Oracle® GoldenGate

DB2 LUW Installation and Setup Guide

Version 10.4

October 2009



Oracle® GoldenGate DB2 LUW 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 DB2 LUW 3
Supported platforms 3
Operating system requirements 3
Database requirements 5
Supported data types 6
Non-supported data types 6
Supported objects and operations 6
Non-supported objects and operations 6
Supported and non-supported object names and case 7
Object names and owners 7
Case sensitivity 7
Supported characters 7
Non-supported characters 8

Chapter 2 Installing GoldenGate 10

Installation overview 10
Upgrades 10
New installations 10
Downloading GoldenGate 10
Setting library paths for dynamic builds 10
Installing GoldenGate on Linux and UNIX 12
Installing into a UNIX or Linux cluster 12
Installing the GoldenGate files 12
Configuring Manager and other processes 12
Installing GoldenGate on Windows and Windows Cluster 13
Installing the Microsoft redistributable package 13
Installing GoldenGate into a Windows Cluster 13
Installing the GoldenGate files 13
Specifying a custom Manager name 13
Installing Manager as a Windows service 14

Adding GoldenGate as a Windows cluster resource 15
Configuring Manager and other processes 19

Chapter 3 Preparing the system for GoldenGate 20

Configuring the transaction logs for GoldenGate 20
Retaining the transaction logs 20
Specifying the archive path 21
Preparing tables for processing 21
Disabling triggers and cascade delete constraints 21
Assigning row identifiers 22
Preventing key changes 22
Enabling change capture 22
Maintaining materialized query tables 23
Preparing for initial extraction 23

Chapter 4 Uninstalling GoldenGate 25

Uninstalling GoldenGate from Linux or UNIX 25
Uninstalling GoldenGate from Windows (non-cluster) 25
Uninstalling GoldenGate from Windows Cluster 26

Appendix 1 GoldenGate installed components 27

GoldenGate Programs and Utilities 27
GoldenGate subdirectories 29
Other GoldenGate files 31
GoldenGate checkpoint table 35

Index 36

CHAPTER 1

System requirements and preinstallation instructions

Overview of GoldenGate for DB2 LUW

With GoldenGate for DB2 LUW, you can replicate data to and from similar or dissimilar supported DB2 LUW versions, or you can move data between a DB2 LUW database and a database of another type. GoldenGate for DB2 LUW supports data filtering, mapping, and transformation of data unless otherwise noted in this documentation.

Supported platforms

DB2-UDB 8.x and 9.5 on Windows and UNIX

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 privileges

- To install on Windows, the user installing GoldenGate must log in as Administrator.
- To install on UNIX, the user installing GoldenGate must have read and write privileges on the GoldenGate installation directory.
- The GoldenGate processes require an operating system user that has privileges to read, write, and delete files and subdirectories in the GoldenGate directory. In addition, the user for the Manager process requires privileges to control GoldenGate processes.
- It is recommended that these operating system users be dedicated to GoldenGate. Sensitive information might be available to anyone running a GoldenGate process.
- To configure Extract and Replicat to access the database using operating system authentication, see “Database user” on page 5 for additional requirements.

Third-party programs

- 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.
- 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 process calls the DB2READLOG function in the Administrative API to read the transaction log files of a DB2 LUW source database. In addition to DB2READLOG, Extract uses a small number of other API routines to check the source database configuration on startup.
- The GoldenGate Replicat process uses the DB2 CLI interface on a DB2 LUW target database. For instructions on installing this interface, see the DB2 documentation.
- One of the following command interfaces must be installed: DB2 Control Center, Command Center, Command Line.

Database user

- Create a database user that is dedicated to GoldenGate. It can be the same user for all of the GoldenGate processes that must connect to a database:
  - Extract (source database)
  - Replicat (target database)
  - DEFGEN (source or target database)
- To preserve the security of your data, and to monitor GoldenGate processing accurately, do not permit other users, applications, or processes to log on or operate as the GoldenGate database user.
- Assign system administrator (SYSADM) or database administrator (DBADM) authority to the database user under which Extract runs. To give the Extract user DBADM authority, a user with SYSADM authority can issue the following grant statement.

  ```
  GRANT DBADM ON DATABASE TO USER <user>
  ```

  This authority can also be granted from the User and Group Objects folder in the DB2 Control Center. The database tab for the user that is assigned to a GoldenGate process should have the Database Administrative Authority box checked.

  NOTE If the Extract user does not have the required authority, Extract will log the following errors and stop.

  ```
  [SC=-1224:SQL1224N A database agent could not be started to
  service a request, or was terminated as a result of a database
  system shutdown or a force command.
  SQL STATE 55032: The CONNECT statement is invalid, because the
  database manager was stopped after this application was
  started]
  ```

- Grant at least the following privileges to the database user under which Replicat runs:
  - Local CONNECT to the target database
  - SELECT on the system catalog views
  - SELECT, INSERT, UPDATE, and DELETE on the target tables

Supported data types

- GoldenGate supports all DB2 LUW data types, except those listed in "Non-supported data types".

Limitations of support

- GoldenGate supports multi-byte character DB2 LUW 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.
- BLOB and CLOB columns must have a LOGGED clause included in their definitions.
- 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.

Non-supported data types

- User-defined types

Supported objects and operations

- Extraction and replication of DML operations on DB2 tables that contain rows of up to 512 KB in length. This size exceeds the maximum row size of DB2. 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.
- Multi Dimensional Clustered Tables (MDC) for DB2 LUW 9.5 and later.
- Materialized Query Tables. GoldenGate does not replicate the MQT itself, but only the base tables. The target database automatically maintains the content of the MQT based on the changes that are applied to the base tables by Replicat.
- Extraction and replication of compressed data (CREATE TABLESPACE COMPRESS YES).

Non-supported objects and operations

- Multiple instances of a database
- Datalinks
- Extraction or replication of DDL (data definition language) operations
- Tables with VALUE COMPRESSION. To include such tables in the GoldenGate configuration, deactivate the value compression and then reorganize the tables.

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 1 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

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

Character	Description
#	Pound symbol
$	Dollar symbol
%	Percent symbol
^	Carot symbol
( )	Open and close parentheses
_	Underscore
-	Dash
+	Plus sign
=	Equal symbol
|	Pipe
[ ]	Begin and end brackets
{ }	Begin and end curly brackets (braces)

[1] The type of key that is being used by 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 2 Non-supported characters in object and column names[1]

Character	Description
&	Ampersand

Table 2 Non-supported characters in object and column names[1]

Character	Description
*	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

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://download.oracle.com/docs/cd/E15881_01/index.htm.

New installations

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

- Downloading GoldenGate
- Setting library paths for dynamic builds
- Installing the software

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

Downloading GoldenGate

1. Navigate to http://edelivery.oracle.com.
2. Select a language.
3. Click **Continue**.
4. Fill in the **Export Validation** form and check the box to agree to the license agreement.
5. Click **Continue**.
6. Select the **Oracle Fusion Middleware** Product Pack, and select the platform on which you will be installing the GoldenGate software.
7. Click **Go**. A list of Oracle Fusion Middleware products for the selected platform is displayed.
8. Select the appropriate Oracle GoldenGate Media Pack.

9. Click **Continue**. The Media Pack contents are displayed, including software builds and documentation.
10. Click **Download** next to each desired component, and follow the automatic download process to transfer the mediapack.zip file to your system.

Setting library paths for dynamic builds

As of version 10, GoldenGate uses shared libraries. When installing GoldenGate on a UNIX system, the following must be true *before running GGSCI or any GoldenGate process*.

1. Make certain that the database libraries are added to the system's shared-library environment variables. This procedure is usually performed at database installation time. Consult your Database Administrator if you have any questions.
2. If you will be running a GoldenGate program from outside the GoldenGate installation directory on a UNIX system:
   - (Optional) Add the GoldenGate installation directory to the PATH environment variable.
   - (Required) Add the GoldenGate installation directory to the shared-libraries environment variable.

   For example, given a GoldenGate installation directory of /ggs/10.0, the second command in the following table requires these variables to be set:

Command	Requires GG libraries in environment variable?
$ ggs/10.0 > ./ggsci	No
$ ggs > ./10.0/ggsci	Yes

To set the variables in Korn shell

```
PATH=<installation directory>:$PATH
export PATH
<shared libraries variable>=<absolute path of installation directory>:$<shared libraries variable>
export <shared libraries variable>
```

To set the variables in Bourne shell

```
export PATH=<installation directory>:$PATH
export <shared libraries variable>=<absolute path of installation directory>:$<shared libraries variable>
```

To set the variables in C shell

```
setenv PATH <installation directory>:$PATH
setenv <shared libraries variable> <absolute path of installation directory>:$<shared libraries variable>
```

Where: <shared libraries variable> is one of the following:

UNIX/Linux library path variables per platform

Platform[1]	Environment variable
◆ IBM AIX ◆ IBM z/OS	LIBPATH
HP-UX	SHLIB_PATH
◆ Sun Solaris ◆ HP Tru64 (OSF/1) ◆ LINUX	LD_LIBRARY_PATH

[1] A specific platform may or may not be supported by GoldenGate for your database. See the Systems Requirements for supported platforms.

Example `export LD_LIBRARY_PATH=/ggs/10.0:$LD_LIBRARY_PATH`

NOTE To view the libraries that are required by a GoldenGate process, use the ldd <process> shell command before starting the process. This command also shows an error message for any that are missing.

Installing GoldenGate on Linux and UNIX

Installing into a UNIX or Linux cluster

- To install GoldenGate into a cluster environment, install the GoldenGate binaries and files on a file system that is available to all cluster nodes, according to the directions that follow.
- After installing GoldenGate, configure the GoldenGate Manager process within the cluster application, as directed by the cluster documentation, so that GoldenGate will fail over properly with the other applications.

Installing the GoldenGate files

1. Extract the Oracle GoldenGate mediapack.zip file on a Windows system by using WinZip or an equivalent compression product. This produces a UNIX .tar file.
2. FTP the .tar file in binary mode to the UNIX system and directory where you want GoldenGate to be installed.
3. Extract the .tar file. The GoldenGate files are extracted into the current directory.

   ```
   tar -xvof <filename>.tar
   ```

4. Run the command shell and change directories to the new GoldenGate directory.
5. From the GoldenGate directory, run the GGSCI program.

   ```
   GGSCI
   ```

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

   ```
   CREATE SUBDIRS
   ```

7. Issue the following command to exit GGSCI.

```
EXIT
```

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*.

Installing GoldenGate on Windows and Windows Cluster

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) 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 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 3 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
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.

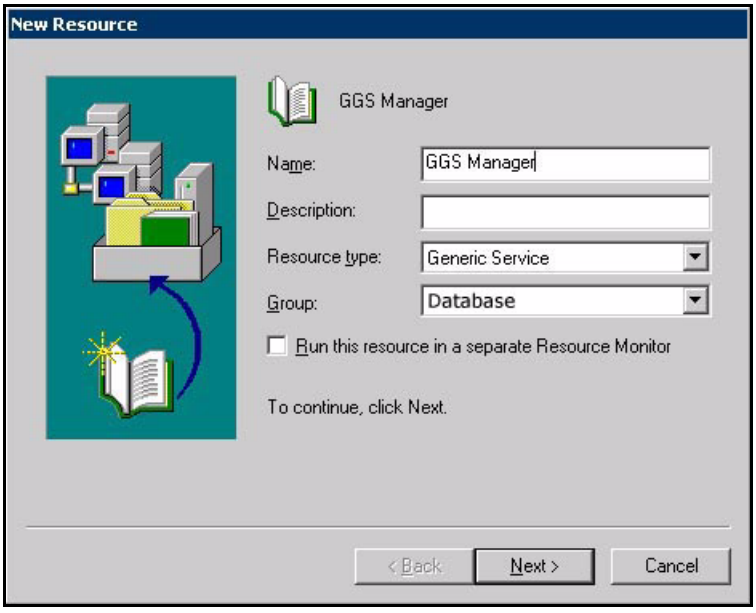
[1] A user account can be changed by selecting the Properties action from the Services applet of the Windows Control Panel.

Adding GoldenGate as a Windows cluster resource

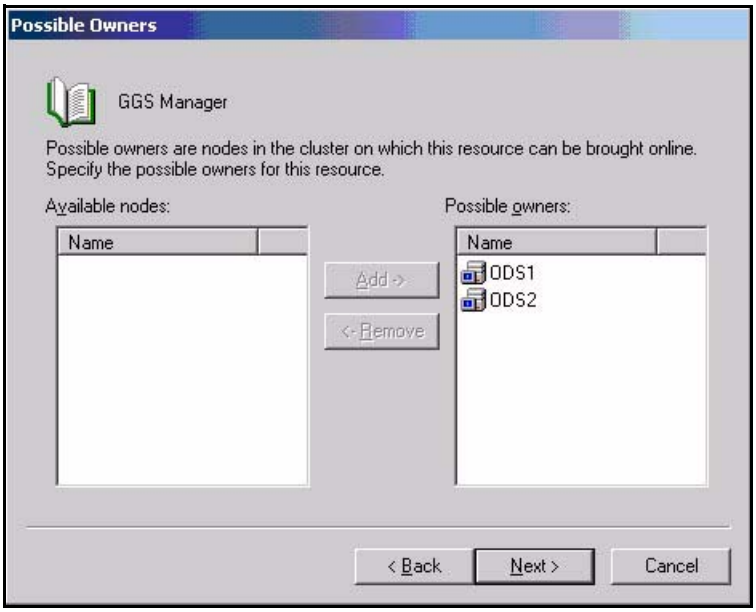
If you installed GoldenGate into a cluster, follow these instructions to establish GoldenGate as a cluster resource and configure the Manager service correctly on all nodes.

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.

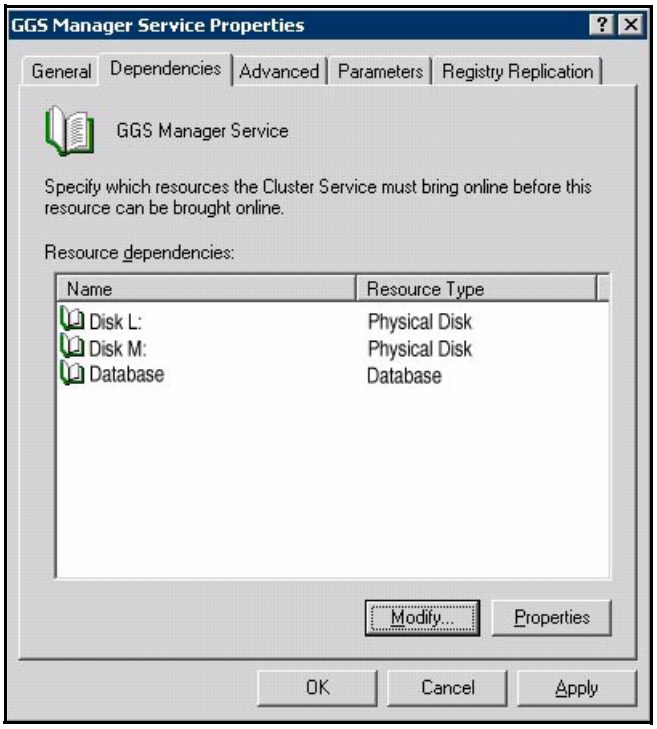


3. Click **Next**.
4. In the Possible Owners dialog box, select the nodes on which GoldenGate will run.

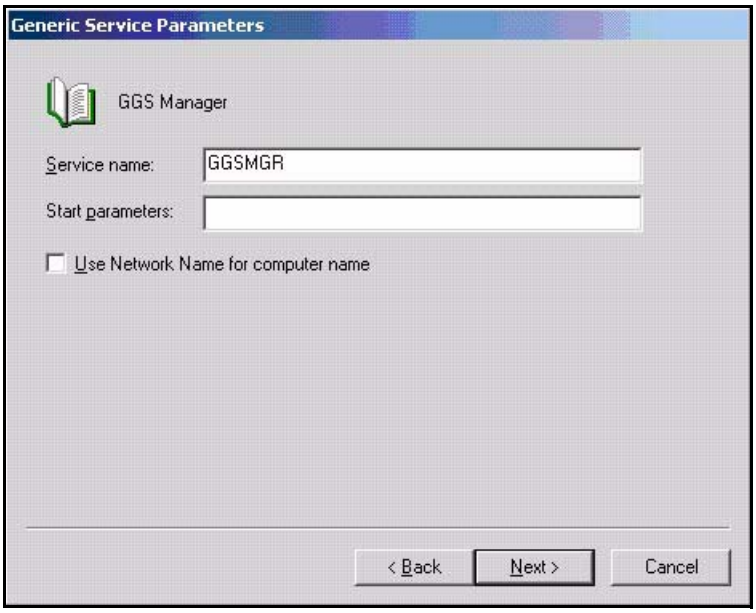


5. Click **Next**.
6. In the GGS Manager Service Properties dialog box, click the Dependencies tab, and add the following to the Resource dependencies list:
   - The database resource group (in this example, it is "Database")

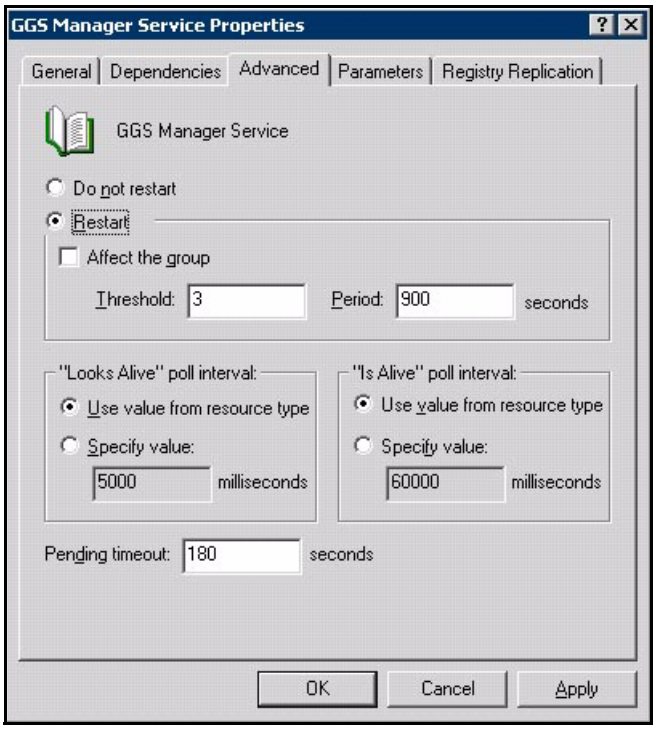
- The disk resource containing the GoldenGate directory
- The disk resource containing the database transaction log files
- The disk resource containing the database transaction log backup files



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

Configuring the transaction logs for GoldenGate

To capture DML operations, GoldenGate reads the online logs by default, but will read the archived logs if an online log is not available. To ensure the continuity and integrity of GoldenGate processing, configure the logs as follows.

Retaining the transaction logs

Configure the database to retain the transaction logs instead of recycling them.

- Set the LOGRETAIN parameter to RECOVERY, which retains the logs and enables them to be used for forward recovery.
- Turn on the USEREXIT parameter, which forces log retention regardless of LOGRETAIN and forces a user exit program to archive and retrieve the log files.

To determine log retention parameters

1. Connect to the database.

   ```
   db2 connect to <database> user <username> using <password>
   ```

2. Get the database name.

   ```
   db2 list db directory
   ```

3. Get the database configuration for the database.

   ```
   db2 get db cfg for <database name>
   ```

   The correct configuration for GoldenGate is:

   ```
   Log retain for recovery status = RECOVERY
   User exit for logging status = YES
   ```

To set log retention parameters

1. Issue the following commands.

   ```
   db2 update db cfg for <database name> using LOGRETAIN ON
   db2 update db cfg for <database name> using USEREXIT ON
   ```

2. Make a full backup of the database by issuing the following commands.

```
db2 stop force
db2 start
db2 backup db <database>
```

3. Place the backup in a directory to which DB2 LUW has access rights. Contact your systems administrator if you get the following message:

```
SQL2061N An attempt to access media "/home/devmgr" is denied.
```

Specifying the archive path

Set the DB2 OVERFLOWLOGPATH parameter to the archive log directory. The node attaches automatically to the path variable that you specify, much like the archive log directory is set.

To set the parameter

```
db2 connect to <database>
db2 update db cfg using overflowlogpath "<path>"
```

Exclude the node itself from the path. For example, if the full path to the archive log directory is /sdb2logarch/oltpods1/archive/OLTPODS1/NODE0000, the OVERFLOWLOGPATH value should be specified as /sdb2logarch/oltpods1/archive/OLTPODS1.

Preparing tables for processing

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

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.

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.

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.

Preventing key changes

Do not add columns to a key after GoldenGate starts extracting data from the table. This rule applies to a primary key, a unique key, a KEYCOLS key, or an all-column key. DB2 LUW does not supply a before image for columns that are added to a table. If any columns in a key are updated on the source, GoldenGate uses the before image to compare with the current values in the target table when it replicates the update.

Enabling change capture

Configure DB2 to log data changes in the expanded format supplied by the DATA CAPTURE CHANGES feature of the CREATE TABLE and ALTER TABLE commands. This format provides GoldenGate with the entire before and after images of rows that are changed with update statements. You can use GGSCI to issue the ALTER TABLE command.

To verify change capture status

Issue the following command from the DB2 command line processor (CLP).

```
Select NAME, CREATOR, DBNAME, DATA_CAPTURE
From SYSIBM.SYSTABLES
Where DATA_CAPTURE <> 'N' Order by CREATOR, NAME;
```

To enable change capture from GGSCI

1. From the GoldenGate directory, run GGSCI.
2. Log on to DB2 from GGSCI as a user that has ALTER TABLE privileges.

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

3. Issue the following command. You can use a wildcard to specify multiple table names.

```
ADD TRANDATA <table>
```

ADD TRANDATA issues the following command, which includes logging the before image of LONGVAR columns:

```
ALTER TABLE < name> DATA CAPTURE CHANGES INCLUDE LONGVAR COLUMNS;
```

To exclude LONGVAR logging

To omit the INCLUDE LONGVAR COLUMNS clause from the ALTER TABLE command, use ADD TRANDATA with the EXCLUDELONG option.

```
ADD TRANDATA <table>, EXCLUDELONG
```

NOTE If LONGVAR columns are excluded from logging, GoldenGate features that require before images, such as the GETUPDATEBEFORES, NOCOMPRESSUPDATES, and NOCOMPRESSDELETES parameters, might return errors if tables contain those columns. For a workaround, see the REQUIRELONGDATACAPTURECHANGES | NOREQUIRELONGDATACAPTURECHANGES options of the TRANLOGOPTIONS parameter.

Maintaining materialized query tables

To maintain parity between source and target materialized query tables (MQT), you replicate the base tables, not the MQT. The target database maintains the MQT based on the changes applied to the base tables by GoldenGate.

The following are the rules for configuring these tables:

- Include the base tables in your TABLE and MAP statements.
- Do not include MQTs themselves in the TABLE and MAP statements.
- Wildcards can be used in TABLE and MAP statements, even though they might resolve MQT names along with regular table names. GoldenGate automatically excludes MQTs from wildcarded table lists. However, any MQT that is explicitly listed in an Extract TABLE statement by name will cause Extract to abend.

Preparing for initial extraction

During the initialization of the GoldenGate environment, you will be doing an initial data synchronization and starting the GoldenGate processes for the first time. In conjunction with those procedures, you will be creating process groups. To create an Extract group, an initial start position must be established in the transaction log. This initial read position will be at a transaction boundary that is based on one of the following:

- a timestamp

- the end of the transaction file
- a specific LSN value

The start point is specified with the BEGIN option of the ADD EXTRACT command.

When Extract starts for the first time, it captures all of the transaction data that it encounters after the specified start point, but none of the data that occurred *before* that point. This can cause partial transactions to be captured if open transactions span the specified start point.

To ensure initial transactional consistency

To avoid the capture of partial transactions, the best practice is to initialize the Extract process at a point in time when the database is in a quiesced state. DB2 provides a QUIESCE command for such a purpose.

> **NOTE** After Extract is past the initialization, subsequent restarts of Extract will not extract partial transactions, because the process uses recovery checkpoints to mark its last read position.

To view open transactions

IBM provides a utility called db2pd for monitoring DB2 databases and instances. You can use it to view information about open transactions and to asses your exposure to transactions that could span the start point. However, due to the lack of timestamps on DB2 LUW log records, it might not be possible to determine your exposure precisely. The best practice is to quiesce the database prior to initialization of GoldenGate. If you cannot quiesce the database, please open a GoldenGate support ticket for assistance.

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.

Uninstalling GoldenGate from Linux or UNIX

1. Run the command shell.
2. (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.
3. Change directories to the GoldenGate installation directory.
4. Run GGSCI.
5. Stop all GoldenGate processes.
6. Stop the Manager process.
7. Exit GGSCI.
8. Remove the GoldenGate files by removing the installation directory.
9. Drop any GoldenGate-related objects from the database as needed.

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 4 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 4 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 5 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 5 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 5 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 6 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 6 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 6 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 6 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 7 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

Symbols

$LD_LIBRARY_PATH variable 11
$PATH variable 11

A

ADD TRANDATA command 23
ADDEVENTS Windows service option 14
ADDSERVICE Windows service option 15
API routines used by Extract 5
archive path, setting 21
AUTOSTART Windows service option 15

C

cascade deletes, disabling 21
case, supported 7
category.dll 14
characters
- multibyte 6
- supported in object names 7

CLI interface 5
cluster, installing on 3, 12, 13, 15
columns
- data types supported 6
- size and number of 6

command interfaces, required 5
compressed data 6
connections, to database 5
constraints, integrity 21
CREATE SUBDIRS command 12

D

DATA CAPTURE CHANGES 22
data compression 6
data types, supported 6
database
- multiple instances of 6
- preparing for GoldenGate 20
- user 5

datalinks 6
DB2READLOG API 5
DDL 6
deletes, cascaded 21
disk requirements 3
downloading GoldenGate 10

E

environment variables, setting 10

F

firewall, configuring 4
FTP access for GoldenGate 4

G

ggmessage.dat file 33
GGSMGR default Manager name 14
ggsmsg.dll 14
GLOBALS file 14
GoldenGate
- downloading 10
- installed programs and files 27
- installing 10
- uninstalling 25
- upgrading 10

I

installing GoldenGate 10

K

key
assigning 22
changes, preventing 22
name, supported characters 7
KEYCOLS option, TABLE or MAP 22

L

large objects, limitations on 6
LIBPATH variable 11
libraries, Visual C++ 4, 13
Linux, installing on 12
LOGGED clause, in LOB definition 6
LOGRETAIN parameter, DB2 20
logs, configuring 20
LONGVAR logging, excluding 23

M

Manager
as Windows service 14
multiple on same system 14
name, customizing 14
MANUALSTART Windows service option 15
MGRSERVNAME parameter 14
Microsoft Visual C ++ 2005 SP1 Redistributable Package 4, 13
Multi Dimensional Clustered Tables (MDC) 6
multi-byte characters 6

N

name
database, getting 20
Manager, specifying 13
non-supported characters in 8
supported characters in 7

O

operating system requirements 3
OVERFLOWLOGPATH parameter, DB2 21

P

PASSWORD Windows service option 15
platforms, supported 3
ports, required by GoldenGate 4
privileges
database 5
operating system 4
privileges, database 5

R

rows
before and after images of 22
size supported 6

S

SHLIB_PATH variable 11
spaces
in folder names 13
in object and column names 9
subdirectories, creating 12

T

tables
kinds supported 6
preparing for processing 21
transaction capture, enabling 22
TCP/IP, configuring 4
transaction logs, configuring 20
transactions, open 23
triggers, disabling on target 21

U

uninstalling GoldenGate 25
UNIX, installing on 12
upgrades, GoldenGate 10
user defined type 6
USER Windows service option 15
user, GoldenGate 5

USEREXIT parameter, DB2 20

V

VALUE COMPRESSION 6
VAMSERV program 28
virtual machine, support for 4
Visual C ++ 2005 SP1 Redistributable Package 4, 13

W

Windows, installing on 13