Oracle® Fusion Middleware

Installing and Configuring Oracle GoldenGate for DB2 z/OS

12*c* (12.2.0.1)

E67824-01

May 2016

Contains instructions for installing and performing initial setup of Oracle GoldenGate for the DB2 z/OS database.

ORACLE

Oracle Fusion Middleware Installing and Configuring Oracle GoldenGate for DB2 z/OS, 12*c* (12.2.0.1)

E67824-01

Contents

3 Preparing the System for Oracle GoldenGate

4 Preparing the DB2 for z/OS Transaction Logs for Oracle GoldenGate

5 Uninstalling Oracle GoldenGate

A Oracle GoldenGate Installed Components

Preface

On the IBM z/OS platform, Oracle GoldenGate for DB2 runs natively on UNIX System Services (USS), also known as Open MVS (OMVS). With Oracle GoldenGate, you can move data between similar or dissimilar supported DB2 for z/OS versions, or you can move data between a DB2 for z/OS database and a database of another type, such as Oracle or DB2 for LUW. Oracle GoldenGate for DB2 on the z/OS platform supports the filtering, mapping, and transformation of data, unless otherwise noted in this documentation.

This guide helps you get started with installing Oracle GoldenGate on a DB2 z/OS database system and performing initial setup. Refer to the other Oracle GoldenGate documentation listed in this Preface for additional information to configure, run, and manage your Oracle GoldenGate environment.

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

The Oracle GoldenGate documentation set includes the following components:

Windows, UNIX, and Linux Platforms

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

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

Conventions

The following text conventions are used in this document:

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

1

System Requirements and Preinstallation Instructions

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

This chapter contains the following topics:

1.1 Verifying Certification and System Requirements

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

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

1.2 Operating System Requirements

This section outlines the requirements to support Oracle GoldenGate on the local operating system.

1.2.1 System Services

The following system services must be enabled on the host system.

- Activate Unix System Services (USS) in full function mode rather than in minimum mode. You can use the z/OS UNIX Configuration Wizard for this purpose. Refer to the IBM UNIX System Services Planning manual for more information. The UNIX customization includes the following:

- Make the Language Environment run-time library (RTL) available to Oracle GoldenGate and other C programs by including it in the link list or Link Pack Area (LPA), or by adding it to the `STEPLIB` environment variable. RTL consists of data sets `SCEERUN` and `SCEERUN2` . If you are using `STEPLIB` , define the `SCEERUN` data sets to LLA to make loading the run-time modules faster. See the UNIX System Services Planning documentation for more information.
- To support the Oracle GoldenGate edit and report commands under the TSO/E *OMVS* command, concatenate the following target libraries to the appropriate ISPF data definition names (`ddnames`):
  * Data set `SYS1.SBPXPENU` to `ddname ISPPLIB`
  * Data set `SYS1.SBPXMENU` to `ddname ISPMLIB`
  * Data set `SYS1.SBPXTENU` to `ddname ISPTLIB`
  * Data set `SYS1.SBPXEXEC` to `ddname SYSEXEC` or `SYSPROC`

- Install Recoverable Resource Manager Services (RRS) for the best performance. Depending on the attachment type, the user by which Oracle GoldenGate runs might need one of the following permissions on the DSNR resource class:
  - If using RRSAF, assign `RACF ACCESS(READ)` to the RRSAF resource. IBM recommends using RRSAF because it has several advantages over CAF, including support for two-phase commit, thread reuse, and control over accounting intervals.
  - If using CAF, assign `RACF ACCESS(READ)` to the `BATCH` environment. If using CAF, it is possible for Oracle GoldenGate to hold locks on the system catalog until it receives a transaction commit.
- Oracle GoldenGate supports Sysplex data sharing.

1.2.2 Memory Requirements

Oracle GoldenGate requires the following memory resources on the local system.

- Allocate approximately 50-100 MB of virtual memory for each Oracle GoldenGate Extract process, depending upon the amount of uncommitted data per unit of work. Oracle GoldenGate stores transaction data on the source system until either a commit or rollback indicator is received. Alternately, you can use spill files to hold most of the data. You may need to monitor the usage of your z/OS page data sets that have extensive USS usage. The default size of the initial Extract memory heap might be too high for some z/OS systems, especially those running applications that primarily generate small transactions. To adjust the size, use the `TRANSMEMORY` parameter in the Extract parameter file.
- The GGSCI program uses the International Components for Unicode (ICU) libraries and thus requires a virtual memory allocation of 48 MB.
- When using TSO to access USS, give the TSO user enough memory allocation to start the Oracle GoldenGate GGSCI process. If you get the error message "`CEE3536S Not enough storage was available for the WSA`," increase the TSO user memory or ask your TSO administrator to do it.

1.2.3 Disk Requirements

This section outlines the disk requirements needed to support Oracle GoldenGate.

- Assign a zFS (zSeries file systems) or Hierarchical File System (HFS) volume. To determine the size of the Oracle GoldenGate download file, view the Size column

before downloading your selected build from Oracle Software Delivery Cloud. The value shown is the size of the files in compressed form. The size of the expanded Oracle GoldenGate installation directory will be significantly larger on disk. For more information, see Section 2.3, "Understanding and Obtaining the Oracle GoldenGate Distribution."

- To install Oracle GoldenGate into a cluster environment, install the Oracle GoldenGate binaries and files on a shared file system that is available to all cluster nodes.
  - `DSN=[hfs dsn]`
  - `DISP=(NEW,CATLG,DELETE)`
  - `SPACE =`
  - `DCB=DSORG=PO,DSNTYPE=HFS`

    Automatically mount the new file at IPL with an update to the `BPXPRM UNIX PARMLIB` member. Make certain that the `MOUNT FILESYSTEM` parameter for HFS in the `BPXPRM PARMLIB` member is not set to `NOSETUID` . To mount the files:

    ```
    MOUNT FILESYSTEM([hfs dsn]') TYPE(HFS) MODE(RDWR)
    MOUNTPOINT('[/u/ggs]')
    ```

- An additional 1 GB of disk space is required on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. You may need more or less than this amount, because the space that is consumed by the trails depends on the volume of data that will be processed. See the guidelines for sizing trails in *Administering Oracle GoldenGate for Windows and UNIX*.

1.2.4 Network Requirements

Oracle GoldenGate requires the following network resources.

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

1.2.5 Other Operating System Requirements

The following additional features of the operating system must be available to support Oracle GoldenGate.

- To use Oracle GoldenGate user exits, install the C/C++ Compiler, which creates the programs in the required shared object or DLL.
- For best results, have the following tools on the system:

  - Gzip to decompress the Oracle GoldenGate installation files. Otherwise, you must unzip the installation on a PC by using a Windows-based product, and then FTP it to the z/OS machine.
  - RACF command processor or comparable security package. (This documentation shows recommendations and examples for RACF.)
  - Time Sharing Option Extensions (TSO/E) command processor, including authorized TSO/E commands.
- For best results, apply HIPER maintenance on a regular basis, staying within one year of the current maintenance release, as recommended by IBM. The HIPER process identifies defects that could affect data availability or integrity. IBM provides Program Temporary Fixes (PTF) to correct defects found in DB2 and z/OS.

1.2.6 Operating System Privileges

Table 1–1 shows the required operating system privileges for Oracle GoldenGate:

Table 1–1 Operating System Privileges

User privilege	Manager	Extract	Replicat
Resource Access Control Facility (RACF) account (or equivalent) with OMVS segment[1]	X		
`CONNECT` to the local DB2 subsystem[2]		X	X
`ACCESS(READ )` to the bootstrap data set (BSDS)		X[3]	
`ACCESS(READ)` to resource `BPX.FILEATTR.APF` in `CLASS(FACILITY)`		X	
`ACCESS(READ)` to resource `BPX.JOBNAME` in `CLASS(FACILITY)`[4]		X	X
`chmod +rw` on the subdirectories in the Oracle GoldenGate directory	X	X	X

[1] The user who starts the Manager process is typically the user by which other Oracle GoldenGate processes run.

[2] Requires access to either the `CAF` or the `RRSAF` protected access profile in the `DSNR RACF` resource class, depending upon the `MVSATTACHTYPE` value in the ODBC initialization file.

[3] Non-data sharing only.

[4] In IBM DB2 10 for z/OS 1.2.6.

1.3 Database Requirements

This section contains Oracle GoldenGate requirements that are specific to the DB2 database on z/OS.

1.3.1 Database Configuration

Configure the following database components to support Oracle GoldenGate.

- Install a DB2 ODBC driver. The Oracle GoldenGate Extract and Replicat processes use ODBC (Open Database Connectivity) to connect to the DB2 subsystem. For information about ODBC, see the *DB2 for z/OS ODBC Guide and Reference* documentation.
- Install and configure the DB2 ODBC dynamic load library.

- Grant Oracle GoldenGate EXECUTE privilege on the plan that is specified in the ODBC initialization file (the default is DSNACLI).
- You might need to insert the name of the local DB2 subsystem into the SYSIBM.LOCATIONS table, which contains the remote DB2 server locations. Use a statement similar to the following (the example uses the name *DB2A*).

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, PORT) VALUES ('DB2A', '446');
```

1.3.2 Database User for Oracle GoldenGate Processes

Oracle GoldenGate requires a database user account. Create this account and assign privileges according to the following guidelines.

- By default, the user who starts the Manager process becomes the default DB2 primary authorization ID for all of the Oracle GoldenGate processes that any users start in that Oracle GoldenGate instance.You can assign a different user to any process by means of JCL or UNIX variables.
- To monitor Oracle GoldenGate processing accurately, do not permit other applications or processes to operate as the Oracle GoldenGate user.
- Assign the DB2 privileges listed in Table 1–2 to the user by which Extract and Replicat will be running (default is the user who starts Manager). These are in addition to any permissions that DB2 ODBC requires. All Extract privileges apply to initial-load and log-based Extract processes, except where noted.

Table 1–2 Privileges Needed by Oracle GoldenGate for DB2 on z/OS

User privilege	Extract	Replicat
MONITOR2 (does not apply to initial-load Extract)	X	
SELECT ON the following SYSIBM tables: SYSTABLES SYSCOLUMNS SYSTABLEPART SYSKEYS SYSINDEXES SYSCOLAUTH SYSDATABASE SYSFOREIGNKEYS SYSPARMS SYSRELS SYSROUTINES SYSSYNONYMS SYSTABAUTH SYSAUXRELS	X	X
SELECT on source tables[1]	X	
INSERT, UPDATE, DELETE on target tables		X
CREATE TABLE[2]		X
EXECUTE on ODBC plan (default is DSNACLI)	X	

Table 1–2 (Cont.) Privileges Needed by Oracle GoldenGate for DB2 on z/OS

User privilege	Extract	Replicat
Privileges required by SQLEXEC procedures or queries that you will be using.[3]	X	X

[1] SELECT on source tables required only if tables contain LOB columns, or for an initial-load Extract, if used.
[2] Required if using ADD CHECKPOINTTABLE in GGSCI to use the database checkpoint feature.
[3] SQLEXEC enables stored procedures and queries to be executed by an Oracle GoldenGate process.

1.4 Supported DB2 on z/OS Data Types

This section lists the DB2 data types that Oracle GoldenGate supports on the z/OS platform and any limitations of this support.

- Oracle GoldenGatee does not perform character set conversion for columns that could contain multi-byte data. This includes GRAPHIC, VARGRAPHIC and DBCLOB data types, as well as CHAR, VARCHAR, and CLOB for tables defined with ENCODING_SCHEME of 'M' (multiple CCSID set or multiple encoding schemes) or 'U' (Unicode). Such data is only supported if the source and target systems are the same CCSID.
- Oracle GoldenGate supports ASCII, EBCDIC, and Unicode data format. Oracle GoldenGate will convert between ASCII and EBCDIC data automatically. Unicode is never converted.
- Oracle GoldenGate supports most DB2 data types except those listed in Section 1.5, "Non-Supported DB2 on z/OS Data Types".

Limitations of Support

- The support of range and precision for floating-point numbers depends on the host machine. In general, the precision is accurate to 16 significant digits, but you should review the database documentation to determine the expected approximations. Oracle GoldenGate rounds or truncates values that exceed the supported precision.
- Oracle GoldenGate does not support the filtering, column mapping, or manipulation of large objects greater than 4K in size. Full Oracle GoldenGate functionality can be used for objects that are 4K or smaller.
- DECFLOAT is fully supported by Replicat. Extract does not capture DECFLOAT from the transaction log, but supports capture of it by fetching from the database.

1.5 Non-Supported DB2 on z/OS Data Types

This section lists DB2 data types that Oracle GoldenGate does not support on the z/OS platform. Data that is not supported may affect the integrity of the target data in relation to the source data.

- XML
- User-defined types
- Negative dates
- TIMESTAMP WITH TIMEZONE

1.6 Supported Objects and Operations for DB2 on z/OS

This section lists the database objects and types of operations that Oracle GoldenGate supports.

- Extraction and replication of DML operations on DB2 for z/OS tables that contain rows of up to 512KB in length. This size exceeds the maximum row size of DB2.
- `INSERT` operations from the IBM `LOAD` utility are supported for change capture if the utility is run with `LOG YES` and `SHRLEVEL CHANGE`, and the source tables that are being loaded have `DATA CAPTURE CHANGES` enabled (required by Oracle GoldenGate) and are specified in the Oracle GoldenGate Extract configuration.
- Oracle GoldenGate also supports initial loads with the `LOAD` utility to instantiate target tables during initial synchronization. For more information, see *Administering Oracle GoldenGate for Windows and UNIX*.
- Oracle GoldenGate supports the maximum number of columns per table that is supported by the database.
- Oracle GoldenGate supports the maximum column size that is supported by the database.
- Extraction and replication of data that is stored using DB2 data compression (`CREATE TABLESPACE COMPRESS YES`).
- `TRUNCATE TABLE` is supported, but because this command issues row deletes to perform the truncate, they are shown in Oracle GoldenGate statistics as such, and not as a truncate operation. To replicate a `TRUNCATE` , Replicat uses a `DELETE` without a `WHERE` clause.
- `TRUNCATES` are always captured from a DB2 for z/OS source, but can be ignored by Replicat if the `IGNORETRUNCATES` parameter is used in the Replicat parameter file.

1.7 Non-Supported Objects and Operations for DB2 on z/OS

- Extraction or replication of DDL operations
- Clone tables
- Data manipulation, including compression, that is performed within user-supplied DB2 exit routines, such as:
  - Date and time routines
  - Edit routines (`CREATE TABLE EDITPROC`)
  - Validation routines (`CREATE TABLE VALIDPROC`)
- AES encryption in Oracle GoldenGate security features
- The Multi-Journal feature does not support multi-journal sync of a transaction across multiple journals.

2

Installing Oracle GoldenGate

This chapter shows you how to install Oracle GoldenGate for DB2 z/OS databases. The installation procedure described here installs all of the components that are required to run and manage the processing (excluding any components required from other vendors, such as drivers or libraries). It also installs the Oracle GoldenGate utilities.

This chapter contains the following sections:

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

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

`http://support.oracle.com`

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

2.1 Installation Overview

To install Oracle GoldenGate, the following steps are required:

2.2 Choosing an Installation System for DB2

To install Oracle GoldenGate, you have the following options:

- To capture DB2 data, Oracle GoldenGate is installed directly on a z/OS source system. On the z/OS platform, Oracle GoldenGate runs natively on UNIX System Services (USS), also known as Open MVS (OMVS). Extract uses ODBC (Open Database Connectivity) to connect to the DB2 subsystem.
- To apply data to a DB2 target, you can use either of the following configurations:
  - Install Oracle GoldenGate directly on the z/OS target system. Replicat will use ODBC (Open Database Connectivity) to connect to the DB2 subsystem. To use this configuration, go to Section 2.4, "Installing Oracle GoldenGate in USS on z/OS".
  - Install Oracle GoldenGate DB2 Remote on a remote Linux, UNIX, or Windows system for remote delivery to the DB2 target system. In this configuration, Replicat connects to the target DB2 database by using the ODBC API that is supplied in DB2 Connect . This configuration requires DB2 LUW to be installed on the remote system. Use the following instructions.

Note: All of the Oracle GoldenGate functionality that is supported for DB2 for z/OS is supported by DB2Connect. In addition, ASCII character data is converted to EBCDIC automatically by DB2 Connect.

To Use Remote Delivery to the z/OS System via DB2Connect

1. For the intermediary system, select any Linux, UNIX, or Windows platform that Oracle GoldenGate supports for the DB2 for LUW database. This will be the system on which Oracle GoldenGate is installed.
2. Install and run DB2 for LUW on the selected remote system so that the Replicat process can use the supplied DB2 Connect driver.
3. Catalog the DB2 target node in the DB2 for LUW database on the remote system by using the following DB2 command:

```
catalog tcpip node db2_node_name remote DNS_name
server DB2_port-number
```

4. Add the target DB2 database to the DB2 for LUW catalog on the intermediary system by using the following DB2 command:

```
catalog db database_name as database_alias at node db_node_name
```

Note: Refer to the IBM DB2 LUW documentation for more information about these commands

5. Follow the instructions in Section 2.3, "Understanding and Obtaining the Oracle GoldenGate Distribution" to download the appropriate build for the selected remote Windows, Linux, or UNIX platform. Select the build that is labeled for *DB2Remote*. Then, follow the appropriate instructions to install Oracle GoldenGate on the intermediary system:
   - Section 2.5, "Installing Oracle GoldenGate on Linux and UNIX"
   - Section 2.6, "Installing Oracle GoldenGate on Windows"
6. Specify the DB2 target database name with the Replicat parameter `TARGETDB` when you configure the Oracle GoldenGate processes.

2.3 Understanding and Obtaining the Oracle GoldenGate Distribution

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

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

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

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

To obtain Oracle GoldenGate follow these steps:

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

2.4 Installing Oracle GoldenGate in USS on z/OS

Follow these instructions to install Oracle GoldenGate in UNIX System Services on a z/OS system.

1. Log on as a superuser or as a user who has z/OS authorization to add the "a" extended attribute. If you cannot log on as this kind of user, then someone with the proper authority will need to set the "a" attribute later. It is required for the Extract process, which uses a privileged API by the name of `IFI IFCID 306` to perform certain work, and then it reverts to normal application privileges for other work. The work that requires elevated privileges is:
   - Call the API in Supervisor State with Storage Key 0.
   - Allocate a return area in Extended Common Storage Area (ECSA) memory subpool 241 (not fetch protected, pageable) with Storage Key 7. The size of the return area will be a minimum of the largest DB2 log record that is returned, plus the additional area that is defined in DB2 macro `DSNDQW04`.
2. Extract the Oracle GoldenGate installation file to the system and directory where you want Oracle GoldenGate to be installed.
3. In the UNIX shell, verify that the DLL files have the write mode set.

   ```
   ls -l *.dll

   -rwxrwx---  a-s-  1 BSTPIER  IPGROUP  6127616 Apr 25 14:56 lib.dll
   ```

4. If the write mode is not set, issue the following command.

   ```
   chmod +w *.dll
   ```

5. In the UNIX shell, check for the "a" authorization for Extract and the DLL files by issuing the following command.

   ```
   ls -E extract *.dll
   ```

   In the output, look for the "a" attribute, indicating the authorization in Authorized Program Facility (APF) as shown in the following example.

```
-rwxrwx---  a-s-  1 BSTPIER  IPGROUP  6127616 Apr 25 14:56 extract
```

6. If the "a" attribute is not set for both Extract and its DLL files (it may be built into one but not the other, depending on the build configuration), issue the following commands to add the authorization for Extract and the DLL files.

```
extattr +a extract
extattr +a *.dll
```

7. Run the command shell and change directories to the new Oracle GoldenGate directory.
8. Set the `LIBPATH` environment variable to include the path to the Oracle GoldenGate installation directory.

```
export LIBPATH=".:$(LIBPATH)"
```

   The "." indicates the current directory. This change must be performed before starting any Oracle GoldenGate processes and can be made in one of the following ways:

   - system-wide
   - for a specific user at login time
   - manually each time that Oracle GoldenGate is executed

9. From the Oracle GoldenGate directory, run the GGSCI program.

```
GGSCI
```

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

```
CREATE SUBDIRS
```

11. Issue the following command to exit GGSCI.

```
EXIT
```

2.5 Installing Oracle GoldenGate on Linux and UNIX

Follow these steps to install Oracle GoldenGate for Oracle on a Linux or UNIX system.

1. Extract the Oracle GoldenGate installation file to the system and directory where you want Oracle GoldenGate to be installed.
2. Run the command shell.
3. Change directories to the new Oracle GoldenGate directory.
4. From the Oracle GoldenGate directory, run the GGSCI program.

```
GGSCI
```

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

```
CREATE SUBDIRS
```

6. Issue the following command to exit GGSCI.

```
EXIT
```

2.6 Installing Oracle GoldenGate on Windows

Follow these instructions to:

- Install Oracle GoldenGate for Oracle on a Windows system.
- Install Oracle GoldenGate DB2Remote on a Windows system to support remote delivery to a DB2 for z/OS target system. To install directly on a z/OS system, see Section 2.4, "Installing Oracle GoldenGate in USS on z/OS."

2.6.1 Installing Oracle GoldenGate into a Windows Cluster

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

2.6.2 Installing the Oracle GoldenGate Files

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

   `C:\"Oracle GoldenGate"` is not valid.

   `C:\Oracle_GoldenGate` is valid.
3. From the Oracle GoldenGate folder, run the GGSCI program.
4. In GGSCI, issue the following command to create the Oracle GoldenGate working directories.

   ```
   CREATE SUBDIRS
   ```

5. Issue the following command to exit GGSCI.

   ```
   EXIT
   ```

2.6.3 Specifying a Custom Manager Name

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

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

To Specify a Custom Manager Name:

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

   ```
   EDIT PARAMS ./GLOBALS
   ```

> **Note:** The ./ portion of this command must be used, because the GLOBALS file must reside at the root of the Oracle GoldenGate installation file.

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

   ```
   MGRSERVNAME name
   ```

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

2.6.4 Installing Manager as a Windows Service

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

Installing Manager as a service is required on a Windows Cluster, but optional otherwise.

1. (Recommended) Log on as the system administrator.
2. Click **Start** then **Run** and then enter 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 utility with the following syntax:

   ```
   install option [...]
   ```

   Where *option* is one of the following:

Table 2–1 INSTALL Options

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

Table 2–1 (Cont.) INSTALL Options

Option	Description
USER *name*	Specifies a domain user account that executes Manager. For *name*, include the domain name, a backward slash, and the user name, for example HEADQT\GGSMGR. By default, the Manager service is installed to use the Local System account.
PASSWORD *password*	Specifies the password for the user that is specified with USER.

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

4. If Windows User Account Control (UAC) is enabled, you are prompted to allow or deny the program access to the computer. Select **Allow** to enable the INSTALL utility to run.

   The INSTALL utility installs the Manager service with a local system account running with administrator privileges. No further UAC prompts will be encountered when running Manager if installed as a service.

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

3

Preparing the System for Oracle GoldenGate

This chapter contains guidelines for preparing the database and the system to support Oracle GoldenGate.

This chapter contains the following sections:

- Preparing Tables for Processing
- Configuring Database Connections
- Accessing Load Modules
- Specifying Job Names and Owners
- Assigning WLM Velocity Goals
- Monitoring Processes
- Supporting Globalization Functions

3.1 Preparing Tables for Processing

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

Disabling Triggers and Cascade Constraints

Assigning Row Identifiers

Handling `ROWID` Columns

3.1.1 Disabling Triggers and Cascade Constraints

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

- A delete is issued for `emp_src`.
- It cascades a delete to `salary_src`.
- Oracle GoldenGate sends both deletes to the target.
- The parent delete arrives first and is applied to `emp_targ`.
- The parent delete cascades a delete to `salary_targ`.

- The cascaded delete from salary_src is applied to salary_targ.
- The row cannot be located because it was already deleted in step 5.

3.1.2 Assigning Row Identifiers

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

3.1.2.1 How Oracle GoldenGate Determines the Kind of Row Identifier to Use

Unless a KEYCOLS clause is used in the TABLE or MAP statement, Oracle GoldenGate selects a row identifier to use in the following order of priority:

1. Primary key
2. First unique key alphanumerically that does not contain a timestamp or non-materialized computed column.
3. If none of the preceding key types exist (even though there might be other types of keys defined on the table) Oracle GoldenGate constructs a pseudo key of all columns that the database allows to be used in a unique key, excluding those that are not supported by Oracle GoldenGate in a key or those that are excluded from the Oracle GoldenGate configuration.

Note: If there are other, non-usable keys on a table or if there are no keys at all on the table, Oracle GoldenGate logs an appropriate message to the report file. Constructing a key from all of the columns impedes the performance of Oracle GoldenGate on the source system. On the target, this key causes Replicat to use a larger, less efficient WHERE clause.

3.1.2.2 Using KEYCOLS to Specify a Custom Key

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

3.1.3 Handling ROWID Columns

Any attempt to insert into a target table that includes a column with a data type of ROWID GENERATED ALWAYS (the default) will fail with the following ODBC error:

```
ODBC error: SQLSTATE 428C9 native database error -798. {DB2 FOR OS/390}{ODBC
DRIVER}{DSN08015} DSNT408I SQLCODE = -798, ERROR: YOU CANNOT INSERT A VALUE INTO A
COLUMN THAT IS DEFINED WITH THE OPTION GENERATED ALWAYS. COLUMN NAME ROWIDCOL.
```

You can do one of the following to prepare tables with ROWID columns to be processed by Oracle GoldenGate:

- Ensure that any ROWID columns in target tables are defined as GENERATED BY DEFAULT.
- If it is not possible to change the table definition, you can work around it with the following procedure.

To Work Around `ROWID GENERATE ALWAYS`:

1. For the source table, create an Extract `TABLE` statement, and use a `COLSEXCEPT` clause in that statement that excludes the `ROWID` column. For example:

   ```
   TABLE tab1, COLSEXCEPT (rowidcol);
   ```

   The `COLSEXCEPT` clause excludes the `ROWID` column from being captured and replicated to the target table.

2. For the target table, ensure that Replicat does not attempt to use the `ROWID` column as the key. This can be done in one of the following ways:

   - Specify a primary key in the target table definition.
   - If a key cannot be created, create a Replicat `MAP` parameter for the table, and use a `KEYCOLS` clause in that statement that contains any unique columns except for the `ROWID` column. Replicat will use those columns as a key. For example:

     ```
     MAP tab1, TARGET tab1, KEYCOLS (num, ckey);
     ```

For more information about `KEYCOLS`, see Section 3.1.2, "Assigning Row Identifiers".

3.2 Configuring Database Connections

The following guidelines assume that an appropriate ODBC driver is installed and configured.

3.2.1 Setting Initialization Parameters

The following DB2 for z/OS initialization parameters apply to Oracle GoldenGate and must be set correctly before starting Oracle GoldenGate processes.

- `MVSDEFAULTSSID`: set to the DB2 subsystem.
- `LOCATION`: set to the DB2 location name as stored in the DB2 Boot Strap Dataset.
- `MVSATTACHTYPE`: set to `RRSAF` (Recoverable Resource Manager Services Attachment Facility) or `CAF` (Call Attachment Facility). IBM recommends using `RRSAF`.
- `MULTICONTEXT`: set to 1 if using `RRSAF`.
- `PLANNAME`: set to the DB2 plan. The default plan name is `DSNACLI`.

Do not use the `CURRENTAPPENSCH` initialization parameter (keyword).

Note: When using the `CAF` attachment type, you must use the Oracle GoldenGate `DBOPTIONS` parameter with the `NOCATALOGCONNECT` option in the parameter file of any Extract or Replicat process that connects to DB2. This parameter disables the usual attempt by Oracle GoldenGate to obtain a second thread for the DB2 catalog. Otherwise, you will receive error messages, such as: `ODBC operation failed: Couldn't connect to` *`data source`* `for catalog queries.`

3.2.2 Specifying the Path to the Initialization File

Specify the ODBC initialization file by setting the DSNAOINI environment variable in the z/OS UNIX profile, as in the following example:

```
export DSNAOINI="/etc/odbc810.ini"
```

3.2.3 Ensuring ODBC Connection Compatibility

To ensure that you configure the DB2 ODBC initialization file correctly, follow the guidelines in the *DB2 UDB for z/OS ODBC Guide and Reference* manual. One important consideration is the coding of the open and close square brackets (the [character and the] character). The square bracket characters are "variant" characters that are encoded differently in different coded character set identifiers (CCSID), but must be of the IBM-1047 CCSID in the ODBC initialization file. DB2 ODBC does not recognize brackets of any other CCSID. Note the following:

- The first (or open) bracket must use the hexadecimal characters X'AD' (0xAD).
- The second (or close) bracket must use the hexadecimal characters X'BD' (0xBD).

To set the correct code for square brackets, use any of the following methods.

- Use the hex command in OEDIT and change the hex code for each character appropriately.
- Use the iconv utility to convert the ODBC initialization file. For example, to convert from CCSID IBM-037 to IBM-1047, use the following command:

  ```
  iconv -f IBM-037 -t IBM-1047 ODBC.ini > ODBC-1047.ini
  mv ODBC-1047.ini ODBC.ini
  ```

- Change your terminal emulator or terminal configuration to use CCSID IBM-1047 when you create or alter the file.

3.2.4 Specifying the Number of Connection Threads

Every Oracle GoldenGate process makes a database connection. Depending on the number of processes that you will be using and the number of other DB2 connections that you expect, you might need to adjust the following DB2 system parameters on the DSNTIPE DB2 Thread Management Panel:

- MAX USERS (macro DSN6SYSP CTHREAD)
- MAX TSO CONNECT (macro DSN6SYSP IDFORE)
- MAX BATCH CONNECT (macro DSN6SYSP IDBACK)

If using RRSAF, allow:

- Two DB2 threads per process for each of the following:
  - Extract
  - Replicat
  - The GGSCI command DBLOGIN (logs into the database)
  - DEFGEN utility (generates data definitions for column mapping)
- One extra DB2 thread for Extract for IFI calls.
- One extra DB2 thread for each SQLEXEC parameter statement that will be issued by each Extract and Replicat process. For more information about SQLEXEC, see the *Reference for Oracle GoldenGate for Windows and UNIX*.

If using CAF, there can be only one thread per Oracle GoldenGate process.

3.3 Accessing Load Modules

Grant Oracle GoldenGate USS access to the SDSNLOAD system load library and to the DSNHDECP load module. You can include the libraries in one of the following places:

- The z/OS system search order.
- The USS profile of the Oracle GoldenGate user. Use a UNIX command similar to the following, where DSN810 is the user-assigned data set prefix from the DB2 installation.

```
export STEPLIB='DSN810.SDSNEXIT:DSN810.SDSNLOAD'
```

The preceding command will cause USS to allocate the equivalent of a STEPLIB DD statement whenever it executes a shell command or Oracle GoldenGate process. If using APF, all libraries in the STEPLIB concatenation must be APF-authorized.

3.4 Specifying Job Names and Owners

By default, USS sets the job name and owner of all Oracle GoldenGate processes to that of the user who started them. You can change the job name or user by setting the _BPX_JOBNAME and _BPX_USERID environment variables, or you can create z/OS jobs or started-task procedures for the Oracle GoldenGate processes. To use the environment variable _BPX_JOBNAME, at a minimum you should have read access to the RACF FACILITY class and BPX.JOBNAME name. For more details, see Table 1–1, " Operating System Privileges" and the IBM *z/OS System Services Planning* document.

3.5 Assigning WLM Velocity Goals

The user who starts the Manager process is typically the user by which other Oracle GoldenGate processes run. Oracle GoldenGate work appears as forked child processes of WLM subsystem type OMVS. Assign the Oracle GoldenGate processes their Workload Manager (WLM) velocity goals based on the following guidelines.

- Assign the Extract process that reads the transaction logs a medium velocity goal, one that is below the velocity of the main DB2 address spaces, but above the velocity of most online transactions, TSO/E sessions, and z/OS batch work. The higher the velocity goal, the more processor power that Extract will receive, and the less lag that it will experience.
- You can assign an initial-load Extract process a velocity goal, or you can treat it as a typical DB2 batch job. For more information about the initial-load processes, see *Administering Oracle GoldenGate for Windows and UNIX*.
- You might need to assign the Replicat process a higher velocity goal. Although Replicat is a typical DB2 batch application, it might require more processing power to prevent backlogs and latency.
- You probably will run Oracle GoldenGate utilities, such as DEFGEN and LOGDUMP, only occasionally, so you can let them perform like the other UNIX terminal-oriented work.
- If executing stored procedures with the SQLEXEC command, make certain that they do not become a bottleneck for Oracle GoldenGate. Their priority should be close to that of the calling Extract or Replicat process. WLM executes them with that

priority, but the z/OS system executes them under the priority of a stored procedure as defined by the DB2 and z/OS system programmers.

- If you run Oracle GoldenGate under the TSO/E `OMVS` command, the Oracle GoldenGate processes are subject to the system and WLM limits of the TSO/E user account, rather than those of the UNIX kernel. Very long TSO/E response times (up to 20 seconds), often with little service consumption, can be recorded for an OMVS user because of the way that OMVS polls for terminal input. This can affect those WLM goals that are based on response time.

You can use multiple WLM service classes for the Oracle GoldenGate processes. The following is an example of how to maintain relative priorities for Oracle GoldenGate and other work, from highest priority to the lowest:

1. z/OS system processes, including the UNIX kernel and IRLM.
2. DB2 for z/OS address spaces for the primary Extract group.
3. Primary Extract group configured for online or batch change synchronization, and any DB2 stored procedures that it calls.
4. z/OS transaction managers, such as CICS and IMS.
5. Collector (Server) for local Extract data pump, if used.
6. Local Extract data pump (reading from trail), if used.
7. Collector for remote trails (files received from a remote site). Such files include the QSAM file created with the Extract `RMTBATCH` parameter on a NonStop system.
8. Online Replicat groups and any DB2 stored procedures that they call.
9. Manager process (required only for startup of Oracle GoldenGate processes and trail cleanup).
10. GGSCI and other user UNIX and TSO/E terminal work.
11. Initial-load Extract and any DB2 stored procedures that it calls.
12. Initial-load Replicat and any DB2 stored procedures that it calls.
13. Other z/OS batch work.

3.6 Monitoring Processes

The following topics provide information about monitoring Oracle GoldenGate with z/OS system facilities.

Section 3.6.1, "Viewing Oracle GoldenGate Messages"

Section 3.6.2, "Identifying Oracle GoldenGate Processes"

Section 3.6.3, "Interpreting Statistics for Update Operations"

Section 3.6.3, "Interpreting Statistics for Update Operations"

3.6.1 Viewing Oracle GoldenGate Messages

If the system log process (`syslog` daemon `syslogd`) is running, USS routes Oracle GoldenGate messages to their configured destination by means of UNIX message priority. For more information about configuring `syslogd`, see the z/OS IP configuration documents and the *UNIX System Services Planning* document.

If `syslogd` is not running, Oracle GoldenGate writes its command output, status information, and error messages to the system console. You can redirect console

messages to the Oracle GoldenGate USS session and to the Oracle GoldenGate report files by using the following UNIX command:

```
export _BPXK_JOBLOG=STDERR
```

3.6.2 Identifying Oracle GoldenGate Processes

The system management facility (SMF) typically creates a separate accounting record for each UNIX process, including Oracle GoldenGate processes. However, if a user invokes the UNIX shell by using the `OMVS` command with the default `SHAREAS` option, or if a user sets the environment variable `_BPX_SHAREAS` to `YES` , it could cause two or more processes to run in the same address space. SMF provides process identification only for the first process, but resource consumption is accumulated for all processes that are running. For Oracle GoldenGate, this means that the work probably will be recorded under the Manager process, which is named `mgr`.

If the DB2 accounting trace is also active to the SMF destination, DB2 will create an SMF accounting record for each of the following Oracle GoldenGate processes:

- Extract
- Replicat
- Manager, if performing maintenance on Oracle GoldenGate tables. Examples of Oracle GoldenGate tables are the marker table and the Replicat checkpoint table.
- GGSCI sessions that issue the Oracle GoldenGate `DBLOGIN` command to log into the database.

3.6.3 Interpreting Statistics for Update Operations

The actual number of DML operations that are executed on the DB2 database might not match the number of extracted DML operations that are reported by Oracle GoldenGate. DB2 does not log update statements if they do not physically change a row, so Oracle GoldenGate cannot detect them or include them in statistics.

3.7 Supporting Globalization Functions

Oracle GoldenGate provides globalization support. The following are things to take into consideration when using this support.

Replicating From a Source that Contains Both ASCII and EBCDIC

Specifying Multi-Byte Characters in Object Names

3.7.1 Replicating From a Source that Contains Both ASCII and EBCDIC

When replicating to or from a DB2 source system to a target that has a different character set, some consideration must be given to the encoding of the character data on the DB2 source if it contains a mix of ASCII and EBCDIC data. Character set conversion by any given Replicat requires source data to be in a single character set.

The source character set is specified in the trail header. Thus, the Oracle GoldenGate trail can contain either ASCII or EBCDIC data, but not both. Unicode tables are processed without any special configuration and are exempt from the one-character set requirement.

With respect to a source that contains both character encoding types, you have the following options:

- You can use one Extract for all of your tables, and have it write the character data to the trail as either ASCII or as EBCDIC.
- You can use different Extracts: one Extract to write the ASCII character data to a trail, and another Extract to write the EBCDIC character data to a different trail. You then associate each trail with its own data pump process and Replicat process, so that the two data streams are processed separately.

To output the correct character set in either of those scenarios, use the `TRAILCHARSETASCII` and `TRAILCHARSETEBCDIC` parameters. The default is `TRAILCHARSETEBCDIC`. Without these parameters, ASCII and EBCDIC data are written to the trail as-is. When using these parameters, note the following:

- If used on a single-byte DB2 subsystem, these parameters cause Extract to convert all of the character data to either the ASCII or EBCDIC single-byte CCSID of the subsystem to which Extract is connected, depending on which parameter is used (except for Unicode, which is processed as-is).
- If used on a multi-byte DB2 subsystem, these parameters cause Extract to capture only ASCII or EBCDIC tables (and Unicode). Character data is written in either the ASCII or EBCDIC mixed CCSID (depending on the parameter used) of the DB2 z/OS subsystem to which Extract is connected.

3.7.2 Specifying Multi-Byte Characters in Object Names

If the name of a schema, table, column, or stored procedure in a parameter file contains a multi-byte character, the name must be double-quoted. For more information about specifying object names, see *Administering Oracle GoldenGate for Windows and UNIX*.

4

Preparing the DB2 for z/OS Transaction Logs for Oracle GoldenGate

This chapter contains information that you need to configure the DB2 transaction logging to support data capture by Oracle GoldenGate Extract.

- Section 4.1, "Specifying a Bootstrap Data Set"
- Section 4.2, "Making Transaction Data Available"

4.1 Specifying a Bootstrap Data Set

The Extract process uses the BSDS to find its initial position in the transaction log within a non-data-sharing environment. You specify the BSDS when you create your Extract groups. It must be the BSDS for the DB2 subsystem to which the Extract process will be connecting. Oracle GoldenGate does not perform any validations of the BSDS specification.

4.2 Making Transaction Data Available

Oracle GoldenGate can extract DB2 transaction data from the active and archived logs. Follow these guidelines to configure the logs so that Extract can capture data.

Section 4.2.1, "Enabling Change Capture"

Section 4.2.2, "Enabling Access to Log Records"

Section 4.2.3, "Sizing and Retaining the Logs"

Section 4.2.4, "Using Archive Logs on Tape"

Section 4.2.5, "Controlling Log Flushes"

4.2.1 Enabling Change Capture

Follow these steps to configure DB2 to log data changes in the expanded format that is supplied by the `DATA CAPTURE CHANGES` feature of the `CREATE TABLE` and `ALTER TABLE` commands. This format provides Oracle GoldenGate with the entire before and after images of rows that are changed with update statements.

1. From the Oracle 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][, encryption_options]
```

3. Issue the following command. where *table* is the fully qualified name of the table. You can use a wildcard to specify multiple table names but not owner names.

   ```
   ADD TRANDATA table
   ```

   By default, ADD TRANDATA issues the following command:

   ```
   ALTER TABLE name DATA CAPTURE CHANGES;
   ```

4.2.2 Enabling Access to Log Records

Activate DB2 Monitor Trace Class 1 ("TRACE(MONITOR) CLASS(1) ") so that DB2 allows Extract to read the active log. The default destination of OPX is sufficient, because Oracle GoldenGate does not use a destination.

To Start the Trace Manually

1. Log on to DB2 as a DB2 user who has the TRACE privilege or at least SYSOPR authority.
2. Issue the following command:

   ```
   start trace(monitor) class(1) scope(group)
   ```

To Start the Trace Automatically When DB2 is Started

Do either of the following:

- Set MONITOR TRACE to "YES" on the DSNTIPN installation tracing panel.
- Set 'DSN6SYSP MON=YES ' in the DSNTIJUZ installation job, as described in the *DB2 UDB Installation Guide*.

> **Note:** The primary authorization ID, or one of the secondary authorization IDs, of the ODBC plan executor also must have the MONITOR2 privilege.

4.2.3 Sizing and Retaining the Logs

When tables are defined with DATA CAPTURE CHANGES, more data is logged than when they are defined with DATA CAPTURE NONE . If any of the following is true, you might need to increase the number and size of the active and archived logs.

- Your applications generate large amounts of DB2 data.
- Your applications have infrequent commits.
- You expect to stop Extract for long periods of time.
- Your network is unreliable or slow.

To control log retention, use the DSN6LOGP MAXARCH system parameter in the DSNTIJUZ installation job.

Retain enough log data so that Extract can start again from its checkpoints after you stop it or after an unplanned outage. 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 archived 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 Oracle GoldenGate environment over again.

> **Note:** The IBM documentation makes recommendations for improving the performance of log reads. In particular, you can use large log output buffers, large active logs, and make archives to disk.

4.2.4 Using Archive Logs on Tape

Oracle GoldenGate can read DB2 archive logs on tape, but it will degrade performance. For example, DB2 reserves taped archives for a single recovery task. Therefore, Extract would not be able to read an archive tape that is being used to recover a table until the recovery is finished. You could use DFHSM or an equivalent tools to move the archive logs in a seamless manner between online DASD storage and tape, but Extract will have to wait until the transfer is finished. Delays in Extract processing increase the latency between source and target data.

4.2.5 Controlling Log Flushes

When reading the transaction log, Extract does not process a transaction until it captures the commit record. If the commit record is on a data block that is not full, it cannot be captured until more log activity is generated to complete the block. The API that is used by Extract to read the logs only retrieves full physical data blocks.

A delay in receiving blocks that contain commits can cause latency between the source and target data. If the applications are not generating enough log records to fill a block, Extract generates its own log records by issuing `SAVEPOINT` and `COMMIT` statements, until the block fills up one way or the other and is released.

In a data sharing group, each API call causes DB2 to flush the data blocks of all active members, eliminating the need for Extract to perform flushes.

To prevent Extract from performing flushes, use the Extract parameter `TRANLOGOPTIONS` with the `NOFLUSH` option.

5 Uninstalling Oracle GoldenGate

This chapter contains platform-specific instructions for removing Oracle GoldenGate from a system.

It contains the following sections:

> **Warning:** These procedures assume that you no longer need the data in the Oracle GoldenGate trails, and that you no longer need to preserve the current Oracle GoldenGate environment. To preserve your current environment and data, make a backup of the Oracle GoldenGate directory and all subdirectories before starting this procedure.

5.1 Uninstalling Oracle GoldenGate from Linux or UNIX (includes USS)

Follow these instructions to remove Oracle GoldenGate from an IBM z/OS source or target system, or from a Linux or UNIX system that is being used for remote delivery to DB2 on z/OS.

On all Systems:

1. Run the command shell.
2. (Suggested) Log on as the system administrator or as a user with permission to issue Oracle GoldenGate commands and delete files and directories from the operating system.
3. Change directories to the Oracle GoldenGate installation directory.
4. Run GGSCI.
5. Stop all Oracle GoldenGate processes.
6. Stop the Manager process.

On a Source System:

Log into the database with the `DBLOGIN` command, and then run the `DELETE TRANDATA` command to alter the tables to `DATA CAPTURE NONE`.

On any System Where a Replicat Checkpoint Table Exists:

Log into the database with the `DBLOGIN` command, and then remove the Replicat checkpoint table by running the `DELETE CHECKPOINTTABLE` command.

On all systems:

1. Make certain all processes are stopped (including GGSCI).
2. Remove the Oracle GoldenGate files by removing the installation directory.

5.2 Removing Oracle GoldenGate from Windows Cluster

Follow these instructions to remove Oracle GoldenGate from a Windows cluster that is being used for remote delivery to DB2 on an IBM z/OS system.

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

   ```
   install deleteevents deleteservice
   ```

7. Move the cluster group to the next node in the cluster, and repeat from step 4.
8. Follow the instructions in Section 5.3, "Uninstalling Oracle GoldenGate from Windows (Non-Cluster)".

5.3 Uninstalling Oracle GoldenGate from Windows (Non-Cluster)

Follow these instructions to remove Oracle GoldenGate from a Windows system that is being used for remote delivery to DB2 on an IBM z/OS system.

On all Systems:

1. (Recommended) Log on as the system administrator or as a user with permission to issue Oracle GoldenGate commands and to delete files and directories from the operating system.
2. From the Oracle GoldenGate installation folder, run GGSCI.
3. Stop all Oracle GoldenGate processes.
4. Stop the Manager program or service.
5. Click **Start** then **Run** and type `cmd` in the Run dialog box to open the command console.
6. Change directories to the Oracle GoldenGate installation directory.

7. Run the INSTALL utility with the following syntax to stop Oracle GoldenGate events from being reported to the Windows Event Manager and to remove the Manager service.

```
install deleteevents deleteservice
```

On a Source System:

Log into the database with the DBLOGIN command, and then run the DELETE TRANDATA command to alter the tables to DATA CAPTURE NONE.

On any System Where a Replicat Checkpoint Table Exists:

Log into the database with the DBLOGIN command, and then remove the Replicat checkpoint table by running the DELETE CHECKPOINTTABLE command.

On all Systems:

Make certain all processes are stopped (including GGSCI) and then remove the Oracle GoldenGate files by removing the installation directory.

A

Oracle GoldenGate Installed Components

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

This appendix contains the following sections:

A.1 Oracle GoldenGate Programs and Utilities

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

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

Table A–1 Oracle GoldenGate Installed Programs and Utilities

Program	Description
`convchk`	Converts checkpoint files to a newer version.
`convprm`	Converts parameter files that do not use SQL-92 rules for quoted names and literals to updated parameter files that use SQL-92 rules. SQL-92 format for quoted object names and literals was introduced as the default with version 12c of Oracle GoldenGate.
`defgen`	Generates data definitions and is referenced by Oracle GoldenGate processes when source and target tables have dissimilar definitions.
`emsclnt`	Sends event messages created by Collector and Replicat on Windows or UNIX systems to EMS on NonStop systems.
`extract`	Performs capture from database tables or transaction logs or receives transaction data from a vendor access module.

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

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

A.2 Oracle GoldenGate Subdirectories

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

Note: Some directories may not exist in all installations.

Table A–2 Oracle GoldenGate Subdirectories

Directory	Description
br	Contains the checkpoint files for the bounded recover feature.
cfg	Contains the property and XML files that are used to configure Oracle GoldenGate Monitor.
dirdb	Contains the data store that is used to persist information that is gathered from an Oracle GoldenGate instance for use by the Oracle GoldenGate Monitor application or within Oracle Enterprise Manager.

Table A–2 (Cont.) Oracle GoldenGate 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 Oracle GoldenGate format. File name format is *group_name+sequence_number.ext* where *sequence_number* is a sequential number appended to aged files and *ext* is either cpe for Extract checkpoint files or cpr for Replicat checkpoint files. Do not edit these files. Examples: ext1.cpe rep1.cpr
dircrd	Contains credential store files.
dirdat	The default location for Oracle GoldenGate trail files and extract files that are created by Extract processes to store extracted data for further processing by the Replicat process or another application or utility. Written in internal Oracle GoldenGate format. File name format is a user-defined two-character prefix followed by either a six-digit sequence number (trail files) or the user-defined name of the associated Extract process group (extract files). Do not edit these files. Examples: rt000001 finance
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 Oracle GoldenGate technical support. Example: defs.dat
dirdmp	Contains trace, or dump, files that support the internal activity logging mechanism.
dirjar	Contains the Java executable files that support Oracle GoldenGate Monitor.

Table A–2 (Cont.) Oracle GoldenGate Subdirectories

Directory	Description
dirpcs	Default location for status files. File name format is *group.extension* where *group* is the name of the group and *extension* is either pce (Extract), pcr (Replicat), or pcm (Manager). These files are only created while a process is running. The file shows the program name, the process name, the port number, and the process ID. Do not edit these files. Examples: mgr.pcm ext.pce
dirprm	The default location for Oracle GoldenGate parameter files created by Oracle GoldenGate users to store run-time parameters for Oracle GoldenGate process groups or utilities. Written in external ASCII format. File name format is *group name/user-defined name.prm* or mgr.prm . These files may be edited to change Oracle GoldenGate parameter values after stopping the process. They can be edited directly from a text editor or by using the EDIT PARAMS command in GGSCI. Examples: defgen.prm finance.prm
dirrec	Not used by Oracle GoldenGate.
dirrpt	The default location for process report files created by Extract, Replicat, and Manager processes to report statistical information relating to a processing run. Written in external ASCII format. 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	Used by the triggen utility to store SQL scripts before triggen was deprecated. Currently used to store training scripts and any user-created SQL scripts that support Oracle GoldenGate.
dirtmp	The default location for storing transaction data when the size exceeds the memory size that is allocated for the cache manager. Do not edit these files.
dirwlt	Contains Oracle GoldenGate wallet files.
UserExitExamples	Contains sample files to help with the creation of user exits.

A.3 Other Oracle GoldenGate Files

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

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

Table A–3 Other Oracle GoldenGate Installed Files

Component	Description
bcpfmt.tpl	Template for use with Replicat when creating a run file for the Microsoft BCP/DTS bulk-load utility.
bcrypt.txt	Blowfish encryption software license agreement.
cagent.dll	Contains the Windows dynamic link library for the Oracle GoldenGate Monitor C sub-agent.
category.dll	Windows dynamic link library used by the INSTALL utility.
chkpt_*db*_create.sql	Script that creates a checkpoint table in the local database. A different script is installed for each database type.
db2cntl.tpl	Template for use with Replicat when creating a control file for the IBM LOADUTIL bulk-load utility.
ddl_cleartrace.sql	Script that removes the DDL trace file. (Oracle installations)
ddl_ddl2file.sql	Script that saves DDL from the marker table to a file.
ddl_disable.sql	Script that disables the Oracle GoldenGate DDL trigger. (Oracle installations)
ddl_enable.sql	Script that enables the Oracle GoldenGate DDL trigger. (Oracle installations)
ddl_filter.sql	Script that supports filtering of DDL by Oracle GoldenGate. This script runs programmatically; do not run it manually.
ddl_ nopurgeRecyclebin.sql	Empty script file for use by Oracle GoldenGate support staff.
ddl_ora11.sql ddl_ora12.sql	Scripts that run programmatically as part of Oracle GoldenGate DDL support; do not run these scripts.
ddl_pin.sql	Script that pins DDL tracing, the DDL package, and the DDL trigger for performance improvements. (Oracle installations)
ddl_purgeRecyclebin.sql	Script that purges the Oracle recyclebin in support of the DDL replication feature.
ddl_remove.sql	Script that removes the DDL extraction trigger and package. (Oracle installations)
ddl_session.sql ddl_session1.sql	Supports the installation of the Oracle DDL objects. This script runs programmatically; do not run it manually.
ddl_setup.sql	Script that installs the Oracle GoldenGate DDL extraction and replication objects. (Oracle installations)
ddl_status.sql	Script that verifies whether or not each object created by the Oracle GoldenGate DDL support feature exists and is functioning properly. (Oracle installations)
ddl_staymetadata_off.sql ddl_staymetadata_on.sql	Scripts that control whether the Oracle DDL trigger collects metadata. This script runs programmatically; do not run it manually.
ddl_trace_off.sql ddl_trace_on.sql	Scripts that control whether DDL tracing is on or off.

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

Component	Description
ddl_tracelevel.sql	Script that sets the level of tracing for the DDL support feature. (Oracle installations)
debug files	Debug text files that may be present if tracing was turned on.
demo_*db_scriptname*.sql demo_*more_db_scriptname*.sql	Scripts that create and populate demonstration tables for use with tutorials and basic testing.
.dmp files	Dump files created by Oracle GoldenGate processes for tracing purposes.
ENCKEYS	User-created file that stores encryption keys. Written in external ASCII format.
exitdemo.c	User exit example.
exitdemo_utf16.c	User exit example that demonstrates how to use UTF16 encoded data in the callback structures for information exchanged between the user exit and the process.
freeBSD.txt	License agreement for FreeBSD.
ggmessage.dat	Data file that contains error, informational, and warning messages that are returned by the Oracle GoldenGate processes. The version of this file is checked upon process startup and must be identical to that of the process in order for the process to operate.
ggserr.log	File that logs processing events, messages, errors, and warnings generated by Oracle GoldenGate.
ggsmsg.dll	Windows dynamic link library used by the install program.
GLOBALS	User-created file that stores parameters applying to the Oracle GoldenGate instance as a whole.
help.txt	Help file for the GGSCI command interface.
icudtxx.dll icuinxx.dll icuucxx.dll	Windows shared libraries for International Components for Unicode, where xx is the currently used version.
jagent.bat	Windows batch file for the Java Agent for Oracle GoldenGate Monitor.
jagent.log jagentjni.log	Log files for the Oracle GoldenGate Monitor Agent.
jagent.sh	UNIX shell script for the Java Agent for Oracle GoldenGate Monitor
LGPL.txt	Lesser General Public License statement. Applies to free libraries from the Free Software Foundation.
libodbc.so	ODBC file for Ingres 2.6 on Unix.
libodbc.txt	License agreement for libodbc.so.
libxml2.dll	Windows dynamic link library containing the XML library for the Oracle GoldenGate XML procedures.
libxml2.txt	License agreement for libxml2.dll .
marker.hist	File created by Replicat if markers were passed from a NonStop source system.
marker_remove.sql	Script that removes the DDL marker table. (Oracle installations)

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

Component	Description
marker_setup.sql	Script that installs the Oracle GoldenGate DDL marker table. (Oracle installations)
marker_status.sql	Script that confirms successful installation of the DDL marker table. (Oracle installations)
notices.txt	Third-party software license file.
odbcinst.ini	Ingres 2.6 on Unix ODBC configuration file.
params.sql	Script that contains configurable parameters for DDL support. (Oracle installations)
pthread-win32.txt	License agreement for pthread-VC.dll .
pthread-VC.dll	POSIX threads library for Microsoft Windows.
prvtclkm.plb	Supports the replication of Oracle encrypted data.
pw_agent_util.bat pw_agent_util.sh	Script files that support the Oracle GoldenGate Monitor Agent.
role_setup.sql	Script that creates the database role necessary for Oracle GoldenGate DDL support. (Oracle installations)
sampleodbc.ini	Sample ODBC file for Ingres 2.6 on UNIX.
sqlldr.tpl	Template for use with Replicat when creating a control file for the Oracle SQL*Loader bulk-load utility.
start.prm stop.prm	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.
xerces-c_2_8.dll	Apache XML parser library.
zlib.txt	License agreement for zlib compression library.

A.4 Oracle GoldenGate Checkpoint Table

When database checkpoints are being used, Oracle GoldenGate creates a checkpoint table with a user-defined name in the database upon execution of the ADD CHECKPOINTTABLE command, or a user can create the table by using the chkpt_*db*_create.sql script (where *db* is an abbreviation of the type of database that the script supports). For a description of this table, see *Administering Oracle GoldenGate for Windows and UNIX*.