Oracle® Fusion Middleware Installing and Configuring Oracle GoldenGate for DB2 LUW





Oracle Fusion Middleware Installing and Configuring Oracle GoldenGate for DB2 LUW, 12c (12.2.0.1)

E66358-02

Copyright © 2013, 2017, 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 is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience		١
	ation Accessibility	`
Related In	formation	\
Conventio	ns	V
System	Requirements and Preinstallation Instructions	
1.1 Veri	fying Certification and System Requirements	1-
1.2 Ope	rating System Requirements	1-
1.2.1	Memory Requirements	1-
1.2.2	Disk Requirements	1-
1.2.3	Network	1-
1.2.4	Operating System Privileges	1-
1.2.5	Console	1-
1.2.6	Other Programs	1-
1.3 Data	abase Requirements	1-
1.3.1	Database Configuration	1-
1.3.2	Database User for Oracle GoldenGate Processes	1-
1.4 Sup	ported DB2 LUW Data Types	1-
1.5 Non	-Supported DB2 LUW Data Types	1-
1.6 Sup	ported Objects and Operations for DB2 LUW	1-
1.7 Non	-Supported Objects and Operations for DB2 LUW	1-
1.8 Sup	ported Object Names	1-
Installir	ng Oracle GoldenGate	
2.1 Insta	allation Overview	2-
2.2 Cho	osing an Installation System for DB2	2-
2.2.1	Choosing and Configuring a System for Remote Capture or Delivery	2-
2.3 Und	erstanding and Obtaining the Oracle GoldenGate Distribution	2-
2.4 Sett	ing Library Paths for Dynamic Builds on UNIX	2-
	paring to Install Oracle GoldenGate within a Cluster	2-



2.5.1	Cluster	2-4
2.6 Ins	talling Oracle GoldenGate on Linux and UNIX	2-5
2.6.1	Installing the Oracle GoldenGate Files	2-5
2.7 Ins	talling Oracle GoldenGate on Windows	2-5
2.7.1	Installing Oracle GoldenGate into a Windows Cluster	2-6
2.7.2	Installing the Oracle GoldenGate Files	2-6
2.7.3	Specifying a Custom Manager Name	2-6
2.7.4	Installing Manager as a Windows Service	2-7
2.8 Inte	egrating Oracle GoldenGate into a Cluster	2-8
2.8.1	General Requirements in a Cluster	2-8
2.8.2	Adding Oracle GoldenGate as a Windows Cluster Resource	2-9
Prepai	ing the System for Oracle GoldenGate	
3.1 Co	nfiguring the Transaction Logs for Oracle GoldenGate	3-1
3.1.1	Retaining the Transaction Logs	3-1
3.1.2	Specifying the Archive Path	3-2
3.2 Pre	paring Tables for Processing	3-2
3.2.1	Disabling Triggers and Cascade Constraints	3-2
3.2.2	Assigning Row Identifiers	3-3
3	.2.2.1 How Oracle GoldenGate Determines the Kind of Row Identifier to Use	3-3
3	.2.2.2 Using KEYCOLS to Specify a Custom Key	3-3
3.2.3	Preventing Key Changes	3-3
3.2.4	Enabling Change Capture	3-4
3.2.5	Maintaining Materialized Query Tables	3-4
3.3 Se	ting the Session Character Set	3-5
3.4 Pre	paring for Initial Extraction	3-5
3.5 Sp	ecifying the DB2 LUW Database in Parameter Files	3-6
Config	uring Oracle GoldenGate for DB2 LUW	
4.1 Wh	at to Expect from these Instructions	4-1
4.2 Wh	ere to Get More Information	4-1
4.3 Co	nfiguring the Primary Extract	4-2
4.4 Co	nfiguring the Data Pump Extract	4-3
4.5 Co	nfiguring Replicat	4-4
4.5.1	Creating a Temporal Table	4-4
4	.5.1.1 Support for Temporal Tables	4-5
4	.5.1.2 Replicating with Temporal Tables	4-5
4	.5.1.3 Converting	4-6



	4	4.5.2 Creating a Checkpoint Table	4-10
	4	4.5.3 Configuring the Replicat Parameter File	4-10
	4.6	Next Steps in the Deployment	4-11
	4.7	When to Start Replicating Transactional Changes	4-12
	4.8	Testing Your Configuration	4-12
5	Uni	installing Oracle GoldenGate	
	5.1	Stopping Processes	5-1
	5.2	Removing the Oracle GoldenGate Database Setup	5-1
	5.3	Removing Oracle GoldenGate Windows Components	5-2
	5.4	Manually Removing the Oracle GoldenGate Files	5-3
А	Ora	acle GoldenGate Installed Components	
	A.1	Oracle GoldenGate Programs and Utilities	A-1
	A.2	Oracle GoldenGate Subdirectories	A-2
	A.3	Other Oracle GoldenGate Files	A-4
	A.4	Oracle GoldenGate Checkpoint Table	A-8
	Ind	lex	



Preface

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

This documentation is meant to be a step by step guide in establishing a basic Oracle GoldenGate configuration that is tailored to the DB2 LUW environment. It should be followed in a linear fashion so that you benefit from important information in previous sections. It is not meant to be used as a reference document. Where needed, it points you to other documentation where you can find additional information to expand the configuration to suit your needs.

Audience

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

Documentation Accessibility

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

Accessible Access to Oracle Support

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

Related Information

The Oracle GoldenGate Product Documentation Libraries are found at

Oracle GoldenGate

Oracle GoldenGate Application Adapters

Oracle GoldenGate for Big Data

Oracle GoldenGate Plug-in for EMCC

Oracle GoldenGate Monitor

Oracle GoldenGate for HP NonStop (Guardian)



Oracle GoldenGate Veridata

Oracle GoldenGate Studio

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

Oracle GoldenGate A-Team Chronicles

Conventions

The following text conventions are used in this document:

Convention	Meaning	
boldface	Boldface type indicates graphical user interface elements associated with an action, such as "From the File menu, select Save ." Boldface also is used for terms defined in text or in the glossary.	
italic	Italic type indicates placeholder variables for which you supply	
italic	particular values, such as in the parameter statement: TABLE table_name. Italic type also is used for book titles and emphasis.	
monospace	Monospace type indicates code components such as user exits and	
MONOSPACE	scripts; the names of files and database objects; URL paths; and injudy and output text that appears on the screen. Uppercase monospace type is generally used to represent the names of Oracle GoldenGat 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 includes the following sections:

- · Verifying Certification and System Requirements
- Operating System Requirements
- Database Requirements
- Supported DB2 LUW Data Types
- Non-Supported DB2 LUW Data Types
- Supported Objects and Operations for DB2 LUW
- Non-Supported Objects and Operations for DB2 LUW
- Supported Object Names

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 describes operating system requirements.

1.2.1 Memory Requirements

The amount of memory that is required for Oracle GoldenGate depends on the amount of data being processed, the number of Oracle GoldenGate processes running, the amount of RAM available to Oracle GoldenGate, and the amount of disk space that is available to Oracle GoldenGate for storing pages of RAM temporarily on disk when the operating system needs to free up RAM (typically when a low watermark is reached). This temporary storage of RAM to disk is commonly known as *swapping* or *paging* (herein referred to as *swapping*). Depending on the platform, the term *swap space* can be a swap partition, a swap file, a page file (Windows) or a shared memory segment (IBM i platforms).

Modern servers have sufficient RAM combined with sufficient swap space and memory management systems to run Oracle GoldenGate. However, increasing the amount of RAM available to Oracle GoldenGate may significantly improve its performance, as well as that of the system in general.

Typical Oracle GoldenGate installations provide RAM in multiples of gigabytes to prevent excessive swapping of RAM pages to disk. The more contention there is for RAM the more swap space that is used.

Excessive swapping to disk causes performance issues for the Extract process in particular, because it must store data from each open transaction until a commit record is received. If Oracle GoldenGate runs on the same system as the database, the amount of RAM that is available becomes critical to the performance of both.

RAM and swap usage are controlled by the operating system, not the Oracle GoldenGate processes. The Oracle GoldenGate cache manager takes advantage of the memory management functions of the operating system to ensure that the Oracle GoldenGate processes work in a sustained and efficient manner. In most cases, users need not change the default Oracle GoldenGate memory management configuration.

For more information about evaluating Oracle GoldenGate memory requirements, see the CACHEMGR parameter in *Reference for Oracle GoldenGate for Windows and UNIX*.

1.2.2 Disk Requirements

Assign the following free disk space:

- 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.
- Allow at least an additional 1 GB of disk space on any system that hosts Oracle GoldenGate trails, which are files that contain the working data. You may need more or less than this amount, because the space that is consumed by the trails depends on the volume of data that will be processed. See the guidelines for sizing trails in Administering Oracle GoldenGate for Windows and UNIX.
- By default, Oracle GoldenGate maintains data that it swaps to disk in the dirtmp sub-directory of the Oracle GoldenGate installation directory. The cache manager assumes that all of the free space on the file system is available. This directory can fill up quickly if there is a large transaction volume with large transaction sizes. To prevent I/O contention and possible disk-related Extract failures, dedicate a disk to this directory. You can assign a name and size to this directory with the CACHEDIRECTORY option of the CACHEMGR parameter. The CACHESIZE option of CACHEMGR sets a soft limit for the amount of virtual memory (cache size) that is available for caching transaction data. See Reference for Oracle GoldenGate for Windows and UNIX for the default values of these options and detailed explanations, in case system adjustments need to be made.

1.2.3 Network

The following network resources must be available to support Oracle GoldenGate.



- 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 the 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.

1.2.4 Operating System Privileges

The following are the privileges in the operating system that are required to install Oracle GoldenGate and to run the processes.

- To install on Windows, the person who installs Oracle GoldenGate must log in as Administrator.
- To install on Windows, the person who installs Oracle GoldenGate must log in as Administrator.
- To install on UNIX, the person who installs Oracle GoldenGate must have read and write privileges on the Oracle GoldenGate installation directory.
- The Oracle GoldenGate Extract, Replicat, and Manager processes must operate
 as an operating system user that has privileges to read, write, and delete files and
 subdirectories in the Oracle GoldenGate directory. In addition, the Manager
 process requires privileges to control the other Oracle GoldenGate processes.
- Dedicate the Extract, Replicat, and Manager operating system users to Oracle GoldenGate.

1.2.5 Console

The operating system and the command console must have the same character sets. Mismatches occur on Microsoft Windows systems, where the operating system is set to one character set, but the DOS command prompt uses a different, older DOS character set. Oracle GoldenGate uses the character set of the operating system to send information to GGSCI command output; therefore a non-matching console character set causes characters not to display correctly. You can set the character set of the console before opening a GGSCI session by using the following DOS command:

chcp character_set

If the characters do not display correctly after setting the code page, try changing the console font to Lucida Console, which has an extended character set.



1.2.6 Other Programs

The following are additional considerations in support of Oracle GoldenGate.

- Before installing Oracle GoldenGate on a Windows system, install and configure
 the Microsoft Visual C ++ 2010 SP1 Redistributable Package. Make certain it is
 the SP1 version of this package, and make certain to get the correct bit version for
 your server. This package installs runtime components of Visual C++ Libraries.
 For more information, and to download this package, go to http://
 www.microsoft.com.
- Oracle has not certified any of its products on VMware virtualized environments.
 Oracle Support will assist customers running Oracle products on VMware in the following manner: Oracle will only provide support for issues that either are known to occur on the native OS, or can be demonstrated not to be as a result of running on VMware.

1.3 Database Requirements

This section describes database requirements.

1.3.1 Database Configuration

- The Oracle GoldenGate Extract process calls the DB2READLOG function in the Administrative API to read the transaction log files of a DB2 LUW source database. In addition to DB2READLOG, Extract uses a small number of other API routines to check the source database configuration on startup.
- The Oracle GoldenGate Replicat process uses the DB2 CLI interface on a DB2 LUW target database. For instructions on installing this interface, see the DB2 documentation.
- The database can reside on a different server from the one where Oracle GoldenGate is installed, so long as the database is defined locally. For example, the following enables you to use database mydb locally with data that is on abc123:

catalog tcpip node abc123 remote abc123.us.mycompany.com server 00000catalog db mydb as abc123 at node abc123 AUTHENTICATION server

1.3.2 Database User for Oracle GoldenGate Processes

- Create a database user that is dedicated to Oracle GoldenGate. It can be the same user for all of the Oracle GoldenGate processes that must connect to a database:
 - Extract (source database)
 - Replicat (target database)
 - Defgen (source or target database)
- To preserve the security of your data, and to monitor Oracle GoldenGate
 processing accurately, do not permit other users, applications, or processes to log
 on as, or operate as, the Oracle GoldenGate database user. It is recommended
 that you store the login credentials in an Oracle GoldenGate credential store. The
 credential store makes use of local secure storage for the login names and



passwords, and permits you to specify only an alias in the Oracle GoldenGate parameter files. For more information about this option, as well as alternative security options, see *Administering Oracle GoldenGate for Windows and UNIX*.

Assign system administrator (SYSADM) or database administrator (DBADM) authority to
the database user under which Extract runs. To give the Extract user DBADM
authority, a user with SYSADM authority can issue the following grant statement.

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

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

Note:

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

```
[SC=-1224: QL1224N A database agent could not be started to service a request, or was terminated as a result of a database system shutdown or a force command.

SQL STATE [5032: The CONNECT statement is invalid, because the database manager was stopped after this application was started]
```

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

1.4 Supported DB2 LUW Data Types

Oracle GoldenGate supports all DB2 LUW data types, except those listed in Non-Supported DB2 LUW Data Types.

Limitations of Support

- Oracle GoldenGate supports multi-byte character data types and multi-byte data stored in character columns. Multi-byte data is only supported in a like-to-like configuration. Transformation, filtering, and other types of manipulation are not supported for multi-byte character data.
- BLOB and CLOB columns must have a logged clause in their definitions.
- GRAPHIC and VARGRAPHIC columns must be in a database where the character set is UTF16. Any other character set causes the Oracle GoldenGate to abend.
- 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.



- Extract fully supports the capture and apply of TIMESTAMP(0) through TIMESTAMP(9). Extract also captures TIMESTAMP(7) through TIMESTAMP(12), but it truncates the data to nanoseconds (maximum of nine digits of fractional time) and issues a warning to the error log. Replicat truncates timestamp data from other sources to nanoseconds when applying it to TIMESTAMP(7) through TIMESTAMP(12) in a DB2 LUW target.
- Oracle GoldenGate supports timestamp data from 0001/01/03:00:00:00:00 to 9999/12/31:23:59:59. If a timestamp is converted from GMT to local time, these limits also apply to the resulting timestamp. Depending on the timezone, conversion may add or subtract hours, which can cause the timestamp to exceed the lower or upper supported limit.
- Oracle GoldenGate does not support the filtering, column mapping, or manipulation of large objects that are larger than 4K. Full Oracle GoldenGate functionality can be used for objects that are 4K or smaller.
- Replication of XML columns between source and target databases with the same character set is supported. If the source and target database character sets are different, then XML replication may fail with a database error because some characters may not be recognized (or valid) in the target database character set.

1.5 Non-Supported DB2 LUW Data Types

- XMLType
- DECFLOAT
- · User-defined types
- Negative dates

1.6 Supported Objects and Operations for DB2 LUW

- Oracle GoldenGate supports the maximum number of columns and column size per table that is supported by the database.
- TRUNCATE TABLE for DB2 LUW version 9.7 and later.
- Multi Dimensional Clustered Tables (MDC) for DB2 LUW 9.5 and later.
- Materialized Query Tables. Oracle GoldenGate does not replicate the MQT itself, but only the base tables. The target database automatically maintains the content of the MQT based on the changes that are applied to the base tables by Replicat.
- Tables with ROW COMPRESSION. In DB2 LUW version 10.1 and later, COMPRESS YES
 STATIC is supported and COMPRESS YES ADAPTIVE are supported. To support COMPRESS
 YES in DB2 LUW versions 9.7 and earlier, the TRANLOGOPTIONS parameter with the
 ALLOWTABLECOMPRESSION option must be used, and the compressed table cannot
 contain LOBs.
- Extended row size feature is enabled by default. It is supported with a workaround using FETCHCOLS. For any column values that are VARCHAR OF VARGRAPHIC data types and are stored out of row in the database, you must fetch these extended rows by specifying these columns using the FETCHCOLS option in the TABLE parameter in the extract parameter file. With this option set, when the column values are out of row then Oracle GoldenGate will fetch its value. If the value is out of and FETCHCOLS is not specified then Extract will abend to prevent any data loss. If you do not want to use this feature, set the extended_row_size parameter to DISABLE.



- Temporal tables with DB2 LUW 10.1 FixPack 2 and greater are supported. This is the default for Replicat.
- Limitations on Automatic Heartbeat Table support are as follows:
 - Oracle GoldenGate heartbeat parameters frequency and purge frequency are accepted in seconds and days. However, the DB2 LUW task scheduler accepts its schedule only in cron format so the Oracle GoldenGate input value to cron format may result in some loss of accuracy. For example:

```
ADD HEARTBEATTABLE, FREQUENCY 150, PURGE_FREQUENCY 20
```

This example sets the FREQUENCY to 150 seconds, which is converted to the closest minute value of 2 minutes, so the heartbeat table is updated every 120 seconds instead of every 150 seconds. Setting PURGE_FREQUENCY to 20 means that the history table is purged at midnight on every 20th day.

- The following are steps are necessary for the heartbeat scheduled tasks to run:
 - 1. Set the DB2_ATS_ENABLE registry variable to db2set DB2_ATS_ENABLE=YES.
 - **2.** Create the SYSTOOLSPACE tablespace if it does not already exist:

```
CREATE TABLESPACE SYSTOOLSPACE IN IBMCATGROUP MANAGED BY AUTOMATIC STORAGE EXTENTSIZE 4
```

Ensure instance owner has Database administration authority (DBADM):

GRANT DBADM ON DATABASE TO instance_owner_name

1.7 Non-Supported Objects and Operations for DB2 LUW

- Schema, table or column names that have trailing spaces.
- Multiple instances of a database
- Datalinks
- Extraction or replication of DDL (data definition language) operations
- Generated columns (GENERATE ALWAYS clause).
- Row size support. This feature is enabled by default in DB2 LUW 10.5 and later, but must be disabled to support Oracle GoldenGate.
- Tables with value compression



To include tables with non-supported types of table compression in the Oracle GoldenGate configuration, deactivate the non-supported compression and then reorganize the tables; otherwise, exclude them from the Oracle GoldenGate configuration.



1.8 Supported Object Names

For a list of characters that are supported in object names, see *Administering Oracle GoldenGate for Windows and UNIX*.



2

Installing Oracle GoldenGate

This chapter describes how to install Oracle GoldenGate and 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*.

Topics:

2.1 Installation Overview

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 at:

Overview of Upgrading Oracle GoldenGateUpgrading Oracle GoldenGate for Windows and UNIX

2.2 Choosing an Installation System for DB2

To install Oracle GoldenGate for DB2 LUW, you can use either of the following configurations:

- Install Oracle GoldenGate on the DB2 database server. See one of the following:
 - Installing Oracle GoldenGate on Linux and UNIX
 - Installing Oracle GoldenGate on Windows
- Install Oracle GoldenGate on another server, and configure Oracle GoldenGate to connect remotely to the database server through DB2 Connect. All of the Oracle GoldenGate functionality that is supported for DB2 LUW is supported in this configuration. To use this option, proceed to Choosing and Configuring a System for Remote Capture or Delivery.



2.2.1 Choosing and Configuring a System for Remote Capture or Delivery

In a remote installation, you install Oracle GoldenGate on a server that is remote from the source or target database server. This server can be any Linux, UNIX, or Windows platform that Oracle GoldenGate supports for the DB2 for LUW database. The Oracle GoldenGate build must match the version of DB2 LUW that is running on the installation server.



Remote capture is not supported if your database is running on a Big Endian (AIX) server and the Oracle GoldenGate capture is running on a Little Endian (Linux) server.

In this configuration, the location of the database is transparent to Extract and Replicat. Extract can read the DB2 logs on a source DB2 LUW database server, and Replicat can apply data to a target DB2 LUW server.

To Configure Remote Capture or Delivery:

- Install and run DB2 for LUW on the remote server that has DB2 Connect.
- Catalog the remote server in the DB2 source or target database by using the following DB2 command.

```
catalog tcpip node db2_node_name remote remote_DNS_name
```

3. Catalog the DB2 target node in the DB2 for LUW database on the remote server by using the following DB2 command:

```
catalog tcpip node db2_node_name remote remote_DNS_name
server remote_port_number
```

4. Add the DB2 source or target database to the DB2 catalog on the remote server by using the following DB2 command:

catalog db database_name as database_alias at node db_node_name



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

Download and install the Oracle GoldenGate build that is appropriate for the DB2 LUW database on the remote server.



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*.

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:

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

2.4 Setting Library Paths for Dynamic Builds on UNIX

Oracle GoldenGate uses shared libraries. When you install Oracle GoldenGate on a UNIX system, the following must be true *before you run GGSCI or any other Oracle GoldenGate process*.

If you will be running an Oracle GoldenGate program from outside the Oracle GoldenGate installation directory on a UNIX system:

- (Optional) Add the Oracle GoldenGate installation directory to the PATH environment variable.
- (Required) Add the Oracle GoldenGate installation directory to the shared-libraries environment variable.

For example, given an Oracle GoldenGate installation directory of /users/ogg, the second command in the following example requires these variables to be set:

Command	Requires GG libraries in environment variable?	
\$ users/ogg > ./ggsci	No	
<pre>\$ users > ./ogg/ggsci</pre>	Yes	

To Set the Variables in Korn Shell

PATH=installation_directory: \$PATH
export PATH
shared_libraries_variable=absolute_path_of_installation_directory: \$shared_libraries
s_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_librarie
s_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 variables shown in Table 2-1:

Table 2-1 UNIX/Linux Library Path Variables Per Platform

Platform	Environment variable
IBM AIX	LIBPATH
HP-UX	SHLIB_PATH
Sun Solaris	LD_LIBRARY_PATH
LINUX	

The following is an example of how to set the path in Bourne shell:

export LD_LIBRARY_PATH=/ggs/11.0:\$LD_LIBRARY_PATH



To view the libraries that are required by an Oracle GoldenGate process, use the 1dd <code>goldengate_process</code> shell command before starting the process. This command also shows an error message for any that are missing.

2.5 Preparing to Install Oracle GoldenGate within a Cluster

This topic covers the installation requirements that apply when Oracle GoldenGate will be installed in a cluster environment. Oracle GoldenGate can be used with any cluster-management solution that is Oracle-certified.

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

You will need to install at least some Oracle GoldenGate objects on shared storage. Select cluster-aware shared storage that is independent of, but available to, all nodes of the cluster.

The best practice is the install Oracle GoldenGate entirely on shared storage. This allows you to start the Oracle GoldenGate processes from any of the nodes without having to make changes to the parameter files. If the active node fails, the processes can be started quickly on another node, using the processing checkpoints that are preserved in the installation directory.

If you decide to install the Oracle GoldenGate binaries and files on each node, rather than on shared storage, the following must be true:



- The Oracle GoldenGate installation must have the same location path on every node
- At minimum, install the following directories on the shared storage to support
 Oracle GoldenGate recovery requirements. On UNIX or Linux, you can create
 symbolic links to them from the installation directory on each node.
 - br
 - dirchk
 - dirdat
 - dirtmp

These directories are among those created when you issue CREATE SUBDIRS during installation.

• The parameter files in the dirprm directory, if not placed on the shared drive, must be identical on all nodes. To resolve environment settings that must be different from one node to the other, you can set environment settings so they are inherited from the local Manager process or reference a node-specific Oracle GoldenGate macro file. Because this scenario can be difficult to enforce, the inherent concerns can be avoided by storing the parameter files on the shared drive.

See also "Integrating Oracle GoldenGate into a Cluster" after you install Oracle GoldenGate.

2.6 Installing Oracle GoldenGate on Linux and UNIX

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

2.6.1 Installing the Oracle GoldenGate Files

To install the Oracle GoldenGate files:

- Extract the Oracle GoldenGate installation file to the system and directory where you want to install Oracle GoldenGate.
- 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

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

CREATE SUBDIRS

6. Issue the following command to exit GGSCI.

EXIT

2.7 Installing Oracle GoldenGate on Windows

Follow these steps to install Oracle GoldenGate for Oracle on a Windows system.



2.7.1 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.
- Ensure that this cluster group is owned by the cluster node that you are logging into.
- 4. Install Oracle GoldenGate according to "Installing the Oracle GoldenGate Files ".

2.7.2 Installing the Oracle GoldenGate Files

To install the Oracle GoldenGate files:

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

```
CREATE SUBDIRS
```

5. Issue the following command to exit GGSCI.

EXIT

2.7.3 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:

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

```
EDIT PARAMS ./GLOBALS
```



Note:

The ./ portion of this command must be used, because the <code>GLOBALS</code> 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 <code>GLOBALS</code>, but without a file extension. Do not move this file. It is used during installation of the Windows service and during data processing.

2.7.4 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.

To install Manager as a Windows service:

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

install option [...]

Where option is one of the following:

Table 2-2 INSTALL Utility 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.



Option	Description
AUTOSTART	Sets the service that is created with ADDSERVICE to start at system boot time. This is the default unless MANUALSTART is used.
MANUALSTART	Sets the service that is created with ADDSERVICE to start manually through GGSCI, a script, or the Services applet of the Control Panel. The default is AUTOSTART.
USER name	Specifies a domain user account that executes Manager. For the $name$, 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.
-	

Table 2-2 (Cont.) INSTALL Utility Options

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.



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 returns a prompt.

2.8 Integrating Oracle GoldenGate into a Cluster

If you installed Oracle GoldenGate in a cluster, take the following steps to integrate Oracle GoldenGate within the cluster solution.

2.8.1 General Requirements in a Cluster

- Register the Oracle GoldenGate Manager process (and only Manager) as a cluster-managed resource as you would any other application. Manager must be the only Oracle GoldenGate process that the cluster-management software starts and stops, because it is the parent process that manages all other processes.
- 2. If the cluster uses a virtual IP address, you may need to obtain an available fixed IP address for the Manager process. The VIP must be an available IP address on the public subnet and cannot be determined through DHCP. In the parameter files of the Extract data pumps, specify the VIP of the remote Manager as the input



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

- value of the RMTHOST parameter. Other Oracle GoldenGate products that access Manager also should use the VIP.
- 3. When you configure Manager, add the AUTOSTART and AUTORESTART parameters so that Manager starts the replication processes automatically. You can, when needed, control Extract, Replicat, and other Oracle GoldenGate processes from within the Oracle GoldenGate user interfaces.
- 4. Mount the shared drive on one node only. This prevents processes from being started on another node. Use the same mount point on all nodes.
- 5. Configure Oracle GoldenGate as directed in this documentation.

2.8.2 Adding Oracle GoldenGate as a Windows Cluster Resource

When installing Oracle GoldenGate in a Windows cluster, follow these instructions to establish Oracle GoldenGate as a cluster resource and configure the Manager service correctly on all nodes.

- In the cluster administrator, add the Manager process to the group that contains the database instance to which Oracle GoldenGate will connect.
- Make sure all nodes on which Oracle GoldenGate will run are selected as possible owners of the resource.
- Make certain the Manager Windows service has the following dependencies (configurable from the Services control panel):
 - The database resource
 - The disk resource that contains the Oracle GoldenGate directory
 - The disk resource that contains the database transaction log files
 - The disk resource that contains the database transaction log backup files



Preparing the System for Oracle GoldenGate

This chapter describes how to prepare the system for Oracle GoldenGate. This chapter contains the following topics:

- Configuring the Transaction Logs for Oracle GoldenGate
- · Preparing Tables for Processing
- Setting the Session Character Set
- Preparing for Initial Extraction
- Specifying the DB2 LUW Database in Parameter Files

3.1 Configuring the Transaction Logs for Oracle GoldenGate

To capture DML operations, Oracle GoldenGate reads the DB2 for LUW online logs by default, but will read the archived logs if an online log is not available. To ensure the continuity and integrity of Oracle GoldenGate processing, configure the logs as follows.

3.1.1 Retaining the Transaction Logs

Configure the database to retain the transaction logs for roll forward recovery by enabling one of the following parameter sets, depending on the database version.

DB2 LUW 9.5 and later:

Set the LOGARCHMETH parameters as follows:

- Set LOGARCHMETH1 to LOGRETAIN.
- Set LOGARCHMETH2 to OFF.

Alternatively, you can use any other LOGARCHMETH options, so long as forward recovery is enabled. For example, the following is valid:

- Set LOGARCHMETH1 to DISK.
- Set LOGARCHMETH2 to TSM.

To Determine Log Retention Parameters:

Connect to the database.

db2 connect to database user username using password

2. Get the database name.

db2 list db directory

3. Get the database configuration for the database.

db2 get db cfg for database

The fields to view are:

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

To Set Log Retention Parameters:

1. Issue one of the following commands.

To enable **USEREXIT**:

db2 update db cfg for database using USEREXIT ON

If not using **USEREXIT**, use this command:

db2 update db cfg for database using LOGRETAIN RECOVERY

To set LOGARCHMETH:

db2 update db cfg for database using LOGARCHMETH1 LOGRETAIN db2 update db cfg for database using LOGARCHMETH2 OFF

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

db2 backup db database to device

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

SQL2061N An attempt to access media "device" is denied.

3.1.2 Specifying the Archive Path

Set the DB2 OVERFLOWLOGPATH parameter to the archive log directory. The node attaches automatically to the path variable that you specify.

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

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

3.2 Preparing Tables for Processing

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

3.2.1 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 <code>emp_src</code> and <code>salary_src</code> and the target tables are <code>emp_targ</code> and <code>salary_targ</code>.

A delete is issued for emp_src.



- 2. It cascades a delete to salary_src.
- 3. Oracle 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.

3.2.2 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.

3.2.2.1 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:

- 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.



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.

3.2.2.2 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*.

3.2.3 Preventing Key Changes

Do not add columns to a key after Oracle GoldenGate starts extracting data from the table. This rule applies to a primary key, a unique key, a KEYCOLS key, or an all-column

key. DB2 LUW does not supply a before image for columns that are added to a table. If any columns in a key are updated on the source, Oracle GoldenGate needs a before image to compare with the current values in the target table when it replicates the update.

3.2.4 Enabling Change Capture

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 by update statements. You can use GGSCI to issue the ALTER TABLE command as follows.

To Enable Change Capture from GGSCI:

- 1. From the Oracle GoldenGate directory, run GGSCI.
- 2. Log on to DB2 from GGSCI as a user that has ALTER TABLE privileges. Specify the data source name with SOURCEDB and specify the user login with USERID and PASSWORD. See Reference for Oracle GoldenGate for Windows and UNIX for encryption options if needed.

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

3. Issue the following command. where <code>owner.table</code> is the fully qualified name of the table. You can use a wildcard to specify multiple table names. Only the asterisk (*) wildcard is supported for DB2 LUW.

```
ADD TRANDATA owner.table
```

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

ALTER TABLE name DATA CAPTURE CHANGES INCLUDE LONGVAR COLUMNS;

Example 3-1 To Exclude LONGVAR Logging:

To omit the include longvar columns clause from the alter table command, use add trandata with the excludelong option.

ADD TRANDATA owner.table, EXCLUDELONG



If LONGVAR columns are excluded from logging, the Oracle GoldenGate features that require before images, such as the <code>getupdatebefores</code>, <code>nocompressupdates</code>, and <code>nocompressdeletes</code> parameters, might return errors if tables contain those columns. For a workaround, see the <code>requirelongdatacapturechanges</code> | <code>norequirelongdatacapturechanges</code> options of the <code>tranlogoptions</code> parameter.

3.2.5 Maintaining Materialized Query Tables

To maintain parity between source and target materialized query tables (MQT), you replicate the base tables, but not the MQTs. The target database maintains the MQTs based on the changes that Replicat applies to the base tables.



The following are the rules for configuring these tables:

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

3.3 Setting the Session Character Set

To support the conversion of character sets between the source and target databases, make certain that the session character set is the same as the database character set. You can set the session character set with the DB2CODEPAGE environment variable.

3.4 Preparing for Initial Extraction

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

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

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

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

To ensure initial transactional consistency:

To avoid the capture of partial transactions, initialize the Extract process at a point in time when the database is in a quiesced state. DB2 provides a QUIESCE command for such a purpose. This is the only way to ensure transactional consistency.



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

To view open transactions:

IBM provides a utility called db2pd for monitoring DB2 databases and instances. You can use it to view information about open transactions and to determine if any of them



span the start point. However, because DB2 LUW log records lack timestamps, it might not be possible to make an accurate assessment. If possible, quiesce the database prior to initialization of Oracle GoldenGate.

For more information on initializing the Oracle GoldenGate environment, see *Administering Oracle GoldenGate for Windows and UNIX*.

3.5 Specifying the DB2 LUW Database in Parameter Files

For an Oracle GoldenGate process to connect to the correct DB2 LUW database, you must specify the name (not an alias) of the DB2 LUW database with the following parameters:

- Specify the DB2 source database with the Extract parameter SOURCEDB.
- Specify the DB2 target database name with the Replicat parameter TARGETDB.

For more information about these parameters, see the Reference for Oracle GoldenGate for Windows and UNIX.



4

Configuring Oracle GoldenGate for DB2

This chapter provides an overview of the basic steps required to configure Oracle GoldenGate for a DB2 LUW source and target database.

This chapter contains the following topics:

- What to Expect from these Instructions
- Where to Get More Information
- Configuring the Primary Extract
- Configuring the Data Pump Extract
- · Configuring Replicat
- Next Steps in the Deployment
- When to Start Replicating Transactional Changes
- Testing Your Configuration

4.1 What to Expect from these Instructions

These instructions show you how to configure basic parameter (configuration) files for the following processes:

- A primary Extract (captures transaction data from the data source)
- A data-pump Extract (propagates the data from local storage to the target system)
- A Replicat (applies replicated data to the target database)

Your business requirements probably will require a more complex topology, but this procedure forms a basis for the rest of your configuration steps.

By performing these steps, you can:

- Get the basic configuration files established.
- Build upon them later by adding more parameters as you make decisions about features or requirements that apply to your environment.
- Use copies of them to make the creation of additional parameter files faster than starting from scratch.

4.2 Where to Get More Information

See Administering Oracle GoldenGate for Windows and UNIX for more information about:

- The processes and files that you are configuring
- Detailed configuration information



- Security options
- Data-integration options (filtering, mapping, conversion)
- Instructions for configuring complex topologies
- Steps to perform initial instantiation of the replication environment
- Administrative topics

4.3 Configuring the Primary Extract

These steps configure the primary Extract to capture transaction data from a source DB2 LUW database and write the data to a local trail for temporary storage.

1. In GGSCI on the source system, create the Extract parameter file.

EDIT PARAMS name

Where: name is the name of the primary Extract.

2. Enter the Extract parameters in the order shown, starting a new line for each parameter statement. See Table 4-1 for more information and parameter descriptions.

Basic parameters for the primary Extract

EXTRACT finance
SOURCEDB mysource, USERIDALIAS myalias
LOGALLSUPCOLS
ENCRYPTTRAIL AES192
EXTTRAIL /ggs/dirdat/lt
TABLE hr.*;

Table 4-1 Basic Parameters for Primary Extract

Parameter	Description	
EXTRACT group	group is the name of the Extract group. For more information, see Reference for Oracle GoldenGate for Windows and UNIX.	
SOURCEDB database, USERIDALIAS alias	Specifies the real name of the source DB2 LUW database (not an alias), plus the alias of the database login credential of the user that is assigned to Extract. This credential must exist in the Oracle GoldenGate credential store. For more information, see Database User for Oracle GoldenGate Processes.	
ENCRYPTTRAIL algorithm	Encrypts the local trail. For more information about Oracle GoldenGate trail encryption options, see <i>Administering Oracle GoldenGate for Windows and UNIX</i> .	
EXTTRAIL pathname	Specifies the path name of the local trail to which the primary Extract writes captured data for temporary storage.	



Table 4-1 (Cont.) Basic Parameters for Primary Extract

Parameter	Description
TABLE schema.object;	 Specifies the database object for which to capture data. TABLE is a required keyword. schema is the schema name or a wildcarded set of schemas. object is the table name, or a wildcarded set of tables. See Administering Oracle GoldenGate for Windows and UNIX for information about how to specify object names with and without wildcards. Note that only the asterisk (*) wildcard is supported for DB2 LUW. The question mark (?) wildcard is not supported for this database.
	Terminate the parameter statement with a semi-colon.
	To exclude tables from a wildcard specification, use the TABLEEXCLUDE parameter. See Reference for Oracle GoldenGate for Windows and UNIX for more information about usage and syntax.
	For more information and for additional TABLE options that control data filtering, mapping, and manipulation, see Reference for Oracle GoldenGate for Windows and UNIX.

- 3. Enter any optional Extract parameters that are recommended for your configuration. You can edit this file at any point before starting processing by using the EDIT PARAMS command in GGSCI. For a list of parameters and links to their detailed reference, see *Reference for Oracle GoldenGate for Windows and UNIX*.
- Save and close the file.

4.4 Configuring the Data Pump Extract

These steps configure the data pump that reads the local trail and sends the data across the network to a remote trail on the target. The data pump is optional, but recommended.

1. In GGSCI on the source system, create the data-pump parameter file.

```
EDIT PARAMS name
```

Where name is the name of the data-pump Extract.

 Enter the data-pump Extract parameters in the order shown, starting a new line for each parameter statement. Your input variables will be different. See Table 4-2 for descriptions.

Basic parameters for the data-pump Extract group:

```
EXTRACT extpump

SOURCEDB mypump, USERIDALIAS myalias

RMTHOST fin1, MGRPORT 7809 ENCRYPT AES192, KEYNAME securekey2

RMTTRAIL /ggs/dirdat/rt

TABLE hr.*;
```

Table 4-2 Basic Parameters for a Data-pump Extract

Parameter	Description
EXTRACT group	group is the name of the data pump Extract. For more information, see Reference for Oracle GoldenGate for Windows and UNIX.



Table 4-2 (Cont.) Basic Parameters for a Data-pump Extract

Parameter	Description
SOURCEDB database, USERIDALIAS alias	Specifies the real name of the source DB2 LUW database (not an alias), plus the alias of the database login credential of the user that is assigned to Extract. This credential must exist in the Oracle GoldenGate credential store. For more information, see Database User for Oracle GoldenGate Processes.
RMTHOST hostname, MGRPORT portnumber, [, ENCRYPT algorithm KEYNAME keyname]	 RMTHOST specifies the name or IP address of the target system. MGRPORT specifies the port number where Manager is running on the target. ENCRYPT specifies optional encryption of data across TCP/IP. For additional options and encryption details, see <i>Reference for Oracle GoldenGate for Windows and UNIX</i>.
RMTTRAIL pathname	Specifies the path name of the remote trail. For more information, see <i>Reference for Oracle GoldenGate for Windows and UNIX</i> .
TABLE schema.object;	Specifies a table or sequence, or multiple objects specified with a wildcard. In most cases, this listing will be the same as that in the primary Extract parameter file. Table is a required keyword. schema is the schema name or a wildcarded set of schemas. object is the name of a table or a wildcarded set of tables. See Administering Oracle GoldenGate for Windows and UNIX for information about how to specify object names with and without wildcards. Note that only the asterisk (*) wildcard is supported for DB2 LUW. The question mark (?) wildcard is not supported for this database. Terminate the parameter statement with a semi-colon. To exclude tables from a wildcard specification, use the TableExclude parameter. See Reference for Oracle GoldenGate for Windows and UNIX for more information about usage and syntax. For more information and for additional Table options that control data filtering, mapping, and manipulation, see Reference for Oracle GoldenGate for Windows and UNIX.

- 3. Enter any optional Extract parameters that are recommended for your configuration. You can edit this file at any point before starting processing by using the EDIT PARAMS command in GGSCI. For a list of parameters and links to their detailed reference, see Reference for Oracle GoldenGate for Windows and UNIX.
- 4. Save and close the file.

4.5 Configuring Replicat

These steps configure a Replicat process on a DB2 LUW target system. If you are configuring Replicat for a database of a different type, see the installation and configuration documentation for that database.

4.5.1 Creating a Temporal Table

A temporal table is a table that maintains the history of its data and the time period when its data are valid. Temporal tables are used in Oracle GoldenGate to keep track of all the old rows that are deleted or updated in the table. Temporal tables are also used to maintain the business validity of its rows and data. For example, Oracle GoldenGate keeps track of the time period during which a row is valid. There are three types of temporal tables, system-period, application-period, and bitemporal table.



4.5.1.1 Support for Temporal Tables

- Replication between system-period temporal tables and application-period temporal tables is not supported.
- Replication from a non-temporal table to a temporal table is not supported.
- Replication of temporal tables with the INSERTALLRECORDS parameter is not supported.
- Bidirectional replication is supported only with the default replication.
- CDR in bidirectional replication is not supported.
- CDR in application-period temporal tables is supported.

4.5.1.2 Replicating with Temporal Tables

You can choose one of the following methods to replicate a system-period or a bitemporal temporal table as follows:

- You can replicate a temporal table to another temporal table only; this is the default behavior. Oracle GoldenGate will not replicate the SYSTEM_TIME period and transaction id columns because these are automatically generated columns at the apply side. The database manager populates the columns in the target temporal table using the system clock time and with the default values. You can preserve the original values these columns then use any of the following:
 - Add extra timestamp columns in the target temporal table and map the columns accordingly. The extra columns are automatically added to the associated history table.
 - Use a non-temporal table at the apply side and map the columns appropriately. In this scenario, you will not be able to maintain the history table.
 - In a heterogeneous configuration where the source is DB2 LUW and the target is a different database, you can either ignore the automatically generated columns or use an appropriate column conversion function to convert the columns value in the format that target database supports and map them to target columns accordingly.

Or

You can replicate a temporal table, with the associated history table, to a temporal and history table respectively then you must specify the replicate parameter, DBOPTIONS SUPPRESSTEMPORALUPDATES. You must specify both the temporal table and history table to be captured in the Extract parameter file. Oracle GoldenGate replicates the SYSTEM_TIME period and transactions id columns value. You must ensure that the database instance has the execute permission to run the stored procedure at the apply side.

Oracle GoldenGate cannot detect and resolve conflicts while using default replication as SYSTEM_TIME period and transactionstart id columns remains auto generated. These columns cannot be specified in set and where clause. If you use the SUPPRESSTEMPORALUPDATES parameter, then Oracle GoldenGate supports CDR.



4.5.1.3 Converting

You can convert an already existing table into a temporal table, which changes the structure of the table. This section describes how the structure of the tables changes. The following sample existing table is converted into all three temporal tables types in the examples in this section:

Example 1 Converting an existing table into System-period temporal table.

You convert the sample existing table into a system-period temporal table by adding SYSTEM_PERIOD, transaction id columns, and SYSTEM_TIME period as in the following:

```
ALTER TABLE policy_info
ADD COLUMN sys_start TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN;
ALTER TABLE policy_info
ADD COLUMN sys_end TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END;
ALTER TABLE policy_info
ADD COLUMN ts_id TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS TRANSACTION START ID;
ALTER TABLE policy_info ADD PERIOD SYSTEM_TIME(sys_start, sys_end);
```

Then you create a history table for the new temporal table using one of the following two methods:

CREATE TABLE hist_policy_info LIKE policy_info with RESTRICT ON DROP;

The RESTRICT ON DROP clause will not allow the history table to get dropped while dropping system-period temporal table. Otherwise the history table gets implicitly dropped while dropping its associated temporal table. You can create a history table without RESTRICT ON DROP. A history table cannot be explicitly dropped.

You should not use the <code>GENERATED</code> ALWAYS clause while creating a history table. The primary key of the system-period temporal table also does not apply here as there could be many updates for a particular row in the base table, which triggers many inserts into the history table for the same set of primary keys. Apart from these,



the structure of a history table should be exactly same as its associated systemperiod temporal table. The history table must have the same number and order of columns as system-period temporal table. History table columns cannot explicitly be added, dropped, or changed. You must associate a system-period temporal table with its history table with the following statement:

```
ALTER TABLE policy_info ADD VERSIONING USE HISTORY TABLE hist_policy_info.
```

The GENERATED ALWAYS columns of the table are the ones that are always populated by the database manager so you do not have any control over these columns. The database manager populates these columns based on the system time.

The extra added SYSTEM_PERIOD and transaction id columns will have default values for already existing rows as in the following:

The associated history table is populated with the before images once you start updating the temporal table.

Example 2 Converting an existing table into application-period temporal table. You can convert the sample existing table into application-period temporal table by adding time columns and a BUSINESS_TIME period as in the following:

```
ALTER TABLE policy_info ADD COLUMN bus_start DATE NOT NULL DEFAULT '10/10/2001'"
ALTER TABLE policy_info ADD COLUMN bus_end DATE NOT NULL DEFAULT '10/10/2002'
ALTER TABLE policy_info ADD PERIOD BUSINESS_TIME(bus_start, bus_end)
```

While adding time columns, you need to make sure that while entering business validity time values of the existing time columns, the <code>bus_start</code> column always has value lesser than <code>bus_end</code> because these columns specify the business validity of the rows.

The new application-period temporal table will look similar to:

POLICY_ID	COVERAGE	BUS_START	BUS_END	
ERT	14000		10/10/2001	10/10/2002
DEF	13000		10/10/2001	10/10/2002
ABC	12000		10/10/2001	10/10/2002

Example 3 Converting an existing table into bitemporal table.

You can convert the sample existing table into bitemporal table by adding System_period, time columns along with the system_time and business_time period as in the following:

```
ALTER TABLE policy_info
ADD COLUMN sys_start TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN;
ALTER TABLE policy info
```



```
ADD COLUMN sys_end TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END;

ALTER TABLE policy_info
   ADD COLUMN ts_id TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS TRANSACTION START ID;

ALTER TABLE policy_info ADD PERIOD SYSTEM_TIME(sys_start, sys_end);

ALTER TABLE policy_info ADD COLUMN bus_start DATE NOT NULL DEFAULT '10/10/2001'"

ALTER TABLE policy_info ADD COLUMN bus_end DATE NOT NULL DEFAULT '10/10/2002'

ALTER TABLE policy_info ADD PERIOD BUSINESS_TIME(bus_start, bus_end)
```

While adding the time columns, you must make sure that while entering business validity time values of already existing time columns, the <code>bus_start</code> column always has value lesser than <code>bus_end</code> because these columns specify the business validity of the rows.

Then you create a history table for the new temporal table using one of the following two methods:

 The RESTRICT ON DROP clause will not allow the history table to get dropped while dropping system-period temporal table. Otherwise the history table gets implicitly dropped while dropping its associated temporal table. You can create a history table without RESTRICT ON DROP. A history table cannot be explicitly dropped.

You should not use the GENERATED ALWAYS clause while creating a history table. The primary key of the system-period temporal table also does not apply here as there could be many updates for a particular row in the base table, which triggers many inserts into the history table for the same set of primary keys. Apart from these, the structure of a history table should be exactly same as its associated system-period temporal table. The history table must have the same number and order of columns as system-period temporal table. History table columns cannot explicitly be added, dropped, or changed. You must associate a system-period temporal table with its history table with the following statement:

```
ALTER TABLE policy_info ADD VERSIONING USE HISTORY TABLE hist_policy_info.
```

The GENERATED ALWAYS columns of the table are the ones that are always populated by the database manager so you do not have any control over these columns. The database manager populates these columns based on the system time.

The extra added SYSTEM_PERIOD and transaction id columns will have default values for already existing rows as in the following:



The associated history table is populated with the before images once you start updating the temporal table.

The extra added SYSTEM_TIME period, transaction id and time columns will have default values for already existing rows as in the following:

The history table is populated with the before images once user starts updating the temporal table.

Example 4 Replication in Heterogeneous Environment.

In heterogeneous configuration in which you do not have temporal tables at the apply side, you can only replicate the system-period and bitemporal tables though *not* the associated history tables. While performing replication in this situation, you must take care of the SYSTEM_PERIOD and transaction id columns value. These columns will have some values that the target database might not support. You should first use the map conversion functions to convert these values into the format that the target database supports, and then map the columns accordingly.

For example, MySQL has a DATETIME range from 1000-01-01 00:00:00.000000 to 9999-12-31 23:59:59.999999. You cannot replicate a timestamp value of 0001-01-01-00.00.00.000000000000 to MySQL. To replicate such values, you must convert this value into the MySQL DATETIME value 1000-01-01 00:00:00.000000, and then map the columns. If you have the following row in the policy_info system-period table:

To replicate the row into MySQL, you would use the colmap() function:



```
map source_schema.policy_info, target target_schema.policy_info colmap
(policy_id=policy_id, coverage=coverage, sys_start= @IF( ( @NUMSTR( @STREXT(sys_start,1,4))) > 1000, sys_start, '1000-01-01 00.00.00.000000'), sys_end=sys_end,
    ts_id= @IF( ( @NUMSTR( @STREXT(ts_id,1,4))) > 1000, ts_id, '1000-01-01
    00.00.0000000'));
```

4.5.2 Creating a Checkpoint Table

The checkpoint table is a required component of Replicat.

Replicat maintains its recovery checkpoints in the checkpoint table, which is stored in the target database. Checkpoints are written to the checkpoint table within the Replicat transaction. Because a checkpoint either succeeds or fails with the transaction, Replicat ensures that a transaction is only applied once, even if there is a failure of the process or the database.

To configure a checkpoint table, see *Administering Oracle GoldenGate for Windows* and UNIX.

4.5.3 Configuring the Replicat Parameter File

These steps configure the Replicat process. This process applies replicated data to a DB2 LUW target database.

1. In GGSCI on the target system, create the Replicat parameter file.

```
EDIT PARAMS name
```

Where: name is the name of the Replicat group.

2. Enter the Replicat parameters in the order shown, starting a new line for each parameter statement. See Table 4-3 for descriptions.

Basic parameters for the Replicat group:

```
REPLICAT financer
TARGETDB mytarget, USERIDALIAS myalias
ASSUMETARGETDEFS
MAP hr.*, TARGET hr2.*;
```

Table 4-3 Basic Parameters for Replicat

Parameter	Description
REPLICAT group	group is the name of the Replicat group.
TARGETDB database, USERIDALIAS alias	Specifies the real name of the target DB2 LUW database (not an alias), plus the alias of the database login credential of the user that is assigned to Replicat. This credential must exist in the Oracle GoldenGate credential store. For more information, see Database User for Oracle GoldenGate Processes.



Table 4-3 (Cont.) Basic Parameters for Replicat

Parameter	Description
ASSUMETARGETDEFS	Specifies how to interpret data definitions. ASSUMETARGETDEFS assumes the source and target tables have identical definitions. (This procedure assume identical definitions.) Use the alternative SOURCEDEFS if the source and target tables have different definitions, and create a source data-definitions file with the DEFGEN utility. For more information about data definitions, see <i>Administering Oracle GoldenGate for Windows and UNIX</i> .
MAP schema.object, TARGET schema.object;	 Specifies the relationship between a source table or multiple objects, and the corresponding target object or objects. MAP specifies the source portion of the MAP statement and is a required keyword. Specify the source objects in this clause. TARGET specifies the target portion of the MAP statement and is a required keyword. Specify the target objects to which you are mapping the source objects. schema is the schema name or a wildcarded set of schemas. object is the name of a table or a wildcarded set of objects. Terminate this parameter statement with a semi-colon. Note that only the asterisk (*) wildcard is supported for DB2 LUW. The question mark (?) wildcard is not supported for this database. To exclude objects from a wildcard specification, use the MAPEXCLUDE parameter. For more information and for additional options that control data filtering, mapping, and manipulation, see MAP in Reference for Oracle GoldenGate for Windows and UNIX.

- 3. Enter any optional Replicat parameters that are recommended for your configuration. You can edit this file at any point before starting processing by using the EDIT PARAMS command in GGSCI. For a list of parameters and links to their detailed reference, see *Reference for Oracle GoldenGate for Windows and UNIX*.
- 4. Save and close the file.

4.6 Next Steps in the Deployment

Because of its flexibility, Oracle GoldenGate offers numerous features and options that must be considered before you start any processes. To further configure Oracle GoldenGate to suit your business needs, see the following:

- For additional configuration guidelines to achieve specific replication topologies, see Administering Oracle GoldenGate for Windows and UNIX. This guide also contains information about:
 - Oracle GoldenGate architecture
 - Oracle GoldenGate commands
 - Oracle GoldenGate initial load methods
 - Configuring security
 - Using customization features
 - Mapping columns that contain dissimilar data



- Data filtering and manipulation
- For syntax options and descriptions of Oracle GoldenGate GGSCI commands and Oracle GoldenGate parameters shown in this guide, see Reference for Oracle GoldenGate for Windows and UNIX.

4.7 When to Start Replicating Transactional Changes

You must start replication when the source and target data is in a synchronized state, where the corresponding rows in the source and target tables contain identical data values. Unless you are starting with brand new source and target databases with no current user activity, you will need to activate change capture and apply processes to handle ongoing transactional changes while an initial load is being applied to the target. This process is known as *initial synchronization*, or also as *instantiation*. The initial load captures a point-in-time snapshot of the source data and applies it to the target, while Oracle GoldenGate maintains any changes that are made after that point.

See Administering Oracle GoldenGate for Windows and UNIX for instantiation options.

4.8 Testing Your Configuration

It is important to test your configuration in a test environment before deploying it live on your production machines. This is especially important in an active-active or high availability configuration, where trusted source data may be touched by the replication processes. Testing enables you to find and resolve any configuration mistakes or data issues without the need to interrupt user activity for re-loads on the target or other troubleshooting activities.



5

Uninstalling Oracle GoldenGate

This chapter describes how to uninstall Oracle GoldenGate. This chapter contains the following sections:

- Stopping Processes
- Removing the Oracle GoldenGate Database Setup
- Removing Oracle GoldenGate Windows Components
- · Manually Removing the Oracle GoldenGate Files

5.1 Stopping Processes

This procedure stops the Extract and Replication processes. Leave Manager running until directed to stop it.

On all Systems:

- Run the command shell.
- 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.

```
STOP ER *
```

6. Stop the Manager process.

STOP MANAGER

5.2 Removing the Oracle GoldenGate Database Setup

Follow these instructions to remove Oracle GoldenGate objects and attribute settings that are configured within the database.

On a Source System:

- Log on as the system administrator or as a user with permission to issue Oracle GoldenGate commands and delete files and directories from the operating system.
- 2. Run GGSCI from the Oracle GoldenGate directory.
- 3. Stop all Oracle GoldenGate processes.

STOP ER *

4. Stop the Manager process.

STOP MANAGER

5. In GGSCI, log into the database with the DBLOGIN command.

DBLOGIN SOURCEDB database, USERIDALIAS alias

In GGSCI, alter the tables that were in the Oracle GoldenGate configuration to DATA CAPTURE NONE.

```
DELETE TRANDATA schema.table [NOSCHEDULINGCOLS | ALLCOLS]
```

On a Target System:

1. Stop Replicat.

```
STOP REPLICAT group
```

2. Log into the database with the DBLOGIN command.

```
DBLOGIN SOURCEDB database, USERIDALIAS alias
```

3. Delete the Replicat group.

```
DELETE REPLICAT group
```

Delete the Replicat checkpoint table.

```
DELETE CHECKPOINTTABLE schema.table
```

5.3 Removing Oracle GoldenGate Windows Components

(Valid for Windows installations) This procedure does the following:

- Removes Oracle GoldenGate as a Windows cluster resource from a source or target Windows system.
- Stops Oracle GoldenGate events from being reported to the Windows Event Manager.
- Removes the Manager service.

Perform these steps on source and target systems.

- 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. Run GGSCI and make certain that all Extract and Replicat processes are stopped. Stop any that are running.

```
STATUS ER *
STOP ER *
```

- 3. (Cluster) Use the Cluster Administrator tool to take the Manager resource offline.
- 4. (Cluster) Right click the resource and select **Delete** to remove it.
- Click Start then Run, and then type cmd in the Run dialog box to open the command console.
- 6. Change directories to the Oracle GoldenGate installation directory.
- Run the INSTALL utility with the following syntax.

```
install deleteevents deleteservice
```

8. (Cluster) Move the cluster group to the next node in the cluster, and repeat from step 5.



5.4 Manually Removing the Oracle GoldenGate Files

Perform these steps on all systems to remove the Oracle GoldenGate installation directory.

1. In GGSCI, verify that all processes are stopped. Stop any that are running.

```
STATUS MANAGER
STATUS ER *
STOP MANAGER
STOP ER *
```

2. Exit GGSCI.

EXIT

3. Remove the Oracle GoldenGate installation directory.



A

Oracle GoldenGate Installed Components

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

This appendix contains the following sections:

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

A.1 Oracle GoldenGate Programs and Utilities

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



Some programs may not exist in all installations. For example, if only capture or delivery is supported by Oracle GoldenGatee 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.
ggmxinstall	Oracle GoldenGate installation script for the SQL/MX database.



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

Program	Description
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.



Some directories may not exist in all installations.

Table A-2 Oracle GoldenGate Installed Subdirectories

Directory	Description
br	Contains the checkpoint files for the bounded recover feature.



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

ctory	Description
	Contains the property and XML files that are used to configure Oracle GoldenGate Monitor.
lb	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.
chk	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 <code>group_name+sequence_number.ext</code> where <code>sequence_number</code> is a sequential number appended to aged files and <code>ext</code> is either <code>cpe</code> for Extract checkpoint files or <code>cpr</code> for Replicat checkpoint files.
	Do not edit these files.
	Examples:
	ext1.cpe
	repl.cpr
erd	Contains credential store files.
lat	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
lef	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
lmp	Contains trace, or dump, files that support the internal activity logging mechanism.
iar	Contains the Java executable files that support Oracle GoldenGate Monitor.
	logging mechanism. Contains the Java executable files that support Oracle



Table A-2 (Cont.) Oracle GoldenGate Installed 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.sq	Empty script file for use by Oracle GoldenGate support staff.
ddl_oral1.sql ddl_oral2.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	Supports the installation of the Oracle DDL objects. This script
ddl_session1.sql	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)



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

Component	Description
ddl_staymetadata_off.sql	Scripts that control whether the Oracle DDL trigger collects
ddl_staymetadata_on.sql	metadata. This script runs programmatically; do not run it manually.
ddl_trace_off.sql	Scripts that control whether DDL tracing is on or off.
ddl_trace_on.sql	
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	Scripts that create and populate demonstration tables for use with tutorials and basic testing.
demo_more_db_scriptname.s	
.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	Windows shared libraries for International Components for
icuinxx.dll	Unicode, where $\mathbf{x}\mathbf{x}$ is the currently used version.
icuucxx.dll	
jagent.bat	Windows batch file for the Java Agent for Oracle GoldenGate Monitor.
jagent.log	Log files for the Oracle GoldenGate Monitor Agent.
jagentjni.log	
jagent.sh	UNIX shell script for the Java Agent for Oracle GoldenGate Monitor
LGPL.txt	Lesser General Public License statement. Applies to free libraries from the Free Software Foundation.



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

Component	Description
libodbc.so	ODBC file for Ingres 2.6 on Unix.
libodbc.txt	License agreement for libodbc.so.
libxml2.dll	Windows dynamic link library containing the XML library for the Oracle GoldenGate XML procedures.
libxml2.txt	License agreement for libxml2.dll.
marker.hist	File created by Replicat if markers were passed from a NonStop source system.
marker_remove.sql	Script that removes the DDL marker table. (Oracle installations)
marker_setup.sql	Script that installs the Oracle GoldenGate DDL marker table. (Oracle installations)
marker_status.sql	Script that confirms successful installation of the DDL marker table. (Oracle installations)
notices.txt	Third-party software license file.
odbcinst.ini	Ingres 2.6 on Unix ODBC configuration file.
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	Script files that support the Oracle GoldenGate Monitor Agent.
pw_agent_util.sh	
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	z/OS paramlib members to start and stop the Manager process.
stop.prm	
startmgr	z/OS Unix System Services scripts to start the Manager process
stopmgr	from GGSCI.
startmgrcom	z/OS system input command for the Manager process.
stopmgrcom	
tcperrs	File containing user-defined instructions for responding to TCP/IP errors.
usrdecs.h	Include file for user exit API.
	Apache XML parser library.
xerces-c_2_8.dll	Apache Aivil parser library.



A.4 Oracle GoldenGate Checkpoint Table



Index

C	installing on <i>(continued)</i>
cluster installing on, 2-6, 2-9	 L
cluster, running in, 2-4	Linux, installing on, 2-4
<u> </u>	U
installing on Linux and UNIX, 2-4	UNIX, installing on, 2-4

