# Oracle® GoldenGate

DB2 z/OS Installation and Setup Guide

version 10.4

October 2009



**Oracle® GoldenGate** DB2 z/OS 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 z/OS ..... 3
Supported platforms ..... 3
Operating system requirements ..... 3
Database requirements ..... 6
Supported data types ..... 7
Non-supported data types ..... 8
Supported objects and operations ..... 8
Non-supported objects and operations ..... 8
Supported and non-supported object names and case ..... 8
Object names and owners ..... 9
Case sensitivity ..... 9
Supported characters ..... 9
Non-supported characters ..... 10

**Chapter 2 Installing GoldenGate** ..... 12

Installation overview ..... 12
Upgrades ..... 12
New installations ..... 13
Downloading GoldenGate ..... 13
Setting library paths for dynamic builds ..... 13
Installing GoldenGate ..... 15
Installing the GoldenGate files ..... 15
Configuring Manager and other processes ..... 15
Uninstalling GoldenGate ..... 15

**Chapter 3 Preparing the system for GoldenGate** ..... 17

Configuring database connections ..... 17
Setting initialization parameters ..... 17
Specifying the path to the initialization file ..... 17
Ensuring ODBC connection compatibility ..... 17

Specifying the number of connection threads ... 18
Accessing load modules ... 19
Specifying job names and owners ... 19
Assigning WLM velocity goals ... 19
Monitoring processes ... 20
Viewing GoldenGate messages ... 20
Identifying processes ... 21
Interpreting statistics for update operations ... 21
Using GoldenGate in a data sharing environment ... 21
Adjusting message buffers and paths ... 21
Managing byte-range locking ... 21
Using data compression ... 22
Preparing tables to be processed ... 22
Disabling triggers and cascade delete constraints ... 22
Assigning row identifiers ... 22
Handling ROWID columns ... 23
Configuring character sets ... 24

**Chapter 4 Preparing the transaction logs for GoldenGate** ... 25

Configuring API privileges ... 25
Specifying a bootstrap data set ... 26
Making transaction data available ... 26
Enabling change capture ... 26
Enabling access to log records ... 26
Sizing and retaining the logs ... 27
Using archive logs on tape ... 27
Controlling log flushes ... 27

**Appendix 1 GoldenGate installed components** ... 29

GoldenGate Programs and Utilities ... 29
GoldenGate subdirectories ... 31
Other GoldenGate files ... 33
GoldenGate checkpoint table ... 37

**Index** ... 38

CHAPTER 1

# System requirements and preinstallation instructions

## Overview of GoldenGate for DB2 z/OS

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

## Supported platforms

- DB2 Universal Database (UDB) for OS/390 and z/OS 7.1 (until IBM end of service date)
- DB2 Universal Database (UDB) for z/OS Version 8.1 and 9.1
- GoldenGate supplies Generally Available (GA) software only for versions of z/OS that are officially supported by IBM Corporation. For assistance with running GoldenGate on other z/OS operating systems, contact GoldenGate Support Services.

## Operating system requirements

***System services***

- 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 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 GoldenGate edit and report commands under the TSO/E OMVS command, concatenate the following target libraries to the appropriate ISPF data definition names (ddnames):
    - Dataset SYS1.SBPXPENU to ddname ISPPLIB
    - Dataset SYS1.SBPXMENU to ddname ISPMLIB
    - Dataset SYS1.SBPXTENU to ddname ISPTLIB
    - Dataset SYS1.SBPXEXEC to ddname SYSEXEC or SYSPROC

- Install Recoverable Resource Manager Services (RRS) for the best performance. Depending on the attachment type, the GoldenGate user might need one of the following permissions on the DSNR resource class:
  - If using CAF (the default), RACF ACCESS(READ) to the BATCH environment.
  - If using RRSAF, 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, it is possible for GoldenGate to hold locks on the system catalog until it receives a transaction commit.

***Disk requirements***

- Assign a Hierarchical File System (HFS) volume with 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.
    - 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]')
    ```

- Allocate approximately 50-100 MB of virtual memory for each GoldenGate Extract process, depending upon the amount of uncommitted data per unit of work. 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 datasets with 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.
- GoldenGate supports Sysplex data sharing. For more information, see page page 21.

#### *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 (usually ports 20/21) to GoldenGate for transfers of data, parameters, and reports between source and target systems. Otherwise, provide for another transfer method, such as the 3270 File Transfer program (IND$FILE) supplied by most 3270 terminal emulators, secure FTP, or SSH. 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).

#### *Other operating system requirements*

- To use 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 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.
  - 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.

- 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.

**Table 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)[4] | | X | |
| cmod +rw on the subdirectories in the GoldenGate directory | X | X | X |

[1] The user who starts the Manager process is typically the user by which other 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] Required for the user that installs the Extract program.

## Database requirements

***Database configuration***

- Install a DB2 ODBC driver. The 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.
- Provide GoldenGate EXECUTE privilege on the plan that is specified in the ODBC initialization file (the default is DSNACLI).
- Configure API privileges for GoldenGate to communicate with DB2 under USS. For more information, see page 25.
- 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 (example uses name of DB2A).

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

***Database user***

- By default, the user who starts the Manager process becomes the default DB2 primary authorization ID for all GoldenGate processes that any users start in that GoldenGate instance.You can assign a different user to any process by means of JCL or UNIX variables.

- To monitor GoldenGate processing accurately, do not permit other applications or processes to operate as the GoldenGate user.
- Assign the DB2 privileges listed in Table 2 to the user with 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 2 Privileges needed by GoldenGate for DB2 on z/OS**

| User privilege | Extract | Replicat |
|---|---|---|
| MONITOR2<br>(does not apply to initial-load Extract) | X | |
| SELECT ON these SYSIBM tables:<br>SYSTABLES<br>SYSCOLUMNS<br>SYSTABLEPART<br>SYSKEYS<br>SYSINDEXES<br>SYSCOLAUTH<br>SYSDATABASE<br>SYSFOREIGNKEYS<br>SYSPARMS<br>SYSRELS<br>SYSROUTINES<br>SYSSYNONYMS<br>SYSTABAUTH | X | X |
| SELECT, INSERT, DELETE on SYSIBM.SYSTRIGGERS | 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 | X |
| 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 a GoldenGate process.

## Supported data types

- ASCII or EBCDIC data format. GoldenGate will convert between ASCII and EBCDIC data automatically. Unicode is never converted.
- Most DB2 data types except those listed in “Non-supported data types”.

**Limitations of support**

- NCHAR is supported for like-to-like columns only. Column conversion is not supported.
- GoldenGate's column conversion functions only support tables with EBCDIC encoding.
- 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

- DECFLOAT
- XML
- User-defined types

## Supported objects and operations

- 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.
- 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.
- Extraction and replication of data that is stored using DB2 data compression (CREATE TABLESPACE COMPRESS YES). For more information, see page 22.
- TRUNCATE TABLE is supported, but because this command issues row deletes to perform the truncate, they are shown in GoldenGate statistics as such, and not as a truncate operation. To replicate a TRUNCATE, Replicat uses a DELETE without a WHERE clause.

## Non-supported objects and operations

- 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)

## Supported and non-supported object names and case

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

## Object names and owners

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

## Case sensitivity

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

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

#### To preserve case-sensitivity

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

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

For example:

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

## Supported characters

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

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

| Character | Description |
|---|---|
| ~ | Tilde |
| < > | Greater-than and less-than symbols |
| / | Forward slash |
| \ | Backward slash |
| ! | Exclamation point |
| @ | At symbol |
| # | Pound symbol |
| $ | Dollar symbol |
| % | Percent symbol |

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

| Character | Description |
|---|---|
| ^ | 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 4 Non-supported characters in object and column names[1]**

| Character | Description |
|---|---|
| & | Ampersand |
| * | Asterisk |
| ? | Question mark |
| : | Colon |
| ; | Semi-colon |
| , | Comma |
| ‘ ’ | Single quotes |
| “ ” | Double quotes |

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

| Character | Description |
|---|---|
| ‘ | 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.

#### *Upgrading in the correct order on z/OS*

If upgrading GoldenGate on a z/OS source system that moves data to a non-z/OS target, upgrade the target system first, and then upgrade the source system. No parameter changes are required for any component. You can remove any ASCIITOEBCDIC or EBCDICTOASICC parameters after all of the old data (from before the upgrade) has been processed.

If upgrading GoldenGate on a z/OS source system that moves data to a NonStop target, upgrade the source system first. No parameter changes are required for Replicat, but the following changes are required for Extract, because the Replicat for a NonStop system does not perform automatic conversion:

- If not using a data pump, or if using a data pump with the PASSTHRU parameter, add EBCDICTOASCII to the primary Extract parameter file before restarting Extract after the upgrade.
- If using a data pump with NOPASSTHRU (the default), add EBCDICTOASCII to the data pump parameter file before restarting Extract after the upgrade. No changes are needed for primary Extract.

If upgrading GoldenGate on a NonStop source system that moves data to a z/OS target, upgrade them in any order. However, Replicat will continue to require the parameter ASCIITOEBCDIC, because Extract on NonStop does not currently write the character format to the trail.

### 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<br>◆ IBM z/OS | LIBPATH |
| HP-UX | SHLIB_PATH |
| ◆ Sun Solaris<br>◆ HP Tru64 (OSF/1)<br>◆ 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

## 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. Set the LIBPATH environment variable to include the path to the GoldenGate installation directory. The "." indicates the current directory. This change can be made system-wide, for a specific user at login time, or manually each time GoldenGate is executed, but it must be performed before starting any GoldenGate processes.

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

6. From the GoldenGate directory, run the GGSCI program.

   ```
   GGSCI
   ```

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

   ```
   CREATE SUBDIRS
   ```

8. 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*.

# 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.

**To uninstall GoldenGate**

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.

CHAPTER 3

# Preparing the system for GoldenGate

## Configuring database connections

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

### Setting initialization parameters

The following DB2 initialization parameters apply to GoldenGate and must be set correctly before starting 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.

**NOTE** When using the CAF attachment type, you must use the GoldenGate parameter 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 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`."

### 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"
```

### Ensuring ODBC connection compatibility

To ensure that you configure the DB2 ODBC initialization file correctly, follow the guidelines set forth in the *DB2 UDB for z/OS ODBC Guide and Reference* manual.

One common mistake that is made when configuring the initialization file is worth noting here: the coding of square brackets; that is, the [ and ] characters. 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:

- 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).

DB2 ODBC does not recognize brackets of any other CCSID.

**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 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.

## Specifying the number of connection threads

Every 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)

**To determine the number of threads required**

If using RRSAF, allow:

- Two DB2 threads per process for each of the following:
  - Extract
  - Replicat
  - The GGSCI command DBLOGIN (logs into the database)
  - DDLGEN (creates DDL statements to create target tables)
  - DEFGEN (generates data definitions for column mapping)
- One extra DB2 thread for Extract for IFI calls.
- One extra DB2 thread for each SQLEXEC parameter statement used by each Extract and Replicat process. For more information about SQLEXEC, see the *GoldenGate for Windows and UNIX Reference Guide*.

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

## Accessing load modules

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

- the z/OS system search order.

  or...

- the USS profile of the GoldenGate user. Use a UNIX command similar to the following, where DSN810 is the user-assigned dataset 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 GoldenGate process. If using APF, all libraries in the STEPLIB concatenation must be APF-authorized. See "Configuring API privileges" on page 25.

## Specifying job names and owners

By default, USS sets the job name and owner of all 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 GoldenGate processes. For more details, see the *z/OS System Services Planning* document.

## Assigning WLM velocity goals

The user who starts the Manager process is typically the user by which other GoldenGate processes run. GoldenGate work appears as forked child processes of WLM subsystem type OMVS. Assign the 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 the *GoldenGate for Windows and UNIX Administrator Guide*.
- 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 GoldenGate utilities, such as DDLGEN, DEFGEN and LOGDUMP, only occasionally, so you can let them perform like the other UNIX terminal-oriented work.
- If using SQLEXEC stored procedures, make certain that they do not become a bottleneck in the GoldenGate processing. 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 GoldenGate under the TSO/E OMVS command, the 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.

#### Sample WLM Solution

You can use multiple WLM service classes for the GoldenGate processes. The following is an example of how to maintain relative priorities for 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 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.

## Monitoring processes

The following provides information about monitoring GoldenGate with z/OS system facilities.

### Viewing GoldenGate messages

If the system log process (syslog daemon syslogd) is running, USS routes 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, GoldenGate writes its command output, status information, and error messages to the system console. You can redirect console messages to the GoldenGate USS session and to the GoldenGate report files by using the following UNIX command:

```
export _BPXK_JOBLOG=STDERR
```

### Identifying processes

The system management facility (SMF) typically creates a separate accounting record for each UNIX process, including 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 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 GoldenGate processes:

- Extract
- Replicat
- Manager, if performing maintenance on GoldenGate tables. Examples of GoldenGate tables are the marker table and the Replicat checkpoint table.
- GGSCI users who use the GoldenGate DBLOGIN command to log into the database.

### Interpreting statistics for update operations

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

## Using GoldenGate in a data sharing environment

The following are aspects of the DB2 data sharing configuration that affect GoldenGate.

### Adjusting message buffers and paths

The increased intersystem communication that is generated by a shared HFS configuration affects system response time and throughput. To control this traffic, you might need to increase the number and size of the message buffers within the Cross-System Coupling Facility (XCF) and define additional message paths. For more information, see the IBM Sysplex documentation.

### Managing byte-range locking

If possible, use a distributed byte-range locking manager (BRLM). In this configuration, each system in the group runs a separate BRLM dedicated to the file systems that are mounted on that system.

If using a central BRLM to handle byte-range locking for all members of a group, use the AUTORESTART parameter in the Manager parameter file. AUTORESTART restarts any GoldenGate processes in the group that fail if the member hosting the BRLM fails. GoldenGate does not handle I/O errors caused by lock failures.

## Using data compression

To prevent problems with dictionary mismatches, always use the KEEPDICTIONARY option with LOAD and REORG utility statements. Recovering or loading data with a compression dictionary that is different from the current one might cause the data to become unreadable by GoldenGate. Before starting LOAD or REORG activities, allow the GoldenGate extraction and replication processes to finish their work, and then stop them. One symptom of mismatched compression dictionaries is an Extract failure that returns DB2 reason codes 00C90063 and 00C90064.

## Preparing tables to be processed

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.

## Handling ROWID columns

Any attempt to insert into a target table that includes a column with 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 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 "Assigning row identifiers" on page 22.

## Configuring character sets

To ensure accurate character representation from one database to another, the following must be true:

- The character set of the target database must be a superset of the character set of the source database.
- If your client applications use different character sets, the database character set must be a superset of the character sets of the client applications. In this configuration, every character is represented when converting from a client character set to the database character set.

CHAPTER 4

# Preparing the transaction logs for GoldenGate

## Configuring API privileges

The Extract process uses a privileged API by the name of *IFI IFCID 306*[1] to get data from the transaction log. Extract must use elevated privileges to:

- 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 returned, plus the additional area defined in DB2 macro DSNDQW04.

After performing that work, Extract reverts to normal application privileges for all other work.

As part of the GoldenGate setup, grant Extract the elevated privileges as follows.

**To grant Extract API privileges**

1. Log in as a superuser or as a user that has the following RACF permissions (or other security equivalents).

   RACF ACCESS(READ) to resource BPX.FILEATTR.APF in CLASS(FACILITY)

2. In the UNIX shell, issue the following command to add authorization in the Authorized Program Facility (APF) for the Extract process.

   ```
   extattr +a extract
   ```

3. Verify the authorization by issuing the following command.

   ```
   ls -E extract
   ```

In the output, look for the "a" attribute, indicating "APF-authorized" as shown in the following example.

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

**NOTE** The privileges that you specify also affect user exits called by Extract. A user exit also must be privileged, or else it must run as part of an unprivileged Extract data pump, rather than as part of the privileged primary Extract group that reads from the logs.

[1] IFI is IBM's Instrumentation Facility Interface. IFCID is IBM's Instrumentation Facility Counter ID.

## 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. GoldenGate does not perform any validations of the BSDS specification.

## Making transaction data available

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.

### 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 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>
   ```

By default, ADD TRANDATA issues the following command:

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

### 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 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.

## 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.

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

If data that Extract needs during processing was not retained, either in online or 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 GoldenGate environment over again.

#### To control log retention

Use the DSN6LOGP MAXARCH system parameter in the DSNTIJUZ installation job.

**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.

## Using archive logs on tape

GoldenGate can read DB2 archive logs on tape, but it will cause performance delays. For example, DB2 reserves taped archives for a single recovery task. Therefore, Extract would be unable to read an archive tape that is being used to recover a table, until the recovery is finished. You could use DFHSM or 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.

## 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 the flushes**

Use the Extract parameter TRANLOGOPTIONS with the NOFLUSH option.

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 5 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 5 Programs and utilities (continued)**

| Program | Description |
|---|---|
| ggsmgr.jcl<br>ggsmgr.proc<br>ggsmgrst.jcl<br>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 6 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.<br>File name format is <group name><sequence number>.<ext> where <sequence number> is a sequential number appended to aged files and <ext> is either cpe for Extract checkpoint files or cpr for Replicat checkpoint files.<br>Do not edit these files.<br>Examples:<br>ext1.cpe<br>rep1.cpr |
| 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.<br>File name format is a user-defined two-character prefix followed by either a six-digit sequence number (trail files) or the user-defined name of the associated Extract process group (extract files).<br>Do not edit these files.<br>Examples:<br>rt000001<br>finance |
| dirdef | The default location for data definitions files created by the DEFGEN utility to contain source or target data definitions used in a heterogeneous synchronization environment. Written in external ASCII. File name format is a user-defined name specified in the DEFGEN parameter file.<br>These files may be edited to add definitions for newly created tables. If you are unsure of how to edit a definitions file, contact GoldenGate technical support.<br>Example:<br>defs.dat |
| dirout | This directory is not used any more. |

**Table 6 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).<br>These files are only created while a process is running. The file shows the program name, the process name, the port number, and the process ID.<br>Do not edit these files.<br>Examples:<br>mgr.pcm<br>ext.pce |
| dirprm | The default location for 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.<br>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.<br>Examples:<br>defgen.prm<br>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.<br>File name format is <group name><sequence number>.rpt where <sequence number> is a sequential number appended to aged files.<br>Do not edit these files.<br>Examples:<br>fin2.rpt<br>mgr4.rpt |
| dirsql | 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.<br>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.<br>These scripts can be edited if needed.<br>Examples:<br>ggslog.sql<br>account.sql |

**Table 6 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 7 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 7 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 7 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 7 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<br>stop.prm | z/OS paramlib members to start and stop the Manager process. |
| startmgr<br>stopmgr | z/OS Unix System Services scripts to start the Manager process from GGSCI. |
| startmgrcom<br>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 8 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

**$LD_LIBRARY_PATH variable** 14
**$PATH variable** 14

## A

**accounting record for GoldenGate** 21
**API privileges for GoldenGate** 6, 25
**archive logs**
  on tape 27
  retaining 27
**ASCII, converting to EBCDIC** 7
**attachment type** 4
**AUTORESTART parameter** 21

## B

**bootstrap data set, specifying** 26
**BPX_JOBNAME environment variable** 19
**BPX_SHAREAS environment variable** 21
**BPX_USERID environment variable** 19
**byte range locking, managing** 21

## C

**C Compiler** 5
**CAF attachment type** 4
**cascade deletes, disabling** 22
**case, supported** 8
**change capture, enabling** 26
**character sets, configuring** 24
**characters**
  ASCII, EBCDIC conversion 7
  supported in object names 9
**cluster, installing on** 4
**columns**
  number and size supported 8
  supported data types 7
**compression, data** 8, 22
**connection threads, number of** 18
**connection, database** 6, 17, 18
**constraints, integrity** 22
**CREATE SUBDIRS command** 15
**Cross-System Coupling Facility** 21

## D

**DATA CAPTURE CHANGES** 26
**data compression** 22
**data sharing, and GoldenGate** 21
**data types, supported** 7
**database**
  character sets 24
  connections 17
  logs, preparing 25
  requirements 6
  tables, preparing to be processed 22
**DBOPTIONS parameter** 17
**DDL operations** 8
**DECFLOAT data type** 8
**deletes, cascaded** 22
**disk requirements** 4
**downloading GoldenGate** 13
**DSN6LOGP MAXARCH system parameter** 27
**DSN6SYSP MON** 26
**DSNACLI plan name** 17
**DSNAOINI environment variable** 17
**DSNHDECP load module, access to** 19
**DSNTIJUZ installation job** 26, 27
**DSNTIPE** 18

## E

**EBCDIC, converting to from ASCII** 7
**exit routines** 8

## F

**file system, dedicated** 4
**firewall, configuring** 5
**flushes of logs, controlling** 27
**FTP access for GoldenGate** 5

## G

**ggmessage.dat file** 35
**GoldenGate**
downloading 13
installed programs and files 29
installing 12
uninstalling 15
**Gzip** 5

## H

**HIPER maintenance** 5

## I

**IFI IFCID 306** 25
**initialization file, path to** 17
**initialization parameters, setting** 17
**installing GoldenGate** 12

## J

**job name and owner, GoldenGate** 19

## K

**KEEPDICTIONARY option, LOAD and REORG** 22
**key**
assigning 22
name, supported characters 9
storage in ECSA 25
**KEYCOLS option, TABLE or MAP** 23

## L

**large objects, limitations on** 8
**LE Runtime Libraries** 3
**LIBPATH variable** 14
**load modules, accessing** 19
**LOCATION parameter** 17
**locks, managing in Sysplex** 21

## M

**MAX BATCH CONNECT parameter** 18
**MAX TSO CONNECT parameter** 18
**MAX USERS parameter** 18
**message buffers, adjusting** 21
**messages, GoldenGate** 20
**MONITOR TRACE** 26
**Monitor Trace Class 1, enabling** 26
**monitoring, guidelines for** 20
**MULTICONTEXT parameter** 17
**MVSATTACHTYPE parameter** 17
**MVSDEFAULTSSID parameter** 17

## N

**name**
non-supported characters in 10
supported characters in 9
**names, supported** 8
**NOCATALOGCONNECT option, DBOPTIONS** 17
**NOFLUSH option, TRANLOGOPTIONS** 28

## O

**OMVS command, running GoldenGate under** 20
**Open Database Connectivity (ODBC) requirements** 6
**Open MVS, DB2** 3
**operating system requirements** 3

## P

**permissions**
database 6
operating system 6
**PLANNAME parameter** 17

**platforms, supported** 3
**ports, required by GoldenGate** 5
**privileged API, using** 25
**privileges**
database 6
operating system 6
**processes, monitoring** 20
**Program Temporary Fixes** 5

## R

**RACF command processor** 5
**Recoverable Resource Manager Services** 4
**ROWID columns, correct definition for** 23
**rows, number and size supported** 8
**RRSAF attachment type** 4

## S

**SCEERUN library** 3
**SDSNLOAD library, access to** 19
**SHAREAS option, OMVS command** 21
**SHLIB_PATH variable** 14
**SMF record for GoldenGate** 21
**spaces**
address
*and velocity goals* 19
*processes in same* 21
in object and column names 11
**SQLEXEC, and velocity goals** 19
**square brackets, proper coding of** 17
**statistics for update operations** 21
**STEPLIB environment variable** 3
**subdirectories, creating** 15
**SYSIBM.LOCATIONS table** 6
**syslogd, and GoldenGate messages** 20
**Sysplex, GoldenGate in** 21

## T

**tables**
creating with DATA CAPTURE CHANGES 27
preparing for processing 22
supported kinds 8
SYSIBM, select on 7
**tape, archive logs on** 27
**TCP/IP, configuring** 5
**Thread Management Panel** 18
**threads, connection** 18
**Time Sharing Option Extensions** 5
**TRANLOGOPTIONS parameter** 28
**transaction logs**
active, access to 26
API privileges 25
archived on tape 27
enabling change capture 26
flushes, controlling 27
initial position in 26
sizing and retaining 27
**triggers, disabling on target** 22
**TRUNCATE TABLE** 8

## U

**uninstalling GoldenGate** 15
**Unix System Services** 3
**updates, statistics for** 21
**user, GoldenGate** 4, 6
**user-defined types** 8

## V

**VAMSERV program** 30
**velocity goals, setting** 19
**virtual machine, support for** 6
**virtual memory requirements** 4

## W

**WLM velocity goals, setting** 19
**Workload Manager velocity goals** 19

## X

**XML data type** 8