byte data is only supported in a like-to-like configuration. Transformation, filtering, and other types of manipulation are not supported for multi-byte character data.

- If data for TEXT, NTEXT, IMAGE, or VARCHAR (MAX) columns will exceed the SQL Server default size set with the 'max text repl size parameter, extend the size. Use sp_configure to view the current value of max text repl size.
- IDENTITY is supported in a unidirectional configuration, and with limitations in a bi-directional configuration.
- GoldenGate supports UDT and UDA data of up to 2 GB in size.
- DATETIME2 is supported for SQL Server 2008 targets, but it requires the DSN (data source name) to be configured with the SQL Server Native Client.

Non-supported data types

- IDENTITY ranges, such as when Sys1 has a range of 1 to 1,000,000 (or 1 to MINVAL) and Sys2 has a range of 1,000,001 to 2,000,000 (or 0 to MAXVAL).
- SQL_Variant
- VARDECIMAL
- DATETIMEOFFSET
- ORDPATH

Supported objects and operations

- GoldenGate supports the extraction and replication of DML operations on tables that contain rows of up to 512 KB in length. TEXT, NTEXT, IMAGE, VARBINARY, and VARCHAR (MAX) columns are supported in their full size.
- GoldenGate supports the maximum number of columns per table that is supported by the database. GoldenGate supports the maximum column size that is supported by the database.
- GoldenGate supports tables that use data compression and transparent data encryption.

Limitations on computed columns

- GoldenGate supports tables with non-materialized computed columns, but GoldenGate does not capture change data for these columns, because the database does not write it to the transaction log. You can use the FETCHCOLS option of the TABLE parameter to fetch the value of a non-materialized computed column. Replicat does not apply DML to a non-materialized 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 non-materialized computed column can be applied to a target column that is not a non-materialized 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-materialized 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 a target non-materialized computed column, because the database does not permit DML on that type of column.
- GoldenGate does not permit a non-materialized computed column to be used in a KEYCOLS clause in a TABLE or MAP statement.

- If a unique key includes a non-materialized computed column, and GoldenGate must use that key, the non-materialized computed column will be ignored. This might affect data integrity if the remaining columns do not enforce uniqueness. Fetching only provides an after value, and GoldenGate requires before and after values of keys.
- If a unique index is defined on any non-materialized computed columns, it will not be used.
- If a unique key or index contains a non-materialized computed column and is the only unique identifier on a table, GoldenGate must use all of the columns as an identifier to find target rows. Because a virtual column cannot be used in this identifier, it is possible that Replicat could apply operations containing this identifier to the wrong target rows.

Non-supported objects and operations

- (SQL Server 2005) All of the SQL Server 2005 tables that are in the GoldenGate configuration will be marked for replication when you perform the installation procedures (see page 34). Some database operations are not supported when tables are marked for replication. *Refer to SQL Server Books Online for a complete list of the operations that are limited by enabling SQL Server Replication.*
- Extraction or replication of DDL (data definition language) operations.
- Extraction from, or replication to, views. The underlying tables can be extracted and replicated.
- Bulk copy operations and operations by the TextCopy utility, WriteText TSQL function, or UpdateText. These programs perform operations that are either not logged by the database or are only partially logged, and so cannot be supported by the Extract process.
- Non-native SQL Server transaction log backups, such as those offered by compression utilities. Do not install GoldenGate on a system where this log backup technology is in use.
- MERGE operations.

Supported and non-supported object names and case

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

Object names and owners

Source and target object names must be fully qualified in GoldenGate parameter files, as in fin.emp.

Case sensitivity

If a database is case-sensitive, GoldenGate supports the case sensitivity of database names, owner names, object names, column names, and user names.

If a database is case-insensitive, or if it supports case-sensitivity but is configured to be case-insensitive, GoldenGate converts all names to upper case.

To preserve case-sensitivity

Case-sensitive names must be specified in GoldenGate parameter files exactly as they appear in the database. Enclose case-sensitive names in double quotes if the other database (the source or target of the case-sensitive objects) is not case-sensitive.

If replicating from a case-insensitive database to a case-sensitive database, the source object names must be entered in the Replicat `MAP` statements in upper case, to reflect the fact that they were written to the trail as uppercase by Extract.

For example:

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

Supported characters

GoldenGate supports alphanumeric characters in object names and the column names of key columns and non-key columns. GoldenGate also supports the following non-alphanumeric characters in columns that are not being used by GoldenGate as a key.

Table 3 Supported non-alphanumeric characters in object names and non-key column names[1]

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

Table 3 Supported non-alphanumeric characters in object names and non-key column names[1]

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

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

Non-supported characters

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

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

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

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

CHAPTER 2

Installing GoldenGate

Installing GoldenGate

Installation overview

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

Upgrades

To upgrade GoldenGate from one version to another, follow the instructions on the GoldenGate support site at http://support.goldengate.com.

New installations

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

- Downloading GoldenGate
- Installing the software

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

Downloading GoldenGate

1. Navigate to http://support.goldengate.com.
2. In the navigation bar, select Downloads.
3. In the navigation bar, select the platform.
4. Select the operating system and database.
5. Locate the correct GoldenGate build.
6. Click Download to transfer the software to your system.

Installing GoldenGate on Windows and Windows Cluster

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

Installing the Microsoft redistributable package

- Before installing GoldenGate on a Windows system, install and configure the Microsoft Visual C ++ 2005 SP1 Redistributable Package. **Make certain it is the SP1 version of this package, and make certain the get the right bit version for your server**. This package installs runtime components of Visual C++ Libraries. For more information, and to download this package, go to http://www.microsoft.com.

Installing GoldenGate into a Windows Cluster

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

Installing the GoldenGate files

1. Unzip the downloaded file(s) using PKUNZIP or WinZip.
2. Move the files in binary mode to a folder on the drive where you want to install GoldenGate. *Do not* install GoldenGate into a folder that contains spaces in its name, for example "GoldenGate Software." GoldenGate relies on path names, but the operating system does not support path names that contain spaces, whether or not they are within quotes.
3. From the GoldenGate folder, run the GGSCI program.

   ```
   GGSCI
   ```

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

   ```
   CREATE SUBDIRS
   ```

5. Issue the following command to exit GGSCI.

   ```
   EXIT
   ```

Specifying a custom Manager name

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

- you want to use a name for Manager other than the default of GGSMGR.

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

To specify a custom Manager name

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

   ```
   EDIT PARAMS ./GLOBALS
   ```

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

   ```
   MGRSERVNAME <name>
   ```

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

Installing Manager as a Windows service

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

To install Manager as a Windows service

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

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

   Where: <option> is one of the following:

Table 5 INSTALL options

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

Table 5 INSTALL options (continued)

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

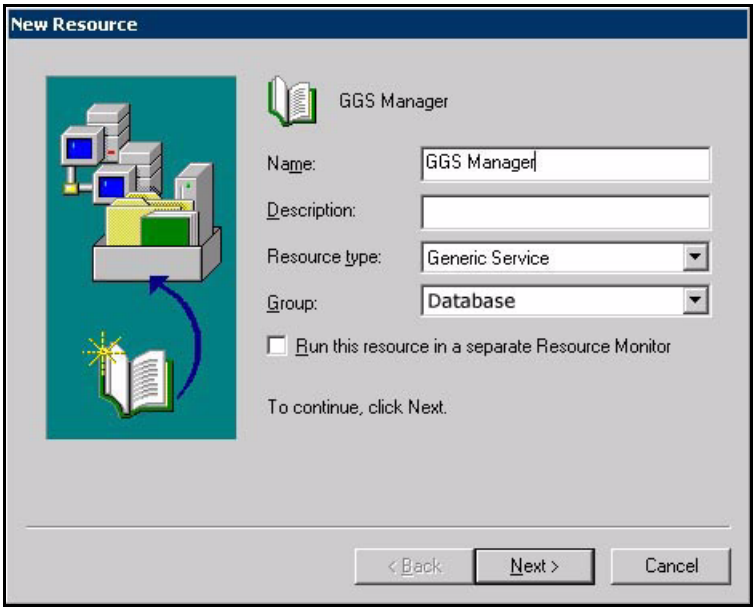
[1] A user account can be changed by selecting the Properties action from the Services applet of the Windows Control Panel.

Adding GoldenGate as a Windows cluster resource

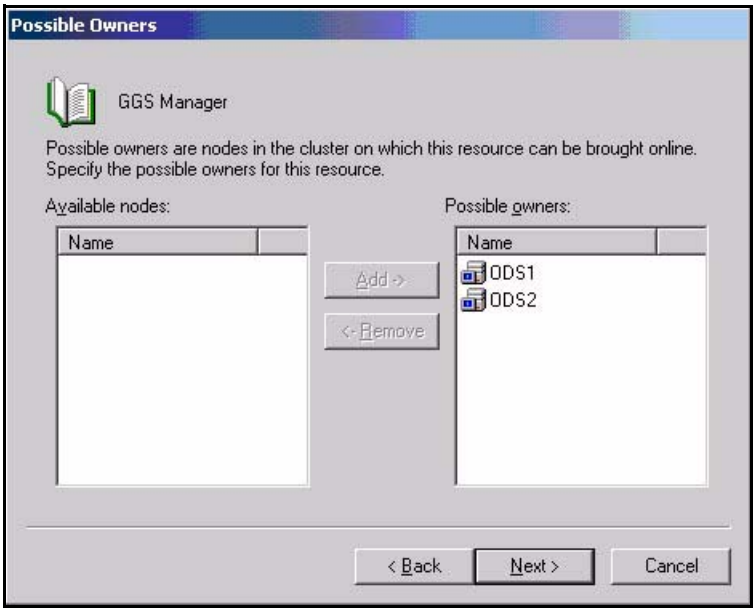
If you installed GoldenGate into a cluster, follow these instructions to establish GoldenGate as a cluster resource and configure the Manager service correctly on all nodes.

1. In the Cluster Administrator, select **File>New>Resource**.

2. In the New Resource dialog box, provide a descriptive name for the GoldenGate Manager (need not be its actual name). For Resource Type, select Generic Service. For Group, select the group that contains the database instance to which GoldenGate will connect.

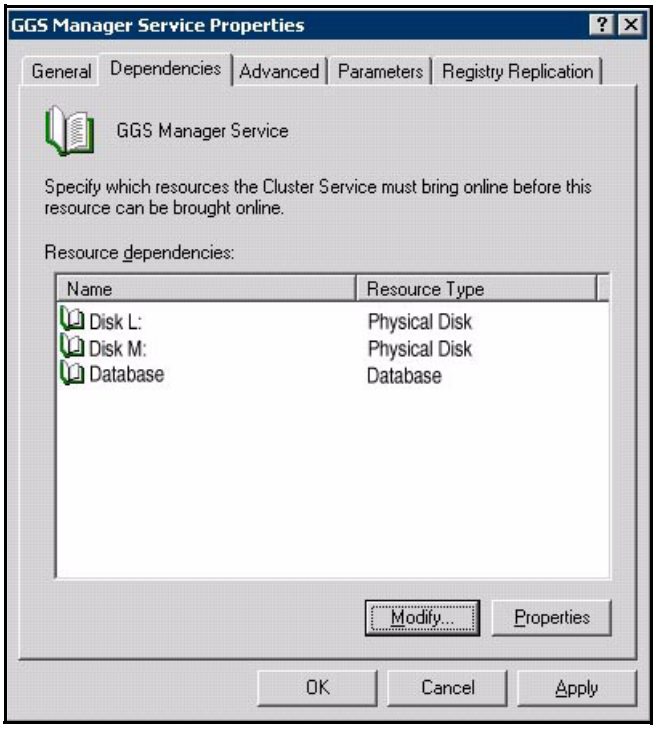


3. Click **Next**.
4. In the Possible Owners dialog box, select the nodes on which GoldenGate will run.

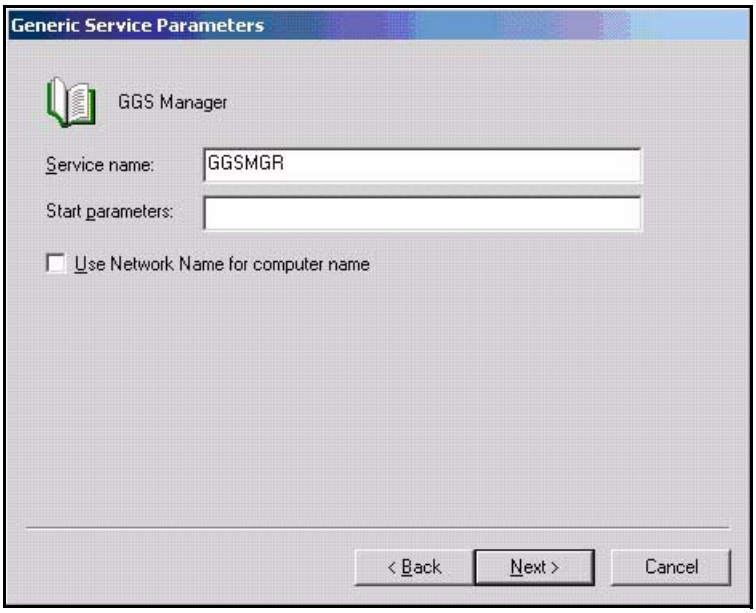


5. Click **Next**.
6. In the GGS Manager Service Properties dialog box, click the Dependencies tab, and add the following to the Resource dependencies list:
   - The database resource group (in this example, it is "Database")

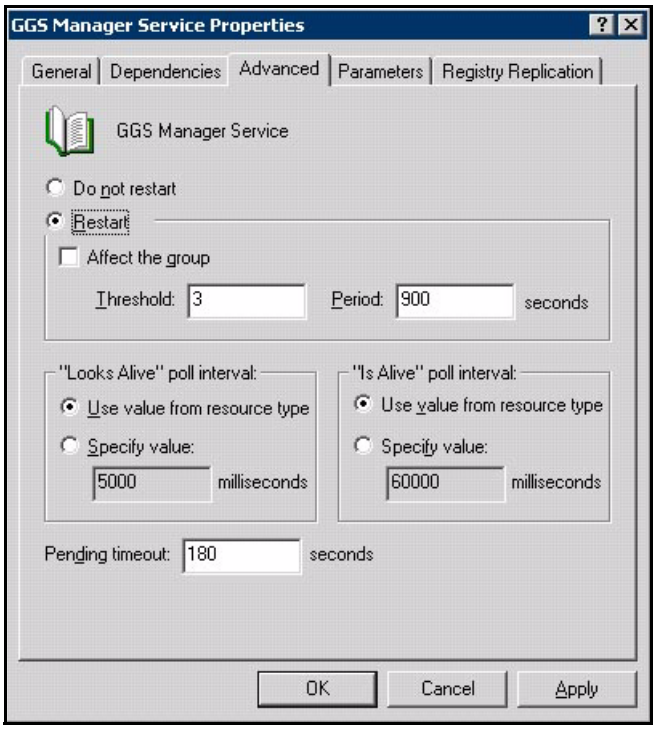
- The disk resource containing the GoldenGate directory
- The disk resource containing the database transaction log files
- The disk resource containing the database transaction log backup files



7. Click **Apply**, then **OK**.
8. In the Generic Service Parameters dialog box, type either the default Manager service name of GGSMGR or, if applicable, the custom name specified in the GLOBALS file.



9. Click **Next**.
10. Click **Finish** to exit the wizard.
11. In the Cluster Administrator tree, right-click the Manager resource and select Properties.
12. Click the Advanced tab, and deselect Affect the Group. This is a recommendation, but you can configure it as needed for your environment.



13. Click **Apply**.
14. Bring the cluster resource online to verify that it was installed correctly.
15. Take the resource offline again.
16. Move the group to the next node in the cluster. When the group has been successfully moved to the second node, the Manager resource should still be offline.
17. Log onto the second node.
18. Install GoldenGate Manager as a service on this node by running the **install** program as you did on the previous node. If you created a custom name for Manager in the GLOBALS file, that name will be used.
19. Bring the resource online to verify that it is running correctly on this node.
20. Repeat steps 18 through 22 for each additional node in the cluster.

Configuring Manager and other processes

- To use GoldenGate, you must configure the Manager process. You must specify a TCP/IP port for Manager to use, and you can specify optional parameters that control dynamic port assignments, trail file maintenance, and other properties.
- To begin using GoldenGate, you need to create and configure at least one Extract and Replicat group. Your instructions for these groups determine which data to capture and replicate, and how that data is processed.
- To configure these processes, and to customize GoldenGate, see the *GoldenGate for Windows and UNIX Administrator Guide*.

CHAPTER 3

Preparing the system for GoldenGate

Preparing tables for processing

The following table attributes must be addressed in a GoldenGate environment.

Assigning row identifiers

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

Different SQL Server versions may have different requirements with respect to row identifiers:

- **SQL Server 2000**: The source tables can have any type of key listed in "How GoldenGate determines the kind of row identifier to use".
- **SQL Server 2005 prior to Cumulative Update 6**: All of the source tables must have a *primary key*. This is a requirement of the SQL Server Replication component, which GoldenGate uses as part of its capture methodology.
- **SQL Server 2005 with Cumulative Update 6 or higher**: Source tables can have any kind of key listed in "How GoldenGate determines the kind of row identifier to use".

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

How GoldenGate determines the kind of row identifier to use

GoldenGate selects a row identifier to use in the following order of priority:

1. Primary key
2. First unique key alphanumerically that does not contain a timestamp or non-materialized computed column
3. If none of the preceding key types exist (even though there might be other types of keys defined on the table) 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 GoldenGate in a key or those that are excluded from the GoldenGate configuration.

> **NOTE** If there are other, non-usable keys on a table (such as one that includes a virtual column), or if there are no keys at all on the table, GoldenGate logs an appropriate message to the report file. Constructing a key of all of the columns impedes the

performance of GoldenGate on the source system. On the target, this key causes Replicat to use a larger, less efficient WHERE clause.

How to specify your own key for GoldenGate to use

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

Constraining updates when a table has no key

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

Disabling triggers and cascade delete constraints

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

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

Preparing IDENTITY columns for replication

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

Using IDENTITY columns in a bidirectional configuration

For a bidirectional SQL Server configuration, define IDENTITY columns to have an increment value equal to the number of servers in the configuration, with a different seed value for each one. For example, a two-server installation would be as follows:

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

A three-server installation would be as follows:

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

Replicating to an IDENTITY column

Because only one table in a session can have IDENTITY_INSERT set to ON, Replicat might need to continuously toggle IDENTITY_INSERT off and on when processing different tables that contain IDENTITY columns. To improve the performance of Replicat in this situation, use the BATCHSQL parameter. BATCHSQL causes Replicat to use array processing instead of applying SQL statements one at a time.

Configuring an ODBC connection

The GoldenGate Extract and Replicat processes use ODBC (Open Database Connectivity) to connect to the source and target databases. Follow these guidelines to make certain these processes can connect successfully.

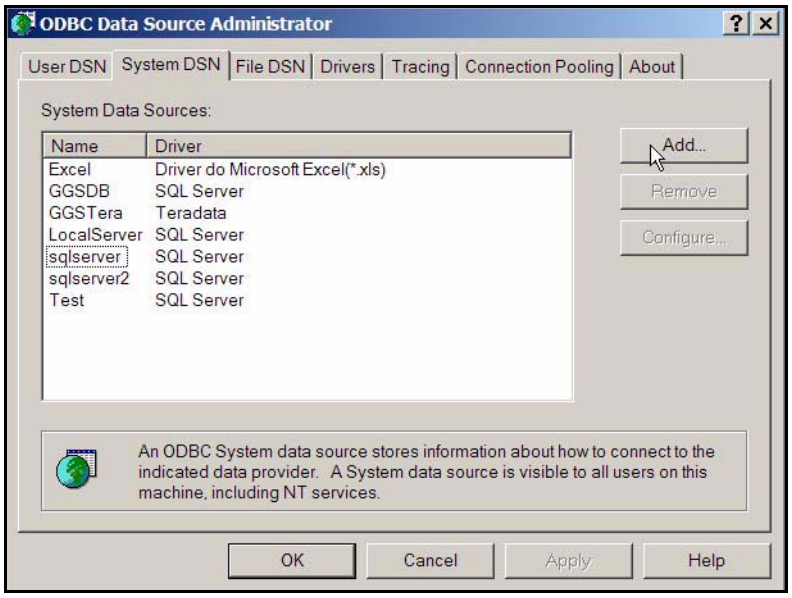
Creating a SQL Server DSN

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

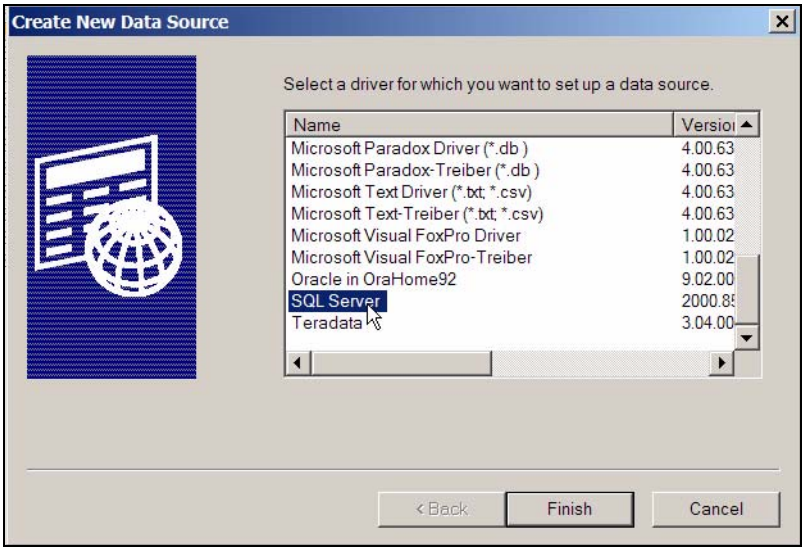
To create the DSN

1. Run one of the following ODBC clients:
   - If using a 32-bit version of GoldenGate on a 64-bit system, create the DSN by running the ODBCAD32.EXE client from the %SystemRoot%\Windows\SysWOW64 folder.
   - If using a 64-bit version of GoldenGate on a 64-bit system, create a DSN by running the default ODBCAD32.EXE client in **Control Panel**>**Administrative Tools**>**Data Sources (ODBC)**.
   - If using a version of GoldenGate other than the preceding, use the default ODBC client in **Control Panel>Administrative Tools>Data Sources (ODBC)**.

2. In the **ODBC Data Source Administrator** dialog box of the ODBC client, select the **System DSN** tab, and then click **Add**.

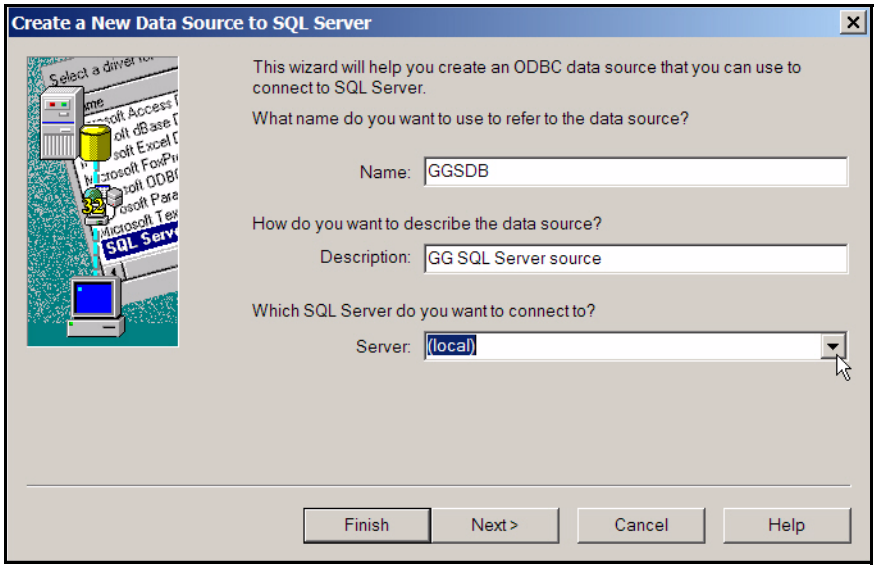


3. Under **Create New Data Source**, select the correct SQL Server driver as follows:
   - SQL Server 2000: SQL Server ODBC driver
   - SQL Server 2005: SQL Native Client driver
   - SQL Server 2008 target: SQL Server Native Client 10.0 driver

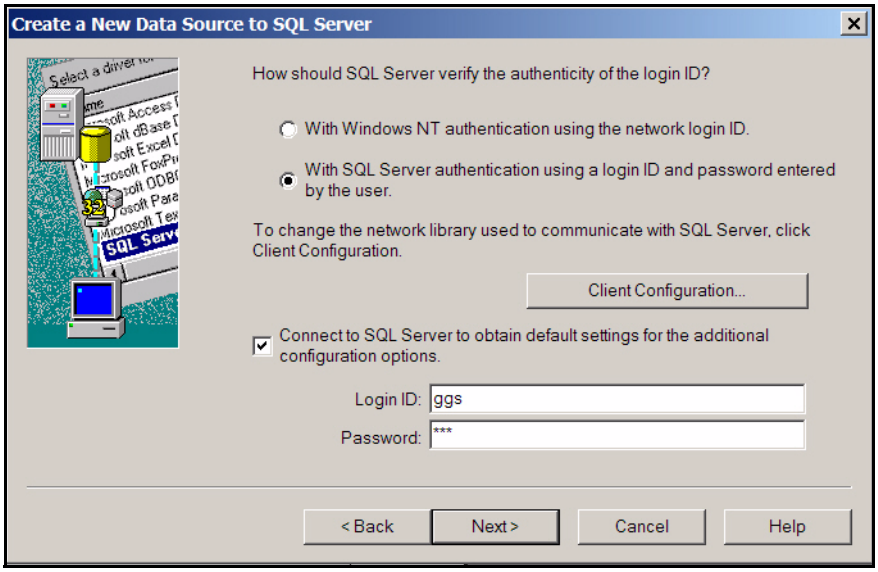


4. Click **Finish**. The **Create a New Data Source to SQL Server** wizard is displayed.

5. Supply the following:
   - **Name**: Can be of your choosing. In a Windows cluster, use one name across all nodes in the cluster.
   - **Description**: (Optional) Type a description of this data source.
   - **Which SQL Server do you want to connect to**: Select the server name.

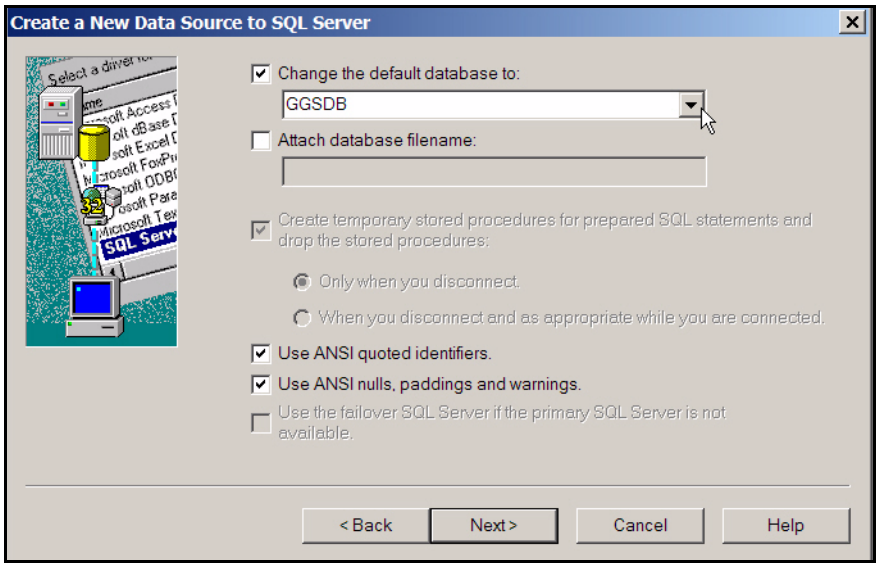


6. Click **Next**.
7. For login authentication, select **With Windows NT authentication** if the GoldenGate process will use a network login, or select **With SQL Server authentication using a login** if the process will use database credentials. Supply login information as appropriate.

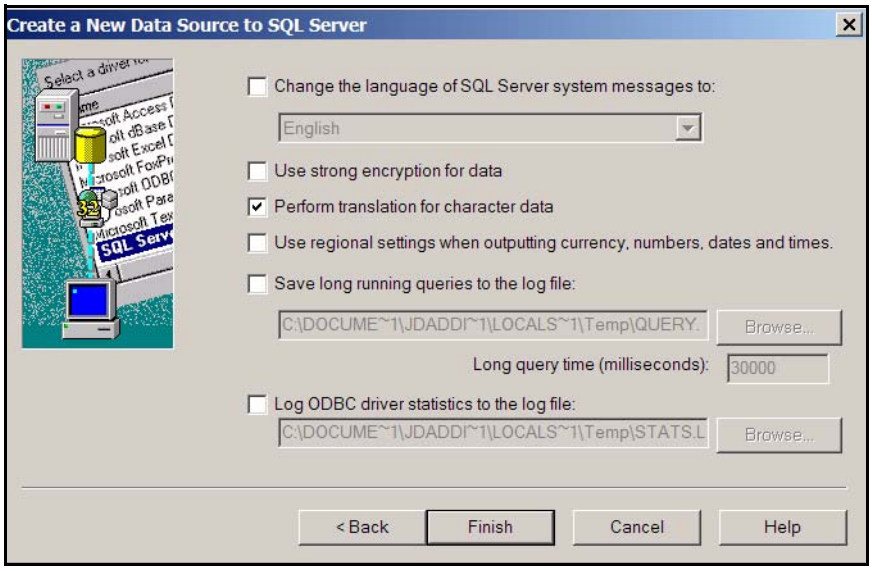


8. Click **Next**.

9. If the default database is not set to the one that GoldenGate will connect to, click **Change the default database to**, and then select the correct name. Set the other settings to use ANSI.

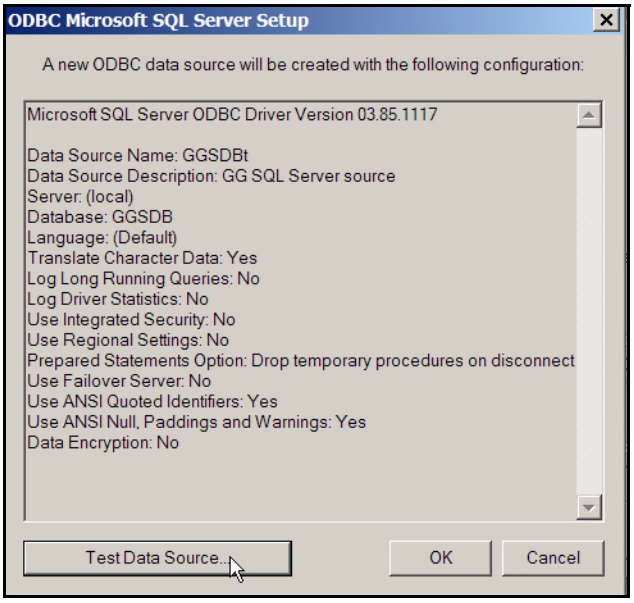


10. Click **Next**.
11. Leave the next page set to the defaults.



12. Click **Finish**.

13. Click **Test Data Source** to test the connection.



14. Close the confirmation box and the **Create a New Data Source** box.
15. Repeat this procedure from step 1 on the next system where SQL Server is installed.

Preventing multiple connections

By default, the Extract and Replicat processes create a new connection for catalog queries. You can prevent this extra connection by using the DBOPTIONS parameter with the NOCATALOGCONNECT option.

Configuring SQL Server logging

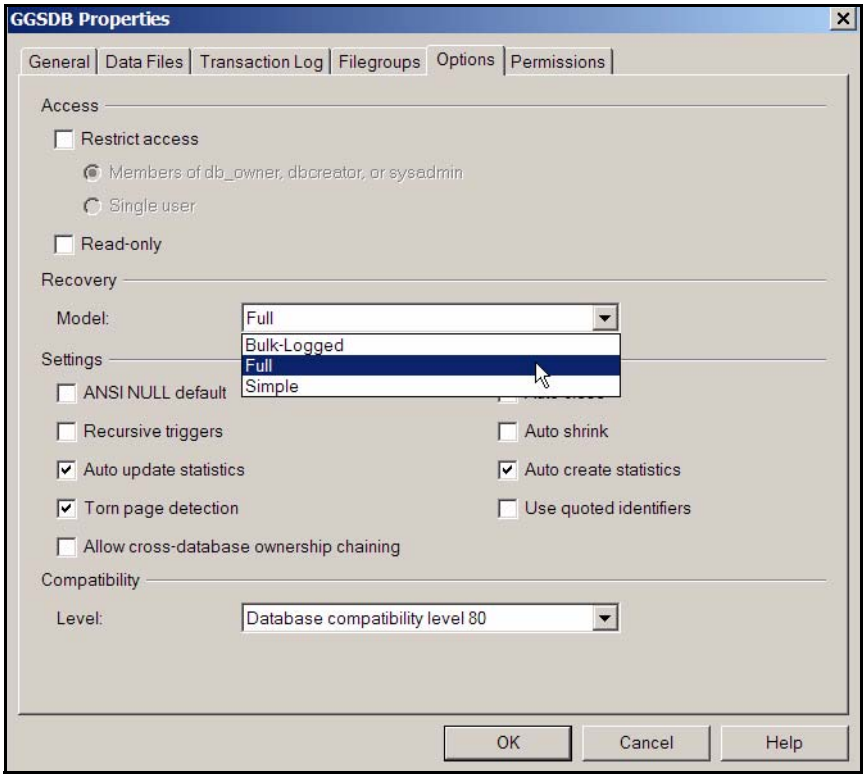
To support GoldenGate extraction on a SQL Server source system, do the following:

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

To verify or set the recovery model

1. Connect to the SQL Server instance with either Enterprise Manager for SQL Server 2000 or SQL Server Management Studio for SQL Server 2005/2000.
2. Expand the **Databases** folder.
3. Right-click the source database, and then select **Properties**.
4. Select the **Options** tab.

5. Under **Recovery**, set **Model** to **Full**.

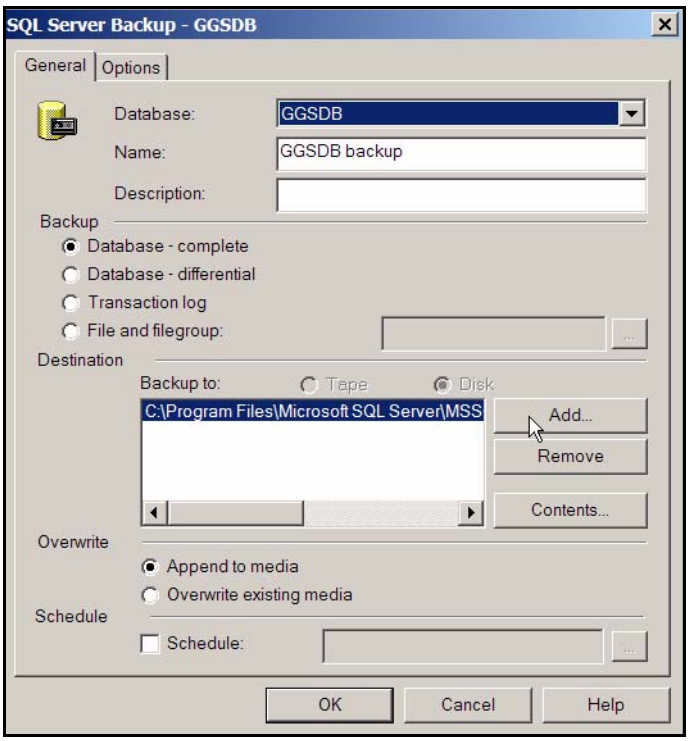


6. Click **OK**.

To make a full backup of the database

1. Right click the database name, and then select **All Tasks > Backup Database**.

2. Select **Database - Complete**. This option makes a full database backup and ensures that no transaction information is lost when GoldenGate starts.



3. Under **Destination**, click **Add** to specify the backup file name and location.
4. Click **OK**. The backup file is added to the **Destination** list box in the **SQL Server Backup** dialog box.
5. Click **OK** to start the backup.

Installing replication components (SQL Server 2005)

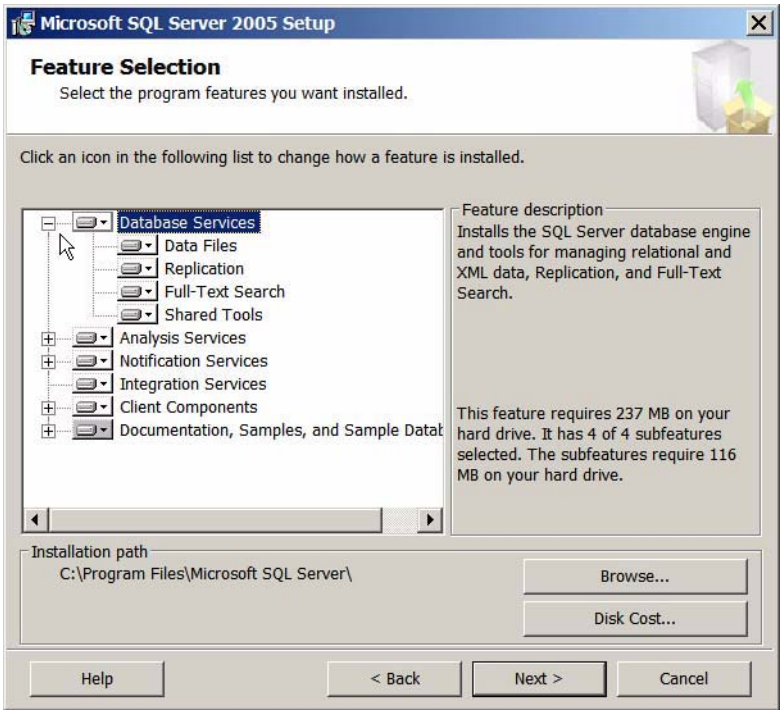
In the absence of the Microsoft Cumulative Update Package 6 (CU6) for SQL Server 2005 Service Pack 2, the SQL Server 2005 replication components must be used on the source system with a Distributor database to support GoldenGate replication. This configuration enables:

- Extract to capture logged SQL Server 2005 UPDATE operations in the absence of the CU6 package.
- GoldenGate to operate concurrently with SQL Server 2005 Replication against the same database. GoldenGate issues a warning message when it detects a Log Reader Agent that is already attached to the database.
- One Distributor to be used for all source databases. GoldenGate does not depend on the Distributor database, but rather reads the logs directly, so you can set transaction retention to zero.

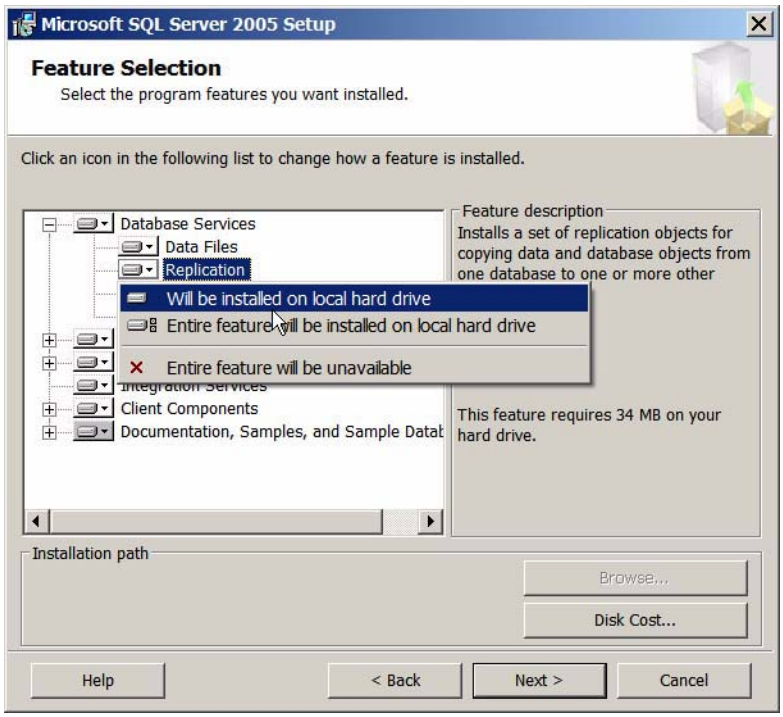
To install the replication components and Distributor database

NOTE Install the replication components only if they are not installed and configured already. If replication components and a distributor database already exist, skip to page 34 to enable extended logging.

1. On the source system, run Setup.exe in the Servers folder of the SQL Server installation directory.
2. Complete the initial licensing pages.
3. On the **Components to Install** page, select the database features that you want to install.
4. Click **Advanced** to open the **Feature Selection** page.
5. Expand **Database Services**.

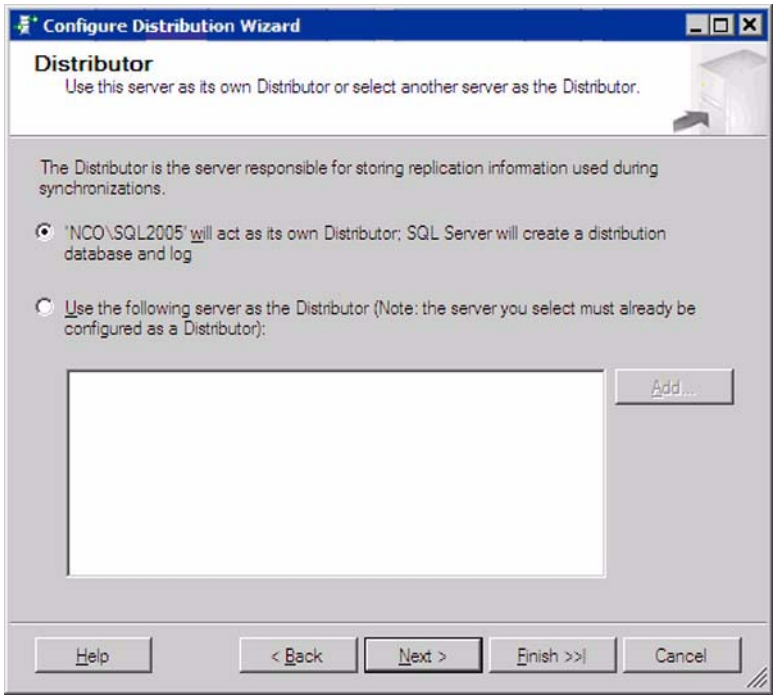


6. Click **Replication**, and then select **Will be installed on local hard drive** from the drop-down menu.



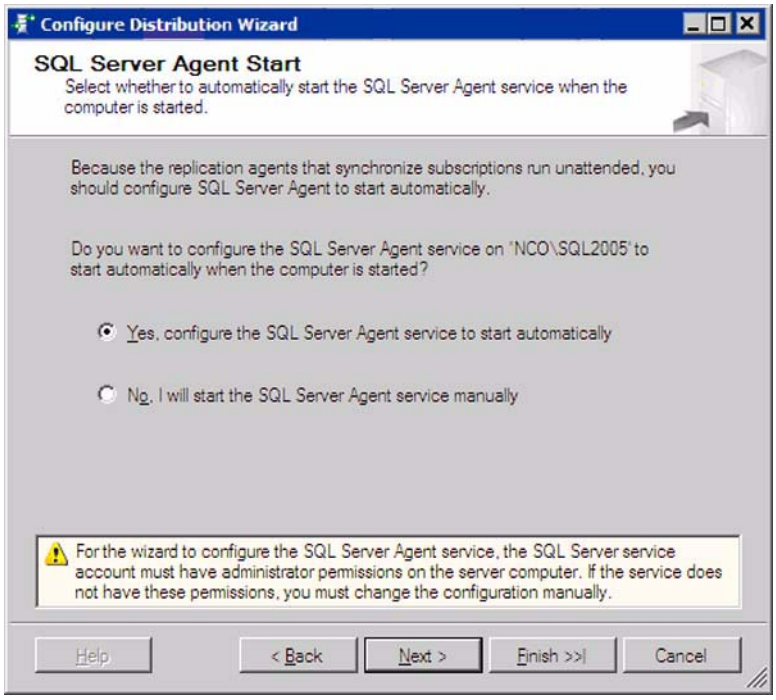
7. Click **Next**.
8. Complete the database setup according to your requirements.
9. Run SQL Server Management Studio.
10. Expand the SQL Server instance.
11. Select the **Replication** folder.
12. Right click **Replication**, and then select **Configure Distribution** to start the **Configure Distribution** wizard.

13. Select the local instance as its own Distributor, or select a remote Distributor.



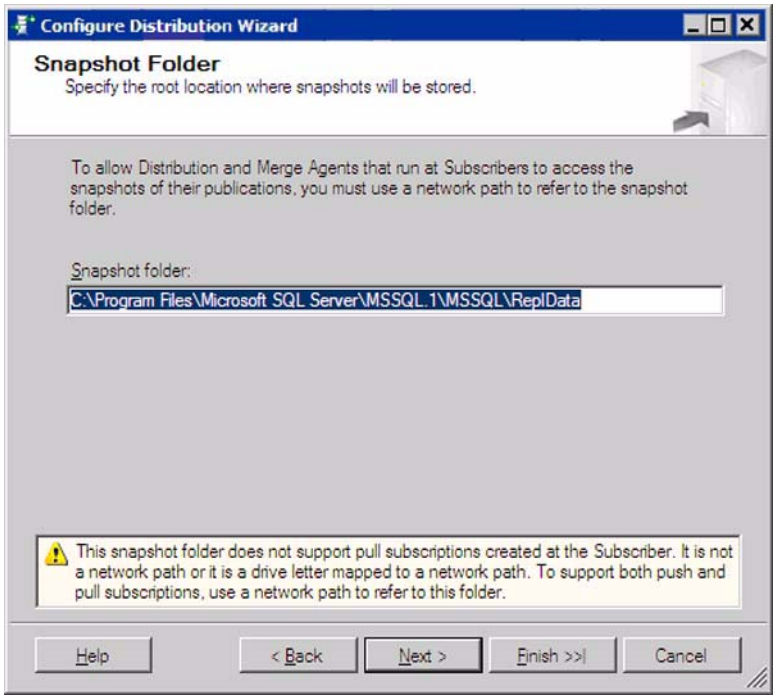
14. Click **Next**.

15. Set the SQL Server Agent service to start automatically, if possible.



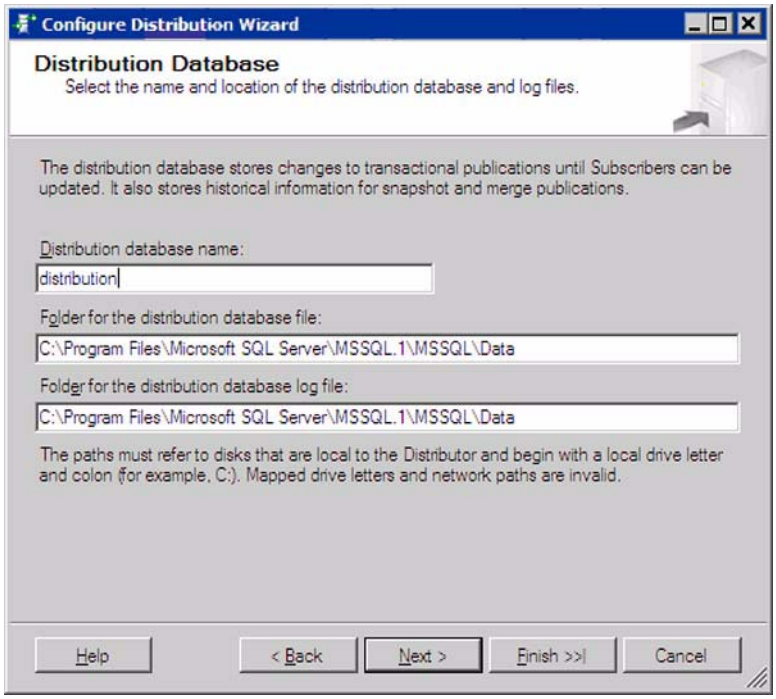
16. Click **Next**.

17. Accept the default Snapshot Folder, or choose a new location. GoldenGate does not use the Snapshot Folder.



18. Click **Next**.

19. Accept the default database name and file locations, or modify them as needed.



20. Click **Finish**, and then click **Finish** again to create the Distributor database and finish the setup.

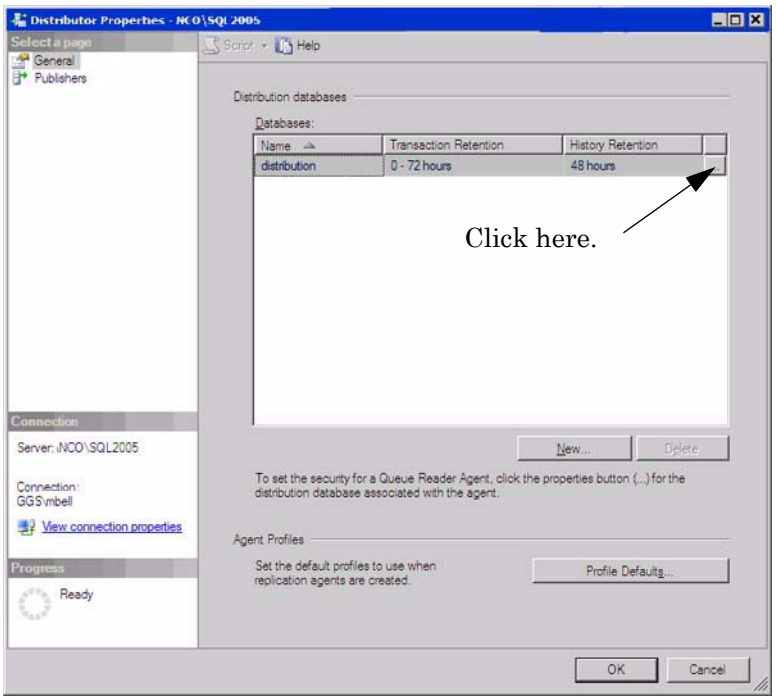
Configuring and cleaning up the Distributor database

After the Distributor database is installed, do the following:

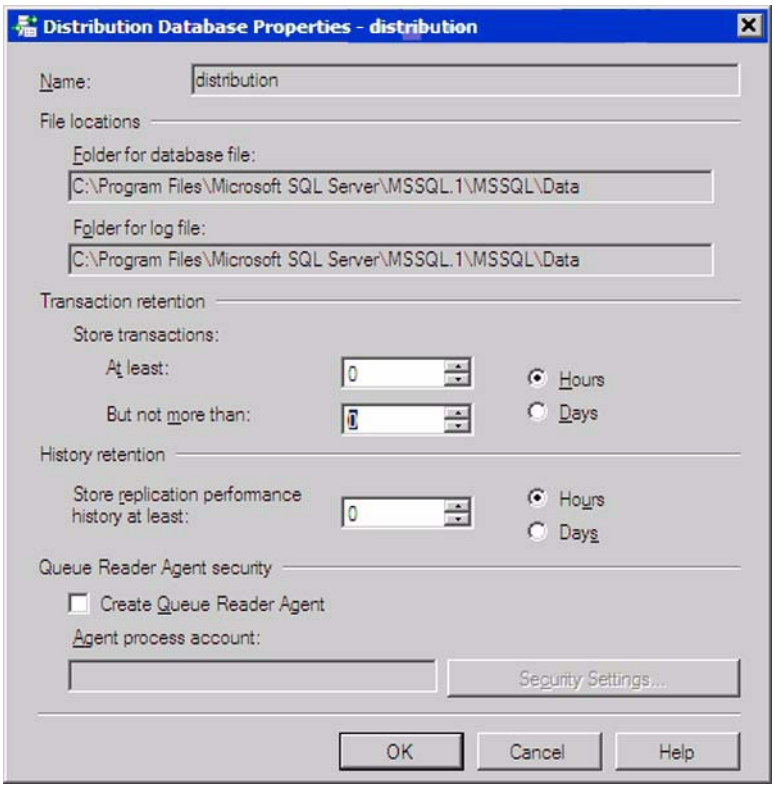
- Set transaction retention to 0.
- Disable replication alerts.
- Stop and disable SQL Server Agent replication jobs, which are created during the Distributor setups.

To configure transaction retention

1. In SQL Server Management Studio, expand the SQL Server instance.
2. Right-click the **Replication** folder, and then select **Distributor Properties**.
3. Click **General Properties**.
4. To the right of the **History Retention** column, next to the Distributor database name, click the ellipsis (...) button to open the **Distribution Database Properties**.



5. Set **Transaction retention** to:
   - **At least 0 Hours**
   - **But not more than 0 Hours**



6. On the same page, set **History retention** to 0.
7. Click **OK**.

To disable SQL Server Agent replication jobs and alerts

1. In SQL Server Management Studio, connect to the SQL Server instance.
2. Start SQL Server Agent.
3. Expand the **SQL Server Agent** folder, and then expand the **Jobs** folder.
4. For the following jobs, right click and then select **Disable**.
   - Agent history clean up: <Distributor database name>
   - Distribution clean up: <Distributor database name>
   - Expired subscription clean up
   - Reinitialize subscriptions having data validation failures
   - Replication agents checkup
5. Under the **SQL Server Agent** folder, expand the **Alerts** folder.
6. Right click all alerts that begin with the name "Replication," and then select Disable.

Enabling extended logging

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

ADD TRANDATA does the following:

- If the source database is SQL Server 2000, ADD TRANDATA sets a flag on the sysobjects table that tells SQL Server to log full before and after images.

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

- If the source database is SQL Server 2005 with the CU6 update package installed, ADD TRANDATA calls the sys.sp_extended_logging stored procedure.
- If the source database is SQL Server 2005 without CU6, ADD TRANDATA creates the following:
  - A Replication Publication named [<source database name>]: GoldenGate<source database name> Publisher. To view this publication, look under Replication>Local Publications in SQL Server Management Studio. This procedure adds the specified table to the publication as an article.
  - A SQL Server Log Reader Agent job for the publication. Because this job cannot run concurrently with the Extract process, this procedure includes steps for stopping and disabling it.

To enable extended logging

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

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

   Where:

   - SOURCEDB <DSN> is the name of the data source.
   - USERID <user>, PASSWORD <password> is the Extract login and password, if Extract uses SQL Server authentication. Can be the Extract user or a member of an account in the System Administrators or Server Administrators fixed server role.

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

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

To disable the Log Reader Agent Job (SQL Server 2005 using native replication only)

1. In SQL Server Management Studio, connect to the SQL Server 2005 instance.

2. Start SQL Server Agent.
3. Expand the **SQL Server Agent** folder.
4. Expand the **Jobs** folder.
5. Find the job that was created by the ADD TRANDATA command. The name shows the server, the instance, and the database name, plus the number of the publication. It looks similar to the following:

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

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

Maintaining the secondary truncation point (SQL Server 2005)

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

SQL Server replication maintains the truncation point

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

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

GoldenGate maintains the truncation point

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

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

If keeping Extract

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

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

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

If removing Extract

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

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

     ```
     EXEC sp_dboption '<Database Name>', 'Publish', 'false'
     ```

   - In the SQL Server Management Studio, delete the publication for the database from which Extract was capturing.

Specifying the location of the online and backup logs

GoldenGate supports database environments in which the online and backup logs are in non-default locations. The following options of the TRANLOGOPTIONS parameter enable you to point GoldenGate to these logs and ensure access to them. For syntax and usage, see the *GoldenGate for Windows and UNIX Reference Guide*.

Specifying an alternate location for the online logs

Point Extract to the alternate location of the online logs by using TRANLOGOPTIONS with the ALTONLINELOGS option. Multiple online log files can be specified.

Specifying an alternate location for the backup logs

By default, the Extract process reads the backup logs if the online logs do not contain the records that need to be processed. This can happen if Extract lags too far behind in the transaction stream, and the transactions get backed up and moved offline before they can be processed.

By default, Extract looks for the backup logs in the SQL Server default location. If you store the backups in a different location from the default, you must specify that location to Extract by using TRANLOGOPTIONS with the ALTARCHIVELOGDEST option in the Extract parameter file.

Invoking third-party restore utilities

If a third-party backup and restore utility is being used to manage the backup logs, you can configure Extract to invoke it to restore any needed backup logs during Extract processing. Use TRANLOGOPTIONS with ARCHIVERESTOREEXECUTABLE to specify the full path name of the executable file of the utility, and additionally use the ARCHIVERESTOREPARAMS option to provide parameters to the utility.

Sizing and retaining the logs

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

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

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

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

NOTE If the source is a SQL Server 2005 database without CU6, and if the Extract process needs to be suspended for a period of time longer than the normal backup frequency of the transaction log, re-enable and start the SQL Server Replication Log Reader job temporarily to manage the last distributed transaction. Stop and disable the job before you restart Extract.

Purging log archives

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

CHAPTER 4

Uninstalling GoldenGate

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

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

Uninstalling GoldenGate from Windows (non-cluster)

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

   ```
   install deleteevents deleteservice
   ```

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

Uninstalling GoldenGate from Windows Cluster

1. Working from the node in the cluster that owns the cluster group containing the Manager resource, run GGSCI and then stop any Extract and Replicat processes that are still running.
2. Use the Cluster Administrator tool to take the Manager resource offline.
3. Right click the resource and select **Delete** to remove it.
4. Run the install program using the following syntax.

   ```
   install deleteevents deleteservice
   ```

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

APPENDIX 1

GoldenGate installed components

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

GoldenGate Programs and Utilities

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

Table 6 Programs and utilities

Program	Description
cobgen	Generates source definitions based on COBOL layouts. Used for GoldenGate for Datawise on Stratus.
convchk	Converts checkpoint files to a newer version.
ddlcob	Generates target DDL table creation statements based on COBOL layouts. Used for GoldenGate for Datawise on Stratus.
ddlgen	Generates target database table definitions based on source database DDL.
defgen	Generates data definitions and is referenced by 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 extraction from database tables or transaction logs or receives transaction data from a vendor access module.
ggmxinstall	GoldenGate installation script for SQL/MX.
ggsci	User interface to GoldenGate for issuing commands and managing parameter files.

Table 6 Programs and utilities (continued)

Program	Description
ggsmgr.jcl ggsmgr.proc ggsmgrst.jcl ggsmgrst.proc	Start the GoldenGate Manager process from a batch job or the operator console on a z/OS system.
install	Installs 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 GoldenGate processes, reporting, and routing of requests through the GGSCI interface.
replicat	Applies data to target database tables.
reverse	A utility that reverses the order of transactional operations, so that Replicat can be used to back out changes from target tables, restoring them to a previous state.
server	The Collector process, an Extract TCP/IP server collector that writes data to remote trails.
triggen	Generates scripts that create the GoldenGate log table and logging triggers to support the trigger-based extraction method.
vamserv	Started by Extract to read the TMF audit trails generated by TMF-enabled applications using the NonStop SQL/MX database.

GoldenGate subdirectories

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

Table 7 Subdirectories

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

Table 7 Subdirectories (continued)

Directory	Description
dirpcs	Default location for status files. File name format is <group>.<extension> where <group> is the name of the group and <extension> is either pce (Extract), pcr (Replicat), or pcm (Manager). These files are only created while a process is running. The file shows the program name, the process name, the port number, and the process ID. Do not edit these files. Examples: mgr.pcm ext.pce
dirprm	The default location for GoldenGate parameter files created by GoldenGate users to store run-time parameters for 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 GoldenGate parameter values. They can be edited directly from a text editor or by using the EDIT PARAMS command in GGSCI. Examples: defgen.prm finance.prm
dirrec	Not used by GoldenGate.
dirrpt	The default location for process report files created by Extract, Replicat, and Manager processes to report statistical information relating to a processing run. Written in external ASCII format. File name format is <group name><sequence number>.rpt where <sequence number> is a sequential number appended to aged files. Do not edit these files. Examples: fin2.rpt mgr4.rpt
dirsql	The default location for scripts created by the TRIGGEN utility to contain SQL syntax for creating GoldenGate logging triggers and GoldenGate log tables. Written in external ASCII format. File name format is a user-defined name or the defaults of GGSLOG (table-creation script) or the table name (trigger-creation script), with the extension of .sql. These scripts can be edited if needed. Examples: ggslog.sql account.sql

Table 7 Subdirectories (continued)

Directory	Description
dirtmp	The default location for storing large transactions when the size exceeds the allocated memory size. Do not edit these files.
dirver	A GoldenGate Veridata directory. Not used unless this software is installed in the GoldenGate location.

Other GoldenGate files

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

Table 8 Other files

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

Table 8 Other files (continued)

Component	Description
ddl_informix.tpl	Template used by the DDLGEN utility to convert source DDL to Informix DDL.
ddl_mss.tpl	Template used by the DDLGEN utility to convert source DDL to SQL Server DDL.
ddl_mysql.tpl	Template used by the DDLGEN utility to convert source DDL to MySQL DDL.
ddl_nssql.tpl	Template used by the DDLGEN utility to convert source DDL to NonStop SQL DDL.
ddl_ora9.sql	A script that gets tablespace information from an Oracle 9 database.
ddl_ora10.sql	A script that disables the Oracle recyclebin and gets tablespace information from an Oracle 10 database.
ddl_oracle.tpl	Template used by the DDLGEN utility to convert source DDL to Oracle DDL.
ddl_pin.sql	Script that pins DDL tracing, the DDL package, and the DDL trigger for performance improvements. (Oracle installations)
ddl_remove.sql	Script that removes the DDL extraction trigger and package. (Oracle installations)
ddl_setup.sql	Script that installs the GoldenGate DDL extraction and replication objects. (Oracle installations)
ddl_sqlmx.tpl	Template used by the DDLGEN utility to convert Tandem Enscribe DDL to NonStop SQL/MX DDL.
ddl_status.sql	Script that verifies whether or not each object created by the GoldenGate DDL support feature exists and is functioning properly. (Oracle installations)
ddl_sybase.tpl	Template used by the DDLGEN utility to convert source DDL to Sybase DDL.
ddl_tandem.tpl	Template used by the DDLGEN utility to convert source DDL to NonStop SQL DDL.
ddl_tracelevel.sql	Script that sets the level of tracing for the DDL support feature. (Oracle installations)
debug files	Debug text files that may be present if tracing was turned on.

Table 8 Other files (continued)

Component	Description
demo_<db>_create.sql	Script that creates demonstration tables in the database associated with the GoldenGate installation.
demo_<db>_insert.sql	Script that inserts initial test data into the demonstration tables.
demo_<db>_misc.sql	Script that simulates transaction activity on the demonstration tables.
ENCKEYS	User-created file that stores encryption keys. Written in external ASCII format.
exitdemo.c	User exit example.
ggmessage.dat	Data file that contains error, informational, and warning messages that are returned by the 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 GoldenGate.
ggsmsg.dll	Windows dynamic link library used by the INSTALL program.
GLOBALS	User-created file that stores parameters applying to the GoldenGate instance as a whole.
help.txt	Help file for the GGSCI command interface.
LGPL.txt	Lesser General Public License statement. Applies to free libraries from the Free Software Foundation.
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 GoldenGate's 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 GoldenGate DDL marker table. (Oracle installations)

Table 8 Other files (continued)

Component	Description
marker_status.sql	Script that confirms successful installation of the DDL marker table. (Oracle installations)
odbcinst.ini	Ingres 2.6 on Unix ODBC configuration file.
params.sql	Script that contains configurable parameters for DDL support. (Oracle installations)
pthread-win32.txt	License agreement for pthread-VC.dll.
pthread-VC.dll	POSIX threads library for Microsoft Windows.
role_setup.sql	Script that creates the database role necessary for 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	z/OS paramlib members to start and stop the Manager process.
startmgr stopmgr	z/OS Unix System Services scripts to start the Manager process from GGSCI.
startmgrcom stopmgrcom	z/OS system input command for the Manager process.
tcperrs	File containing user-defined instructions for responding to TCP/IP errors.
usrdecs.h	Include file for user exit API.
zlib.txt	License agreement for zlib compression library.

GoldenGate checkpoint table

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

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

Table 9 Checkpoint table definitions

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

Index

A

ADD TRANDATA command 34
ADDEVENTS Windows service option 13
ADDSERVICE Windows service option 14
ALTARCHIVELOGDEST option, TRANLOGOPTIONS 36
ALTONLINELOGS option, TRANLOGOPTIONS 36
ARCHIVERESTOREEXECUTABLE option, TRANLOGOPTIONS 37
ARCHIVERESTOREPARAMS option, TRANLOGOPTIONS 37
AUTOSTART Windows service option 14

B

backup logs, see *logs*
BATCHSQL parameter 21
before images, logging 34
bulk copy 8

C

cascade deletes, disabling 20
case, supported 8
category.dll 13
characters
multi-byte 6
supported in object names 9
cluster, installing on 4, 12, 14
columns
datatypes supported 6
number of supported 7
components, GoldenGate 40
connections, to database 5
constraints, integrity 20

D

data source name (DSN), creating 21
data types, supported 6
database
Distributor, SQL Server 2005 27
requirements 5
user, GoldenGate 5
DBOPTIONS parameter 25
deletes, cascaded 20
disk requirements 3
Distributor database, configuring 27, 32
downloading GoldenGate 11

E

extended logging, enabling 34

F

files, installed by GoldenGate 40
firewall, configuring 4
FTP access for GoldenGate 4

G

ggmessage.dat file 46
GGSMGR default Manager name 12
ggsmsg.dll 13
GLOBALS file 13
GoldenGate
downloading 11
installed programs and files 40
installing 11
uninstalling 38

I

IDENTITY columns 7, 20
IMAGE data type 7
INFO EXTRACT command 37
installing GoldenGate 11

K

key
assigning 19
name, supported characters 9
tables without 20
TIMESTAMP in 6
KEYCOLS option, TABLE or MAP 20

L

large objects, limitations on 6
libraries, Visual C++ 5, 12
LIMITROWS option, DBOPTIONS 20
logging, configuring 25
logging, extended 34
logs
alternate locations for 36
configuring 22, 25
extended 34
non-native backups 8

M

Manager
as Windows service 13
multiple on same system 13
name, customizing 12
MANUALSTART Windows service option 14
MGRSERVNAME parameter 13
Microsoft Visual C ++ 2005 SP1 Redistributable Package 5, 12
multi-byte characters 6

N

name
non-supported characters in 10
supported characters in 9
names, supported 8
NOCATALOGCONNECT option, DBOPTIONS 25
NTEXT data type 7

O

objects, supported 7
ODBCAD32.EXE client 21
online logs, see *logs*
operating system requirements 3
operations, supported 7

P

PASSWORD Windows service option 14
permissions, database 5
platforms, supported 3
ports, required by GoldenGate 4

R

recovery model, setting 25
replication components, installing 27
rows, size supported 7

S

secondary truncation point, maintaining 35
spaces
in folder names 12
in object and column names 10
SQL Server Authentication, using 5
SQL_Variant data type 7

T

tables
preparing for processing 19
supported kinds 7
TCP/IP, configuring 4
TEXT data type 7
TextCopy utility 8
TIMESTAMP data type 6
TRANLOGOPTIONS parameter 35
transaction retention, configuring 32
triggers, disabling on target 20

U

uninstalling GoldenGate 38
UpdateText 8
USER Windows service option 14
user, database 5
USERID parameter 6

V

VAMSERV program 41
VARCHAR (MAX) data type 7
VARDECIMAL data type 7
virtual machine, support for 5
Visual C ++ 2005 SP1 Redistributable Package 5, 12

W

Windows Authentication, using 5
WriteText TSQL 8