This guide explains how to install, run, and administer the Oracle GoldenGate Veridata data comparison solution.
# Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>vii</td>
</tr>
<tr>
<td>1</td>
<td>About Oracle GoldenGate Veridata</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata Architecture</td>
</tr>
<tr>
<td></td>
<td>Comparing Data with Oracle GoldenGate Veridata</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata Comparison Objects</td>
</tr>
<tr>
<td></td>
<td>Satisfying Uniqueness Requirements</td>
</tr>
<tr>
<td></td>
<td>How Oracle GoldenGate Veridata Compares Data</td>
</tr>
<tr>
<td></td>
<td>Viewing Comparison Results</td>
</tr>
<tr>
<td></td>
<td>Out-of-Sync Report</td>
</tr>
<tr>
<td></td>
<td>Comparison Report</td>
</tr>
<tr>
<td>2</td>
<td>Oracle GoldenGate Veridata System Requirements</td>
</tr>
<tr>
<td></td>
<td>General Information</td>
</tr>
<tr>
<td></td>
<td>Supported Databases</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata Agent System Requirements</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata Java agent</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata C-Agent</td>
</tr>
<tr>
<td></td>
<td>Disk and Memory Requirements for the Agent Component</td>
</tr>
<tr>
<td></td>
<td>Database Privileges for the Agent Component</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata Server System Requirements</td>
</tr>
<tr>
<td></td>
<td>Location for the Server Component</td>
</tr>
<tr>
<td></td>
<td>Disk and Memory Requirements for the Server Component</td>
</tr>
<tr>
<td></td>
<td>Repository Requirements for the Server Component</td>
</tr>
<tr>
<td></td>
<td>Oracle GoldenGate Veridata Web User Interface Requirements</td>
</tr>
<tr>
<td></td>
<td>Location for the Web Component</td>
</tr>
<tr>
<td></td>
<td>Java Environment for the Web User Interface</td>
</tr>
<tr>
<td></td>
<td>Supported Browsers for the Web User Interface</td>
</tr>
<tr>
<td></td>
<td>Recommended Screen Resolution for the Web User Interface</td>
</tr>
<tr>
<td></td>
<td>Port Numbers for the Web User Interface</td>
</tr>
<tr>
<td></td>
<td>Security for the Web User Interface</td>
</tr>
<tr>
<td></td>
<td>SQL/MP Data Exclusions</td>
</tr>
<tr>
<td>3</td>
<td>Installing Oracle GoldenGate Veridata Java Agent</td>
</tr>
<tr>
<td></td>
<td>Installation Overview</td>
</tr>
</tbody>
</table>
Installing the Java Agent on a UNIX or Linux System ................................................................. 3-1
Installing the Java Agent on a Windows System ......................................................................... 3-2

4 Installing Oracle GoldenGate Veridata C-Agent

Installation Overview ............................................................................................................... 4-1
Installing the C-agent on a UNIX or Linux System ............................................................... 4-1
Installing the C-Agent on a Windows System ....................................................................... 4-2
  Installing the Oracle GoldenGate Veridata Agent Files ..................................................... 4-2
  Specifying a Custom Manager Name .................................................................................. 4-3
  Installing Manager as a Windows service ........................................................................... 4-3
  Configuring Manager ......................................................................................................... 4-4
Installing the C-Agent on a NonStop System ....................................................................... 4-5
  Installing the Oracle GoldenGate Veridata Agent Files ..................................................... 4-5
  Copying VSNSERV to Remote Nodes .................................................................................. 4-6
  Creating a GLOBALS File .................................................................................................. 4-7
  Configuring Manager ......................................................................................................... 4-7

5 Installing Using a MySQL Repository

Installation Prerequisites ...................................................................................................... 5-1
  Verify the Installed MySQL Database ................................................................................ 5-1
  Verify the Database Character Encoding .......................................................................... 5-2
Installing the Oracle GoldenGate Veridata Software ............................................................ 5-2

6 Installing Using an Oracle Repository

Preinstallation System Check .................................................................................................. 6-1
Installing the Oracle GoldenGate Veridata Software ............................................................ 6-2

7 Installing Using a SQL Server Repository

Installing the Oracle GoldenGate Veridata Software ............................................................ 7-1

8 Configuring Oracle GoldenGate Veridata Security

Oracle GoldenGate Veridata Security Overview ................................................................... 8-1
Securing the Oracle GoldenGate Veridata Files ................................................................... 8-1
  Controlling Access to the Installation Directories ............................................................ 8-1
  Securing files that Contain Business Data ......................................................................... 8-1
Securing access to Oracle GoldenGate Veridata Web .......................................................... 8-2
Maintaining Passwords .......................................................................................................... 8-3
  Changing User Passwords .................................................................................................. 8-4
  Changing the Repository Password .................................................................................... 8-4

9 Running the Oracle GoldenGate Veridata Programs

Starting and Stopping the C-agent and Manager ................................................................. 9-1
Starting and Stopping the Java-Based Components ............................................................. 9-2
  Controlling the Java-Based Components from the Command Line .................................. 9-2
  Controlling the Java-Based Components as Windows Services ..................................... 9-3
10 Running Comparisons from the Command Line

Overview of the Command Line Interface ................................................................. 10-1
Running vericom ..................................................................................................... 10-2
Vericom exit statuses ......................................................................................... 10-6
Vericom Output Examples .................................................................................. 10-6

11 Oracle GoldenGate Veridata Server Configuration Parameters

Overview of the Server Memory ............................................................................ 11-1
Estimating Memory Usage .................................................................................... 11-2
How to Set a Parameter ....................................................................................... 11-2
Parameter Descriptions ....................................................................................... 11-3
  server.max_sort_memory ..................................................................................... 11-3
  server.max_concurrent_comparison_threads ..................................................... 11-3
  server.max_comparison_sort_memory ................................................................. 11-3
  server.sort_waiting_threshold ......................................................................... 11-4

12 Upgrading Oracle GoldenGate Veridata

Upgrading the Server and Web User Interface ..................................................... 12-1
  Upgrade Support Overview ............................................................................... 12-1
  Server and Web Upgrade Instructions for All Platforms .................................. 12-2
Upgrading to a New Release of the Agent .......................................................... 12-2
  Upgrading a Java Agent to a Newer Release .................................................. 12-3
  Upgrading from a C-Agent to a Java Agent ...................................................... 12-4
  Upgrading a C-Agent to a Newer Release ....................................................... 12-4

13 Uninstalling Oracle GoldenGate Veridata

Uninstalling the Server and Web Components (all platforms) ............................ 13-1
Uninstalling the Java Agent from a Windows System .......................................... 13-2
Uninstalling the Java Agent from a UNIX or Linux System ................................. 13-2
Uninstalling the C-agent (all platforms) ............................................................. 13-2

A Downloading Oracle GoldenGate Veridata

B Changing Oracle GoldenGate Veridata Web Port Numbers

C Configuring an ODBC connection for a SQL Server Repository

Glossary

Index
This document describes how to install, deploy, and administer Oracle GoldenGate Veridata.

**Audience**

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

**Documentation Accessibility**

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

**Access to Oracle Support**

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

**Related Documents**

The complete Oracle GoldenGate documentation set includes the following components:

**HP NonStop Guardian Platform**
- Oracle GoldenGate for Mainframe Reference Guide
- Oracle GoldenGate for Mainframe Administrator’s Guide
- Oracle GoldenGate Application Adapters BASE24 Administrator’s Guide
- Oracle GoldenGate Application Adapters BASE24 Supplemental Guides for D24, T24, and N24

**Windows, UNIX, and Linux Platforms**
- Oracle GoldenGate Installation and Setup Guides per supported database
- Oracle GoldenGate Windows and UNIX Administrator’s Guide
- Oracle GoldenGate Windows and UNIX Reference Guide
Oracle GoldenGate Windows and UNIX Troubleshooting and Tuning Guide

Oracle GoldenGate Upgrade Guide

Other Oracle GoldenGate Products

Oracle GoldenGate Adapters Administrator’s Guide for Flat Files
Oracle GoldenGate Adapters Administrator’s Guide for Java
Oracle GoldenGate Director Administrator’s Guide
Oracle GoldenGate Monitor Administrator’s Guide
Oracle GoldenGate Veridata Administrator’s Guide

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>boldface</td>
<td>Boldface type indicates graphical user interface elements associated with an action, such as &quot;From the File menu, select Save.&quot; Boldface also is used for terms defined in text or in the glossary.</td>
</tr>
<tr>
<td>italic, Italic</td>
<td>Italic type indicates placeholder variables for which you supply particular values, such as in the parameter statement: TABLE table_name. Italic type also is used for book titles and emphasis.</td>
</tr>
<tr>
<td>MONOSPACE, monospace</td>
<td>Monospace type indicates code components such as user exits and scripts; the names of files and database objects; URL paths; and input and output text that appears on the screen. Uppercase monospace type is generally used to represent the names of Oracle GoldenGate parameters, commands, and user-configurable functions, as well as SQL commands and keywords.</td>
</tr>
<tr>
<td>UPPERCASE</td>
<td>Uppercase in the regular text font indicates the name of a utility unless the name is intended to be a specific case.</td>
</tr>
<tr>
<td>{}</td>
<td>Braces within syntax enclose a set of options that are separated by pipe symbols, one of which must be selected, for example: (option1</td>
</tr>
<tr>
<td>[]</td>
<td>Brackets within syntax indicate an optional element. For example in this syntax, the SAVE clause is optional: CLEANUP REPLICA t group_name [, SAVE count]. Multiple options within an optional element are separated by a pipe symbol, for example: [option1</td>
</tr>
</tbody>
</table>
1

About Oracle GoldenGate Veridata

This chapter describes how to use Oracle GoldenGate Veridata. It provides an overview of roles and interactions of the components, how to configure components, and how Oracle GoldenGate Veridata compares tables.

This chapter includes the following sections:

Section 1.1, "Oracle GoldenGate Veridata Architecture"
Section 1.2, "Comparing Data with Oracle GoldenGate Veridata"
Section 1.3, "Viewing Comparison Results"

1.1 Oracle GoldenGate Veridata Architecture

Oracle GoldenGate Veridata compares one set of data to another and identifies data that is out of synchronization. Oracle GoldenGate Veridata supports high-volume, 24x7 replication environments where downtime to compare data sets is not an option. By accounting for data that is being replicated while a comparison takes place, Oracle GoldenGate Veridata can run concurrently with data transactions and replication, while still producing an accurate comparison report.

Oracle GoldenGate Veridata will map column data types across different types of databases automatically, or you can map columns manually in cases where the automatic mapping is not sufficient to accommodate format differences in a heterogeneous environment. For detailed information about this feature, see the online help.

For the purposes of this documentation, the following terms are considered synonymous:

- tables and files
- columns and fields
- rows and records
Oracle GoldenGate Veridata Server
The Oracle GoldenGate Veridata Server performs the following functions:

- Coordinate the execution of Oracle GoldenGate Veridata tasks
- Sort rows (optional)
- Compare data
- Confirm out-of-sync data
- Produce a report for review

Oracle GoldenGate Veridata Web User Interface
Oracle GoldenGate Veridata Web User Interface (UI) is a browser-based graphical user interface for these activities:

- Configure comparison objects and rules
- Initiate comparisons
- Review the status and output of comparisons
- Review out-of-sync data

Oracle GoldenGate Veridata Repository
The Oracle GoldenGate Veridata repository is a collection of database objects that persists configuration information to disk, saving it permanently as a user environment.

Oracle GoldenGate Veridata Agent
The Oracle GoldenGate Veridata Agent executes the following database-related requests on behalf of the Oracle GoldenGate Veridata Server:

- Fetch and return blocks of rows to compare
- Return column-level detail for out-of-sync rows

Oracle GoldenGate Veridata Manager
The Manager process is part of the C-code based agent that is required for the NonStop platform and optional for the Oracle database. It controls the Oracle GoldenGate Veridata Agent process. A Manager is not used in a Java agent, which is used for the other databases that are supported by Oracle GoldenGate Veridata.
1.2 Comparing Data with Oracle GoldenGate Veridata

This section explains how to configure the objects that are to be compared and how Oracle GoldenGate Veridata processes comparisons.

1.2.1 Oracle GoldenGate Veridata Comparison Objects

To begin using Oracle GoldenGate Veridata, you need to create some objects that help you identify the data you want to compare and help you manage your work.

- Configure data source connections: Oracle GoldenGate Veridata Server connects to Oracle GoldenGate Veridata Agents that interact with the databases that contain the data that is to be compared. A connection is defined by a host, the port number of an Oracle GoldenGate Veridata Agent on that host, and the data source that is accessed by the agent.

- Configure groups: You need to configure at least one compare group that is linked to a set of source and target data source connections. A group is a logical container for organizing the objects to be compared.

- Configure compare pairs: You need to configure one or more compare pairs for each group that you create. A compare pair is a set of corresponding source and target tables or files. As part of configuring compare pairs, you map source and target columns to establish a structural relationship between the two objects.

- Configure profiles: A profile contains settings for run-time parameters and can be applied globally to a job, as well as to a specific compare pair as an override to the job profile. Profile parameters encompass considerations such as sorting method, thread and memory usage, reporting output, and so forth. Defining run profiles is optional, because Oracle GoldenGate Veridata includes a default profile that contains settings that apply to most usage scenarios. However, as you gain experience with Oracle GoldenGate Veridata, you may want to customize the default profile or create your own custom profiles.

- Configure jobs: A job is a logical container for one or more compare groups and is the unit of work by which comparison processing is executed. Within one or more jobs, you can manage and run large volumes of compare groups across numerous databases and systems, and you can control the timing of those comparisons.

For more information, see the Oracle GoldenGate Veridata online help.

1.2.2 Satisfying Uniqueness Requirements

Oracle GoldenGate Veridata relies on some form of unique identifier to order rows for comparison.

- **Primary Key**: By default, Oracle GoldenGate Veridata uses the primary key if one is available.

- **Unique Key**: If no primary key is defined, Oracle GoldenGate Veridata uses the smallest unique index
User-defined Key: If a table or file has neither a primary nor unique key, you can define an existing index or set of columns for comparison purposes when defining a compare pair. However, although primary or unique keys can be mapped automatically, user-defined keys must be mapped manually. A user-defined key can also be used to override existing keys or indexes if you prefer a different ordering method.

For more information about choosing and mapping keys for comparison, see the Oracle GoldenGate Veridata online help.

1.2.3 How Oracle GoldenGate Veridata Compares Data

Comparison activities are divided into the following steps. You can change some of the aspects of these steps by making parameter changes in Oracle GoldenGate Veridata Web.

1.2.3.1 Initial Comparison Step

In the initial comparison (or row hash) step, rows are retrieved from the source and target tables with a query. If the source and target databases are of different types, the columns are converted to a standardized data type format for accurate comparison. By default, Oracle GoldenGate Veridata compares rows by comparing all columns of the primary key literally (value-for-value) and by using a hash value for all non-key columns. The unique digital signature that is used to calculate the hash value shrinks the data to be transferred over the network for the comparison, while still providing a highly reliable (but not absolute) and efficient mechanism for determining whether two rows contain the same or different column values.

For more assurance of discovering out-of-sync rows, you can configure Oracle GoldenGate Veridata to compare non-key rows column-for-column, instead of using a hash. Full-column comparisons reduce the processing performance in proportion to the number of columns, and they increase network usage.

On the NonStop platform, you can use the delta processing feature during the initial comparison step if you are using server-side sorting. Delta processing is a performance feature by which Oracle GoldenGate Veridata detects which data blocks in the database were modified since a previous comparison and only compares the rows in those blocks. Rows in unchanged blocks are skipped. The default is to compare all rows regardless of whether they changed or not.

There are two steps to delta processing:

- Collecting the base modification time of the previous run for subsequent delta comparisons. This step is always done when delta processing is enabled for a compare pair.
- Comparing data that has been modified since the base comparison, using the information that was collected in the first step. This step is enabled by clicking the Enable Delta Processing button on the Compare Pair Configuration page and the Run/Execute Job page of Oracle GoldenGate Veridata Web. The Disable Delta Processing button allows you to disable the delta comparison step in case there were modifications, such as table reorganizations, that can invalidate the collected delta base information.

For more information about delta processing, see the Oracle GoldenGate Veridata online help.

After the initial comparison, rows that appear to be out-of-sync are stored in a maybe out-of-sync (MOOS) queue in memory, because at this point the comparison is inconclusive. When replication is working concurrently with a comparison, especially
if there is replication latency, rows can appear to be out-of-sync when, in fact, the current data is in flight (somewhere in the replication flow) and replication will soon synchronize them again.

1.2.3.2 Confirmation Step
The confirmation, or confirm-out-of-sync (COOS), step ensures accurate results by confirming row status in a changing environment. In this step, rows are extracted from the MOOS queue, and their status is evaluated as one of the following:

- **in-flight**: the row was out-of-sync in the initial comparison step, but has since been updated. In this case, it is assumed that replication or another mechanism applied the change, but Oracle GoldenGate Veridata was unable to confirm that the rows were in-sync.

- **in-sync**: the source row values were applied to the target row by replication or another method. Even a status of in-sync does not guarantee that the rows are synchronized at any particular moment if the underlying tables are continuously changing, but it does indicate that replication is working.

- **persistently out-of-sync**: the row has not been updated since the initial comparison step took place, and therefore can be assumed to be out-of-sync.

By default, confirmation processing occurs in a thread that is parallel to the initial comparison step, but the confirmation of each row waits until after a specified replication latency threshold has expired. For example, if latency is 60 seconds, and the initial comparison step revealed an out-of-sync row at 9:30, then the confirmation step for that row is not performed until 9:31 to allow replication to apply any change that was in-flight. After latency is accounted for, rows can be confirmed as persistently out-of-sync and are stored in one or more out-of-sync reports.

1.3 Viewing Comparison Results
Upon completion of a job, you can view the comparison reports and the out-of-sync report by using Oracle GoldenGate Veridata Web or by viewing the files themselves.

1.3.1 Out-of-Sync Report
You have the option to store an out-of-sync report in binary format, in XML format, or both (or none).

- **OOS file**: When stored in binary form, the OOS report is compatible with the Oracle GoldenGate Veridata programs. As such, it can be viewed from the Oracle GoldenGate Veridata Web interface, and it also can be used as the basis for re-comparing rows that were out of synchronization. To re-compare rows, you select run options to execute another confirmation step, which compares the current state of just those rows and then reports which ones remain out-of-sync after replication or another restorative procedure has been applied.

- **OOSXML file**: When stored as XML, the OOS report is written to an OOSXML file and is stored in a structured way that conforms to an internal XML schema. XML has many advantages, the largest being that it can be manipulated easily by many tools. In its XML form, the file contains all of the information, including metadata, that is needed to select rows for resynchronization by external programs.
1.3.2 Comparison Report

Each finished job, group, and compare pair generates a comparison report. The report file contains details about the comparisons that were performed, such as:

- Comparison parameters used
- The number of rows compared and out-of-sync
- The timing of the comparison
- Performance statistics
- Source and target data values

The files themselves are stored as follows:

The OOS files are located in sub-directories of the Oracle GoldenGate Veridata Server installation directory:

- OOS files: shared/data/oos
- OOSXML files: shared/data/oosxml

These directories are further organized by run ID, job name, group name, and compare pair. In the OOSXML directory, the files with the .oosxml extension are the control files. The files with sequential file extensions are the OOSXML chunks. The XML data is spread into multiple files (called "chunks") for performance purposes.
This chapter discusses the supported databases and system requirements to install and run Oracle GoldenGate Veridata.

This chapter includes the following sections:

Section 2.1, "General Information"
Section 2.2, "Supported Databases"
Section 2.3, "Oracle GoldenGate Veridata Agent System Requirements"
Section 2.4, "Oracle GoldenGate Veridata Server System Requirements"
Section 2.5, "Oracle GoldenGate Veridata Web User Interface Requirements"
Section 2.6, "SQL/MP Data Exclusions"

2.1 General Information

Oracle GoldenGate Veridata does not require an installation of the Oracle GoldenGate replication software. If you do use that software, install Oracle GoldenGate Veridata in a different location.

2.2 Supported Databases

Oracle GoldenGate Veridata supports the following databases for comparisons:

- DB2 LUW and z/OS
- Enscribe
- Oracle
- NonStop SQL/MP
- SQL Server
- Sybase Adaptive Server Enterprise (ASE)
- Teradata

A list of currently supported database versions and operating systems can be found at http://support.oracle.com.

2.3 Oracle GoldenGate Veridata Agent System Requirements

One Oracle GoldenGate Veridata Agent must be installed for each database instance that contains data that is to be compared. At minimum, therefore, you will install two agents — one to retrieve source rows and one to retrieve target rows (unless you are
comparing data within the same database instance). One agent can retrieve rows from multiple databases or schemas within a given database instance. However, one agent cannot retrieve rows from different database instances.

**Comparing Multibyte Data**
The following considerations apply when you are comparing tables with multibyte data:

- A Java agent should be used for all platforms except NonStop, which has only a C-agent.
- The Java agent uses the UTF-8 character for comparing character data. Out-of-sync data is written to the report file using the UTF-8 character set.
- The Oracle GoldenGate Veridata Oracle C-Agent can be used for comparisons between Oracle databases where the source and target use the same character set and the host system uses the same byte order. An Oracle C-Agent cannot be used in a comparison with a Java agent. Character fields that contain characters that are not valid UTF-8 characters are displayed as hexadecimal in the out-of-sync reports.

### 2.3.1 Oracle GoldenGate Veridata Java agent

Oracle GoldenGate Veridata provides a Java-based agent for the following databases:

- DB2 LUW and Z/OS
- Oracle
- SQL Server
- Sybase ASE
- Teradata

The Java agent connects to the database by using JDBC (Java Database Connectivity). The Java agent enables Oracle GoldenGate Veridata to support comparisons in a heterogeneous environment, where different kinds of databases contain similar, but not identical data types. With a Java agent, one executable supports many operating systems and databases.

You can install the Java agent on the same system as the one where the database is running, or you can install it on a different system, such as the one where Oracle GoldenGate Veridata Server is installed. When considering where to install the agent, weigh the additional and significant use of network bandwidth that will be incurred if the agent is remote from the database, versus the savings in processing resources on the database host when an agent is not running there.

### 2.3.1.1 Environment Requirements to Use a Java Agent

**UNIX and Linux, all supported databases**

A Java environment is required on UNIX and Linux systems. Download and install either the Java Software Developer’s Kit (JSDK) or the Java Runtime Environment (JRE) from the Sun website. The minimum Java version is 1.6, and 1.6_24 is preferred because of the performance improvements for Oracle GoldenGate Veridata.

**DB2**

A TCP/IP port must be configured and active.
Oracle
The listener must be configured and running.

SQL Server
A static TCP/IP port must be configured and enabled.

Sybase
Know the ASE listen port.

Teradata
- Know the host name and port number of the database.
- Download the appropriate JDBC driver from the database vendor’s website before installing the Java agent. You will be prompted for the location during the installation steps in this guide.

2.3.2 Oracle GoldenGate Veridata C-Agent
Oracle GoldenGate Veridata also provides a C-code based agent for the following databases:
- The C-agent is required for NonStop SQL/MP and Enscribe databases running on the NonStop platform. This agent can be installed in a Guardian environment.
- The C-agent is supported for the Oracle database, but as of release 11.2.1.0.0, it will no longer be enhanced.
- You must install the C-agent on the same system as the one that hosts the database.
- For Oracle, a dynamically linked C-agent is available. If you use this agent, you must set the following environment variables:

<table>
<thead>
<tr>
<th>Environment Variables for Dynamically Linked C-Agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
</tr>
<tr>
<td>PATH=Oracle_libraries</td>
</tr>
<tr>
<td>SHLIB_PATH=Oracle_libraries</td>
</tr>
<tr>
<td>LIBPATH=Oracle_libraries</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH=Oracle_libraries</td>
</tr>
</tbody>
</table>

2.3.3 Disk and Memory Requirements for the Agent Component
- The agent requires at least 1GB of RAM.
- The disk space requirements for the Oracle GoldenGate Veridata Agent vary by platform, but up to 200 MB may be required. On UNIX and Linux, additional space might be required to install the Java environment (if not already installed).
- The main consumers of processing resources are the row sorting operations that are required during a comparison. To improve performance, you might need to increase the temporary memory space in the database if the columns that are being used as keys are not a native unique index or primary key. You specify the columns to use as keys when configuring Oracle GoldenGate Veridata.
Using server-side sorting instead of database sorting might reduce the load on the database server and improve comparison performance, depending on the number of rows, the indexes defined, the keys used, and the way the database is tuned. See Section 2.4.2, “Disk and Memory Requirements for the Server Component.”

2.3.4 Database Privileges for the Agent Component

Oracle GoldenGate Veridata Agent makes use of a database login, which must be created before you can run comparisons. You provide the login and password when you configure connection objects in the Oracle GoldenGate Veridata Web interface. The following are the database privileges that are required for the database user.

Required database privileges for Oracle GoldenGate Veridata Agent

**DB2**
- SELECT privileges on the tables that will be compared.

**Oracle**
- GRANT CONNECT
- GRANT SELECT on the tables to be compared. It is recommended, but not necessary, to GRANT SELECT ANY TABLE.
- SELECT_CATALOG_ROLE

**NonStop SQL/MP and Enscribe**
- Read access to the SQL/MP system catalog (for queries to CATALOGS table).
- Read access to the SQL/MP catalogs that you want Oracle GoldenGate Veridata to use.
- Read access to the DDL dictionaries that you want Oracle GoldenGate Veridata to use.
- Read access to the Enscribe and SQL/MP tables that will be compared.
- Read, write, create, purge permissions for the Oracle GoldenGate Veridata report and trace files, and access to the subvolumes where they are installed.

**SQL Server**
- **db_datarreader** or the equivalent on the tables to be compared.
- **VIEW DEFINITION** in the databases to be compared.
- The database must allow SQL Server authentication.

**Sybase ASE**
- Access to the databases to be compared.
- SELECT privileges on the tables to be compared.
- SELECT privileges on the sysdatabases system table in the master database to view the list of databases available in the server.

**Teradata**
- SELECT privileges on the tables to be compared.
2.4 Oracle GoldenGate Veridata Server System Requirements

This section describes the installation location, additional programs, disk, memory, and repository requirements for Oracle GoldenGate Veridata Server.

2.4.1 Location for the Server Component

The server and web user interface components are installed from one installation program on Windows, UNIX, and Linux systems. The installer includes all files that are needed to run those programs. One installation can be used for comparisons among all of the supported databases, but multiple installations can be used as needed.

Do not install the server and web user interface components on a NonStop system. To use Oracle GoldenGate Veridata for NonStop databases:

- Install the server and web user interface components on a supported Windows, UNIX, or Linux system.
- Make certain that this system has access over high-speed network connections to the NonStop systems.

2.4.2 Disk and Memory Requirements for the Server Component

The server component uses about 200 MB of fixed virtual memory for basic tasks. The remaining virtual memory is used for comparisons. The main consumers of processing resources on the Oracle GoldenGate Veridata machine are the row sorting operations of the initial comparison step when using server-side sorting.

Enough combined disk space and virtual memory is needed to store all of the rows that are sent for comparison from the source and target systems. To estimate the amount of memory per row:

\[
((\text{number of cols in key} + 1) \times 4) + 16 + \text{(comparison width of a key col)}
\]

Where:

- comparison width of a key col depends on the comparison format that is selected by Oracle GoldenGate Veridata (or a user override) to use for a comparison.

Comparison format data sizes:

<table>
<thead>
<tr>
<th>Comparison Format</th>
<th>Data Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>One byte for each significant digit. Leading zeros and trailing zeros after the decimal point (such as the rightmost zeros in 1234.500) are not counted.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>19 to 32 bytes depending on the fractional precision.</td>
</tr>
<tr>
<td>Date</td>
<td>10 bytes.</td>
</tr>
<tr>
<td>Time</td>
<td>8 to 18 bytes depending on the fractional precision.</td>
</tr>
<tr>
<td>String</td>
<td>1 to 4 bytes per character for the UTF-8 encoding of the Java agent. The NonStop agent and the Oracle C agent use the database native character set.</td>
</tr>
<tr>
<td>Binary</td>
<td>The bytes as stored in the database.</td>
</tr>
</tbody>
</table>

For example, the number 109998877, if compared as a decimal float, would require:

\[
((1 + 1) \times 4) + 16 + 9 = 33 \text{ bytes of memory for this row}
\]
Oracle GoldenGate Veridata uses an external merge sort to sort the data. As data is received from the agent, the rows are sorted in memory. When a memory buffer is full, the sorted rows are written to disk.

In order to sort the data, the sort process matches the initial data set size for temporary storage space. The required amount of temporary space is determined by the number of rows, the row size, and the amount of available sort memory. The following cases illustrate the different modes of the sort depending on the available resources.

- **In-Memory Sort**: This sorts the data entirely in memory and is the fastest method, but the memory requirements may exceed what is available. The sort memory must be approximately 2.5 times larger than the size of the data set.

- **One Disk Pass**: This sorts data and writes to the disk only once. It requires sort disk space equal to the size of the data set. This process is almost as fast as the in-memory sort and the memory requirements are lower. In general, the Oracle GoldenGate Veridata server can write the rows to disk faster than the agent can read them from the database.

- **Two Disk Passes**: This sorts and writes to the disk twice, requiring sort disk space twice the size of the data set. Although the disk requirement is greater, very large data sets can be sorted with a reasonable amount of memory.

- **Three or More Disk Passes**: After all of the rows have been received from the agent, additional sorting may be required before the rows are ready for the final write to disk. If it is necessary to access the disk three or more times, the required sort space will be three or more times the data set size. This is slow and should be avoided.

The following table shows memory requirements for some sample data sets.

<table>
<thead>
<tr>
<th>Data Set Size</th>
<th>In-Memory Sort</th>
<th>One Disk Pass</th>
<th>Two Disk Passes</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 MB</td>
<td>50 MB</td>
<td>Not Applicable</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>400 MB</td>
<td>1 GB</td>
<td>50 MB</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>2 GB</td>
<td>5 GB</td>
<td>125 MB</td>
<td>50 MB</td>
</tr>
<tr>
<td>8 GB</td>
<td>20 GB</td>
<td>250 GB</td>
<td>50 MB</td>
</tr>
<tr>
<td>50 GB</td>
<td>125 GB</td>
<td>600 MB</td>
<td>80 MB</td>
</tr>
</tbody>
</table>

**Note:** The numbers in the previous table are for only one side of the comparison. You must allocate twice this to cover both sides.

Beyond this allocation, memory is required for storing rows during the second step of processing, the confirmation step. This can be up to 20 MB if you expect a large number of rows to require confirmation, as is usually the case when replication latency is very high. These rows are staged in the main memory before they are confirmed.

The sort performs better if all of the data that needs to be sorted during the initial comparison step can be stored in main memory. On 64-bit systems, more memory can be addressed, so more data can be stored in main memory instead of on slower disk devices. The memory that is used in the initial comparison step is not necessarily all...
released at once to be available for the confirmation step. Consequently, some memory will be shared between processes. When the sort cannot hold all of the rows in memory, it uses disk storage.

When deciding how much memory to allocate, be aware of the following ways that you can manage it with parameter settings within the Oracle GoldenGate Veridata application:

- The temporary space should be located on a reasonably fast file system. A network file system located on a remote server may slow the comparison processing.
- You can increase disk I/O performance by specifying multiple temporary directories with profile settings. For maximum benefit, put the directories on different physical disks.
- You can control the amount of memory that is allocated to each of the processing steps with profile settings.
- You can use a profile setting to terminate the confirmation step after a given number of out-of-sync rows, to work around resource limitations.
- Additional memory properties can be controlled with server parameters. See Chapter 11, "Oracle GoldenGate Veridata Server Configuration Parameters".

### 2.4.3 Repository Requirements for the Server Component

Oracle GoldenGate Veridata Server requires a database to serve as a repository for objects that store the information and environment preferences that users create when using Oracle GoldenGate Veridata Web. You can use the following databases as a repository:

- MySQL
- Oracle
- SQL Server

When installing Oracle GoldenGate Veridata Server, you can have the installer create a database (or schema if using Oracle) for the repository or install the repository objects into an existing database or schema.

The repository can be installed in a database on the same system as the server and web components, or it can be installed on a different system in a multi-tier configuration. For a multi-tier installation, enable communication between hosts as follows:

- To use a MySQL database repository, all required MySQL client software is installed with the server and web components.
- To use an Oracle database repository with Oracle GoldenGate Veridata server and web components installed on a Windows, UNIX, or Linux system, the Oracle client software is installed with the server and web components. Oracle GoldenGate Veridata supports TNSNAMES and EZCONNECT connection naming methods.
  - To use TNSNAMES, the database instance where the repository will be installed must be included in the tnsnames.ora file and in the listener.ora file. Make certain that TNSNAMES is one of the values in the NAMES.DIRECTORY_PATH parameter in the sqlnet.ora file on the database system.
  - To use EZCONNECT, you will need to supply the installation program with the name or IP address of the database server (domain-qualified if required by the operating system), the database listening port number, and the instance name of the database.
To use a SQL Server database repository with server and web components installed on a different system, make certain that both systems are registered with the DNS.

### 2.4.3.1 Database Privileges

During installation, you will be prompted for a user (or login) and password for Oracle GoldenGate Veridata to use when updating the repository. You can create the user with the required privileges before running the installation program, or you can allow the installation program to create it with the required privileges. Dedicate this user to Oracle GoldenGate Veridata.

To have the Oracle GoldenGate Veridata installation program create this user, an Administrator or DBA user and password are required. This user will only be used for this purpose and will not be retained in the Oracle GoldenGate Veridata configuration.

If creating the user outside the installer, grant the following privileges to create, update, and delete the tables in the repository. Have the user information available for the installation.

### Required Database Privileges for Oracle GoldenGate Veridata Server

**MySQL**

1. Create a user and identically named database.
2. Grant the user all DDL and DML privileges in that database when this user connects to the MySQL server from the host where Oracle GoldenGate Veridata is being installed.

**Oracle**

1. Create a user and password.
2. Create a database role named VERIDATA_ROLE.
3. Grant these to VERIDATA_ROLE:
   - CREATE SESSION
   - CREATE TABLE
   - CREATE VIEW
   - CREATE PROCEDURE
   - CREATE SYNONYM
4. Grant VERIDATA_ROLE to the user.
5. Specify QUOTA UNLIMITED on the user’s default tablespace.

**SQL Server**

1. Create a database and login that is a user of the database where you will be installing the repository.
2. Grant this login sufficient privileges to perform the following operations:
   - CONNECT to the database
   - CREATE, ALTER, DROP TABLE in the database
   - CREATE and DROP INDEX in the database
2.4.3.2 Database Environment

During installation, you will be prompted for the following components. Have this information available for the installation.

Required environment components for Oracle GoldenGate Veridata Server

MySQL
1. The database host name and port number.
2. A login and password, if using an existing database user for the repository.

Oracle
1. The database instance name
2. The ORACLE_HOME (if using TNSNAMES connection method)
3. Default and temporary table spaces for the repository (if you will be creating a new user for the server component during installation)
4. The host name, port number, and service name of the database instance (if using EZCONNECT connection method)
5. A login and password, if using an existing database user for the repository.

SQL Server
1. An ODBC data source name (DSN). This must be a system DSN. You can create the DSN before installing Oracle GoldenGate Veridata or during the installation (you will be prompted for it). A system administrator can create a DSN from the Data Sources (ODBC) applet of the Administrative Tools control panel. (For basic instructions, see Appendix C, "Configuring an ODBC connection for a SQL Server Repository"). Any of the SQL Server clients can be used.
2. The SQL Server instance must be configured to use a static port number. This port is stored in a configuration file so that it can be referenced to make JDBC connections to the database.
3. A login and password (or integrated authentication), if using an existing database user for the repository.

2.5 Oracle GoldenGate Veridata Web User Interface Requirements

This section describes the Java package, browsers, screen resolution, security, and ports recommended or required for the Oracle GoldenGate Web User Interface.

2.5.1 Location for the Web Component

See Section 2.4.1, "Location for the Server Component". One installer program installs both Oracle GoldenGate VeridataWeb Server component and Oracle GoldenGate Veridata Web User Interface.
2.5.2 Java Environment for the Web User Interface

Except for UNIX and Linux, the installation package includes the required Java Run-time Environment (JRE) 1.6.x, which is installed into the JRE directory within the main Oracle GoldenGate Veridata installation. To ensure that the web client runs properly, do not change this JRE environment.

2.5.3 Supported Browsers for the Web User Interface

- Microsoft Internet Explorer version 6 or higher
- Mozilla Firefox version 1.0.4 and higher

2.5.4 Recommended Screen Resolution for the Web User Interface

Minimum of 1024 x 768

2.5.5 Port Numbers for the Web User Interface

- The Oracle GoldenGate Veridata Web application interacts with the Apache Tomcat web server. Tomcat is installed with Oracle GoldenGate Veridata Server on default ports. You can change the port numbers if needed (see Appendix B, "Changing Oracle GoldenGate Veridata Web Port Numbers").
- Oracle GoldenGate Veridata Web makes use of the following default ports, which you can change during installation:
  - 8820: Shutdown port
  - 8830: HTTP port
  - Other port numbers are used by default. You can change these port numbers after installation by following the directions in Appendix B, "Changing Oracle GoldenGate Veridata Web Port Numbers".

2.5.6 Security for the Web User Interface

The Tomcat web server provides user security roles that control access to certain pages of the Oracle GoldenGate Veridata Web interface. You will be asked to specify an initial Oracle GoldenGate Veridata Administrator user and password during installation. You can use this user to create additional users after installation and assign them to roles.

2.6 SQL/MP Data Exclusions

On SQL/MP tables with clustered keys, the right-most portion of the primary key is a timestamp added by the file system. For compare pairs composed of such tables, the timestamp will be different on the source and target systems. As a result, it must be excluded from the comparison, because Oracle GoldenGate Veridata compares keys to ensure that the correct rows are compared.

If, after the timestamp portion of a key is eliminated, the remaining key columns do not produce a unique key value, then those tables must have a unique index or other columns that can be specified to create a unique value. Otherwise, they cannot be compared with Oracle GoldenGate Veridata.
This chapter describes how to install the Oracle GoldenGate Veridata Java Agent.

This chapter includes the following sections:

Section 3.1, "Installation Overview"
Section 3.2, "Installing the Java Agent on a UNIX or Linux System"
Section 3.3, "Installing the Java Agent on a Windows System"

### 3.1 Installation Overview

These instructions are for installing a new, clean copy of the Java-based Oracle GoldenGate Veridata Agent software. To upgrade an existing installation, see Chapter 12, "Upgrading Oracle GoldenGate Veridata".

You can install Oracle GoldenGate Veridata Agent to work with any of the following databases:

- DB2 LUW and z/OS
- Oracle
- SQL Server
- Sybase Adaptive Server Enterprise (ASE)
- Teradata

The Java agent can be installed on the database host or on a remote host that has network connectivity to the database host. The agent connects by JDBC (Java Database Connectivity).

### 3.2 Installing the Java Agent on a UNIX or Linux System

**Note:** Make certain you have reviewed the System Requirements before installing Oracle GoldenGate Veridata Agent.

1. Create a directory for the agent.
2. Follow the steps in Appendix A, "Downloading Oracle GoldenGate Veridata" to obtain the software.
3. Extract the mediapack.zip file to the UNIX or Linux system and directory where you want the software to be installed.
4. In the agent directory within the installation directory, make a copy of the agent.properties.sample file, and rename the copy to agent.properties.

5. Open the agent.properties file with a text editor. This file contains comments that help guide your work.

6. Edit the file to provide values for the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.port</td>
<td>A local port where the agent listens for communication requests from Oracle GoldenGate Veridata Server.</td>
</tr>
</tbody>
</table>
| database.url        | The standard JDBC connection URL string for the database. Samples of the correct format to use are listed with comments in the properties file. You must supply the following values that are appropriate for your database:  
  - Name of the host that contains the database. To use Oracle OCI bequeath protocol, the Oracle OCI libraries must be available locally.  
  - Port number that the database uses.  
  - Name of the database or (for Oracle) the instance.  
  This is a sample URL for Oracle.  
  database.url=jdbc:oracle:thin:@localhost:1521:orcl |
| server.driversLocation | Specify the location of the JDBC driver. The JDBC drivers are installed by default into the drivers directory. Change this value only if you need to move the location (not recommended). |
| server.jdbcDriver   | The name of the JDBC driver. The drivers are listed with comments in the properties file. Uncomment the correct one for the database. Example (in this case Oracle):  
  server.jdbcDriver=ojdbc6.jar |
| database.transaction.isolation | Controls the transaction isolation level that is used during the initial comparison step. |

7. Save and close the file.

3.3 Installing the Java Agent on a Windows System

**Note:** Make certain you have reviewed the System Requirements before installing Oracle GoldenGate Veridata Agent.

1. Start the database for which you are installing this agent.

2. Follow the steps in Appendix A, "Downloading Oracle GoldenGate Veridata" to obtain the software.

3. Unzip the .zip file that you downloaded.

4. Run the GoldenGate_Veridata_Agent_platformrelease.exe executable in either of these ways:  
  - As a GUI wizard: Double click the program and then follow these steps. This is the method shown in this documentation.
As a shell process: From the command console, issue the following command, and then continue with these steps. The prompts are similar to those shown on the wizard screens.

```bash
Start /wait GoldenGate_Veridata_Agent_platformrelease.exe -c
```

5. Welcome screen: Click Next on the first page to start the installation.

6. Destination Directory: Select an installation directory. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.

7. Start Menu Folder: To add Oracle GoldenGate Veridata Agent to the Programs list of the Start menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list.
   - To make the shortcut available to everyone who uses this system, click Create Shortcuts for all users; otherwise, it will only be available to the current user.
   - To exclude Oracle GoldenGate Veridata Agent from the Start menu, click Don't create a Start Menu folder. If you are installing this software as a service, it can be started from the installation folder or the Services control panel.

8. Agent Port: Specify a local port for the agent to use for communication with Oracle GoldenGate Veridata Server.

9. Database Details: Select the database type, and specify the name of the host system, the port number that is used by the database, the name of the database (if applicable), and (if Oracle) the name of the instance to which the agent will connect. Under JDBC URL, the resultant connection string is displayed.

10. Teradata JDBC Drivers: (Teradata only) Supply the location of the Teradata JDBC driver.

11. Database Connection Test: If you did not override the default of Test Database Connection, a test screen prompts for the authentication credentials that you created for the agent to use with this database or for an ODBC driver location, depending on the database type. Click Test Connection. Any connection errors are displayed in the output area.

12. Service Option: It is recommended that you install Oracle GoldenGate Veridata Agent as a Windows service. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up. Either accept the default service name or change it as required. If there are, or will be, other instances of this software on the local system, make certain that each name is unique. To prevent installation as a service, deselect the check box.

13. Start Veridata Agent After Install: Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.

14. Information: Review your installation selections. Click Back to make changes or click Next to begin the installation.

15. Click Finish to exit the installer.
This chapter explains how to install the Oracle GoldenGate Veridata C-agent on a UNIX/Linux, Windows, or NonStop platform.

This chapter includes the following sections:

Section 4.1, "Installation Overview"
Section 4.2, "Installing the C-agent on a UNIX or Linux System"
Section 4.3, "Installing the C-Agent on a Windows System"
Section 4.4, "Installing the C-Agent on a NonStop System"

4.1 Installation Overview

These instructions are for installing a new, clean copy of the C-based Oracle GoldenGate Veridata Agent software. To upgrade an existing installation, see Chapter 12, "Upgrading Oracle GoldenGate Veridata".

The Oracle GoldenGate Veridata C-agent is installed on the same system that hosts the database that contains compare data. You will install one Oracle GoldenGate Veridata C-Agent for each database instance where there is data that is to be compared.

4.2 Installing the C-agent on a UNIX or Linux System

To install the C-agent on a UNIX or Linux system, perform the following steps:

1. Create a directory for the agent.
2. Follow the steps in Appendix A, "Downloading Oracle GoldenGate Veridata" to obtain the software.
3. Extract the mediapack.zip file to directory that you created.
4. From the agent subdirectory, run GGSCI.
5. In GGSCI, issue the following command to create the working directories for Oracle GoldenGate Veridata.
   CREATE SUBDIRS
6. In GGSCI, issue the following command to create and edit a Manager parameter file.
   EDIT PARAMS MGR
7. On the first line of the file, add the following parameter to specify a port number for the Manager process. This port number must be a unique number that is not being used by any other process, including any Manager processes for other Oracle GoldenGate software.

   PORT number

8. (Optional) On the next line, add the following parameter to specify a range of up to 256 ports that the Manager process can allocate dynamically. This parameter can be used to specify ports for concurrent processing threads if you will be running batch comparisons.

   DYNAMICPORTLIST {port | port-port} [, ...]

   ■ To specify multiple ports, use a comma-delimited list, for example 7830, 7833.
   ■ To specify a range of ports, use a dash (-) to separate the first and last port in the range, for example 7830-7835.
   ■ To specify a range of ports plus an individual port, place a comma between the range and the individual port number, for example 7830-7835, 7839.

9. Save and close the parameter file. This file is stored in the dirprm directory within the agent subdirectory. Do not move it.

10. In GGSCI, start the Manager process. You can defer this step until you are ready to run comparisons. To perform comparisons, Manager must be running.

    START MANAGER

11. To confirm that Manager is running, issue the following command in GGSCI.

    INFO MGR

---

**Note:** Oracle GoldenGate Veridata cannot be installed as a service on a UNIX or Linux system.

---

### 4.3 Installing the C-Agent on a Windows System

To install the C-agent on a Windows system, the following steps are required:

- Install the Oracle GoldenGate Veridata Agent files.
- Specify a custom Manager name (if installed as a service)
- Install the Manager process as a Windows service (recommended).
- Configure the Manager process.

#### 4.3.1 Installing the Oracle GoldenGate Veridata Agent Files

1. Create a directory for the agent.

2. Follow the steps in Appendix A, "Downloading Oracle GoldenGate Veridata" to obtain the software.

3. Using WinZip or an equivalent compression product, extract the files from the mediapack.zip file to the directory that you created, and make certain to preserve folder names. The extraction creates the installation directory and a subdirectory named agent that contains the software files.
4. From the agent subdirectory, run GGSCI.

5. In GGSCI, issue the following command to create the working directories for the agent.

   \[ \text{CREATE SUBDIRS} \]

6. If you will be installing the Manager as a Windows service and there are other local Manager services, continue with the following steps to specify a custom Manager name. Otherwise, skip that section and follow the instructions for installing Manager as a service on Section 4.3.3, "Installing Manager as a Windows service".

### 4.3.2 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 \( \text{GGSMGR} \).
- there will be multiple Manager processes running as Windows services on this system, such as one for the Oracle GoldenGate replication software and one for Oracle GoldenGate Veridata. Each Manager on a system must have a unique name. Before proceeding further, verify the names of any local Manager services.

To specify a custom Manager name

1. From the directory that contains the Manager program, run GGSCI.

2. Issue the following command.

   \[ \text{EDIT PARAMS ./GLOBALS} \]

3. In the file, add the following line, where \( \text{name} \) is a one-word name for the Manager service.

   \[ \text{MGRSERVNAME name} \]

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

### 4.3.3 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.

To install Manager as a Windows service

1. (Recommended) Log on as the system administrator.

2. Click **Start**, then **Run**, and type `cmd` in the Run dialog box.

3. From the directory that contains the Manager program that you are installing as a service, run the INSTALL program with the following syntax:

   \[ \text{install option [...]} \]

   Where:

   \[ \text{option} \] is one of the following:
4. (Windows Server 2008) If Windows User Account Control (UAC) is enabled, you are prompted to allow or deny the program access to the computer. Click Allow to enable the INSTALL program to run. This installs the Manager service with a local system account running with administrator privileges. No further UAC prompts will be encountered when running Manager if installed as a service.

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

### 4.3.4 Configuring Manager

1. In GGSCI, issue the following command to create and edit a Manager parameter file.

   ```bash
   EDIT PARAMS MGR
   ```

2. On the first line of the file, add the following parameter to specify a port number for the Manager process. This port number must be a unique number that is not
being used by any other process, including any Manager processes for other Oracle GoldenGate software.

```
PORT number
```

3. (Optional) On the next line, add the following parameter to specify a range of up to 256 ports that the Manager process can allocate dynamically. This parameter can be used to specify ports for concurrent processing threads if you will be running batch comparisons.

```
DYNAMICPORTLIST {port | port-port} [ , ...]
```

   Where:
   
   ■ To specify multiple ports, use a comma-delimited list, for example 7830, 7833.
   
   ■ To specify a range of ports, use a dash (-) to separate the first and last port in the range, for example 7830-7835.
   
   ■ To specify a range of ports plus an individual port, place a comma between the range and the individual port number, for example 7830-7835, 7839.

4. Save and close the parameter file. This file is stored in the `dirprm` subdirectory of the agent directory. Do not move it.

5. In GGSCI, start the Manager process. You can defer this step until you are ready to run comparisons. To perform comparisons, Manager must be running.

```
START MANAGER
```

6. To confirm that Manager is running, issue the following command in GGSCI.

```
INFO MGR
```

## 4.4 Installing the C-Agent on a NonStop System

To install the agent on a NonStop SQL/MP system, the following steps are required:

■ Install the Oracle GoldenGate Veridata Agent files.

■ Copy `VSNSERV` to remote nodes if they contain table partitions.

■ Create a `GLOBALS` parameter file that contains specifications for:
  
  □ Locations of remote nodes where there is a `VSNSERV` process.
  
  □ A unique Manager name (if other Manager processes exist on the system)

■ Configure the Manager process.

### 4.4.1 Installing the Oracle GoldenGate Veridata Agent Files

1. Follow the steps in Section A, "Downloading Oracle GoldenGate Veridata" to download the Oracle GoldenGate Veridata Agent build file to a Windows workstation.

2. Using WinZip or an equivalent compression product, unzip the files to a temporary directory on your workstation.

3. Transfer the files in binary mode to the volume and subvolume on the NonStop Server where you want to install and run the agent. The agent software must be installed in a dedicated subvolume, including one that is separate from other Oracle GoldenGate software.
4. Alter the VERUNPAK to be an edit file by issuing the following TACL command.

   FUP ALTER VERUNPAK, CODE 101

5. Run the VERUNPAK macro by issuing the following TACL command.

   RUN VERUNPAK

6. At the prompt, verify the installation location. Type Y to confirm the location shown or N to select another location.

   Installing GoldenGate at $DATA.GoldenGate Veridata
   Is this correct? (Y/N) y
   UNPAK - File decompression program - T1255G06 - (2002-05-06)
   Archive version: 1
   File Mode RESTORE Program - T9074G07 (15JAN2002)
   Copyright Tandem Computers Incorporated 1981-2002
   Summary Information
   Files restored = 7  Files not restored = 0
   GoldenGate Veridata for Nonstop Installation
   Installs the GoldenGate Veridata Product
   Enter X at any prompt to quit.

7. You are prompted for a SQL catalog for the agent to use. Type the catalog name or type X for no catalog.

   SQL Catalog for Compilation (X for no catalog)? $data.cpscat
   SQL compiling VERIAGT
   GoldenGate Veridata Installation Complete.

8. Continue with Section 4.4.2, "Copying VSNSERV to Remote Nodes" and Section 4.4.3, "Creating a GLOBALS File" as necessary for your environment.

4.4.2 Copying VSNSERV to Remote Nodes

   If your tables have partitions on remote nodes, you will need to place a copy of the VSNSERV module on each of those nodes.

   If all of the remote nodes are the same hardware type, you can use a copy of the VSNSERV that is in the Oracle GoldenGate Veridata agent subvolume. Otherwise, you might need to download the correct agent build for that hardware type. It will include the correct VSNSERV.

   To place the VSNSERV on each node, you can do either of the following:

   ■ Install the entire Oracle GoldenGate Veridata Agent package on each of the remote nodes, even though the agent itself will not be running on them.

   ■ Copy the VSNSERV object to each of the remote nodes. To use this option, take the following steps.

   To copy VSNSERV to remote nodes

   1. Copy the appropriate VSNSERV program to each of the remote nodes.

   2. Log onto each remote node as a super user.

   3. Issue the following commands on each remote node:

      FUP GIVE vsnserv, SUPER.SUPER
      FUP secure vsnserv, "NNNN", PROGID

      ■ The first command sets the VSNSERV owner as SUPER.SUPER.
4. Specify the location of VSNSERV on each remote node by adding a HOST parameter for the node in the GLOBALS file that resides in the Oracle GoldenGate Veridata Agent installation directory. See Section 4.4.3, "Creating a GLOBALS File."

4.4.3 Creating a GLOBALS File

You need to create a GLOBALS file in the Oracle GoldenGate Veridata Agent directory if:

- Other Manager processes exist on this system, such as the one used by Oracle GoldenGate data synchronization software. A unique name for each Manager process must be specified in this file with the GGSPREFIX parameter, including the one that will be used by the Oracle GoldenGate Veridata agent.
- Partitions for tables that will be compared with Oracle GoldenGate Veridata are stored on remote nodes. The name of each node must be specified with the HOST parameter in the GLOBALS file.

To create a GLOBALS file

1. At the TACL prompt, issue the following command.

   EDIT GLOBALS

2. If prompted to create the file, enter Yes.

3. In the GLOBALS file, add one or both of the following parameters, depending on your environment:

   GGSPREFIX $aa
   HOST system_name [, GGSSUBVOL subvol] [, NODENUM node_number]
   [HOST system_name [, GGSSUBVOL subvol] [, NODENUM node_number]]

   - GGSPREFIX specifies a unique, two-character prefix that will be attached to the Manager process name, for example GGSPREFIX $GV.
   - HOST specifies the location of remote nodes where there is a VSNSERV component.

   **Note:** If you do not know the expand node number of a system, run SYSSINFO on that node.

   SYSSINFO - T9268H01 - (01 OCT 2004) SYSTEM \TEST Date 10 Jul 2008, 10:44:54
   Copyright 2003 Hewlett-Packard Development Company, L.P.

   System name \TEST
   EXPAND node number 110
   Current SYSnm SYS10
   System number 012345
   Software release ID H06.13.00

4. Save the file without a file extension. The file is stored in the subvolume where the agent resides. Do not move it.

4.4.4 Configuring Manager

1. From TACL, run the GGSCI program that is installed with the agent.

   RUN GGSCI
2. In GGSCI, issue the following command to create and edit a Manager parameter file.

```
EDIT PARAMS MRPARM
```

3. On the first line of the file, add the following parameter, where `number` is a unique port number that is not being used by any other process, including any Manager processes for other Oracle GoldenGate software.

```
PORT number
```

4. (Optional) On the next line, add the following parameter to specify a range of up to 256 ports that the Manager process can allocate dynamically. You can specify ports for concurrent processing threads if you will be running batch comparisons.

```
DYNAMICPORTLIST {port | port-port} [ , ...]
```

Where:

- To specify multiple ports, use a comma-delimited list, for example 7830, 7833.
- To specify a range of ports, use a dash (-) to separate the first and last port in the range, for example 7830-7835.
- To specify a range of ports plus an individual port, place a comma between the range and the individual port number, for example 7830-7835, 7839.

5. Save and close the file.

6. In GGSCI, issue the following command to start the Manager process. You can defer this step until you are ready to run comparisons. To perform comparisons, Manager must be running.

```
START MANAGER
```

7. To confirm that Manager is running, issue the following command in GGSCI.

```
INFO MGR
```
This chapter explains how to install a new, clean copy of the Oracle GoldenGate Veridata software for use with a MySQL database repository.

To upgrade an existing installation, see Chapter 12, "Upgrading Oracle GoldenGate Veridata."

This chapter includes the following sections:
Section 5.1, "Installation Prerequisites"
Section 5.2, "Installing the Oracle GoldenGate Veridata Software"

5.1 Installation Prerequisites

Before you run the installer you need to ensure that the MySQL database is installed and can store the internationalized object names that will be used.

5.1.1 Verify the Installed MySQL Database

Perform the following steps to verify that your MySQL database is installed.

1. Run the operating system’s command shell.
2. From the command shell, issue the following command to verify the MySQL installation. If the MySQL software is not installed on the Oracle GoldenGate Veridata host, the command can be executed on the host where MySQL is running, so long as you supply the network name or IP address of that host.

   mysqladmin -u admin_user -h hostname -P port
   -p admin_user_password ping

   Where:
   - There is no space between the -p and the password.
   - -h hostname is always required on UNIX and only required on Windows if the MySQL server is not running on the local machine. If the MySQL server is running on the local machine, use the loopback address of 127.0.0.1, because localhost has special meaning for MySQL and might not work with Oracle GoldenGate Veridata Server.
   - -P port is only required when the MySQL server is using a port other than the default of 3306.

3. The following message indicates a correct installation. Any other message indicates a problem that must be resolved before running the installer.

   mysqlld is alive.
5.1.2 Verify the Database Character Encoding

You must also ensure that the database can handle internationalized object names. For this, the repository database must:

- Be in the character set that is used for the table, column, and other object names that will be stored in it.
- Have been created in the correct character set prior to running the Oracle GoldenGate Veridata installer.

5.2 Installing the Oracle GoldenGate Veridata Software

1. Start the database that you are using for the Oracle GoldenGate Veridata repository.
2. Close all applications that use this database.
3. Follow the steps in Appendix A, "Downloading Oracle GoldenGate Veridata" to obtain the software.
4. Unzip the downloaded .zip file using WinZip or a similar compression program. If installing on a UNIX or Linux system, unzip the file on a Windows system and then FTP the file to the UNIX or Linux system in binary mode.
5. Run the installer.
   - As a wizard: Double click the GoldenGate_Veridata_platformrelease program (Windows) or use a GUI such as X Window (UNIX and Linux) and then follow the installation steps. The Windows wizard is shown in this documentation and contains additional steps not included in the UNIX or Linux versions.
   - As a shell process: From the command console of the operating system, issue the following command, and then follow the installation steps. The text prompts are similar to those shown on the wizard screens. To accept the default when using the command console, click Enter.
     - Windows:
       Start /wait GoldenGate_Veridata_platformrelease.exe -c
     - UNIX and Linux:
       GoldenGate_Veridata_platformrelease.sh –c
6. Welcome: Click Next to start the installation.
7. Destination Directory: Select an installation directory. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.
8. Data Location: Select a location for the user data. The data directory is where Oracle GoldenGate Veridata stores reports and other files that contain information about out-of-sync rows. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. This directory will contain data from the tables that are compared, so it should reside on a secure system.
9. Web Server Configuration: Accept the default shutdown and HTTP ports or specify different ones. The shutdown port is used by Oracle GoldenGate Veridata to shut down the server processes. The HTTP port is used by the embedded Tomcat web server application to connect to the Oracle GoldenGate Veridata Web component.

10. Create Veridata Users: Specify which of the following methods you want to use to create Oracle GoldenGate Veridata users:
   - Use the Tomcat Web Service administrative tool to manually create users after the installation.
   - Have the installation program import users from an existing Oracle GoldenGate Veridata installation. You will need to know the Oracle GoldenGate Veridata administrator user for that installation.

Whichever option you select, you can add users as needed after the installation using the Tomcat Web Service administrative tool.

11. Import Veridata Web: This screen is displayed if you elect to import users from an existing Oracle GoldenGate Veridata installation. Enter the installation directory and the administrator’s login user name and password.

12. Web Server Admin User: Specify an initial Oracle GoldenGate Veridata administrator user. The administrator is the most privileged role in Oracle GoldenGate Veridata and can perform all configuration, execution, and monitoring functions. Keep a record of this login. It is required to create other users after installation.

13. Veridata Repository: Select MySQL as the database that will be the Oracle GoldenGate Veridata repository.

14. Optional Software: To download and install a free trial version of MySQL for use as the Oracle GoldenGate Veridata repository, click the link to the MySQL website. Install MySQL and start it. You can leave the installer running while you do this.

15. Database Driver Configuration: Specify the name or IP address of the system where you are installing the Oracle GoldenGate Veridata server and web components. Specify the database port number, if it is not the MySQL default port number. If you use localhost as the host name, the installer will convert it to the IP address and alert you with a message.

16. Login Screens: The next screens vary, depending on your previous selection for the repository configuration.
   - Veridata Repository Admin:
     To create a new user ID or database (or both), you are prompted for the credentials of an existing database user ID that can be used to create the object(s). The user ID will be used only for this purpose and must have administrator or DBA privileges.
   - Veridata Repository:
     To create a new user ID, you are prompted to give it a name and password.
     To create a new database, you are prompted to give it a name.
     To use an existing user ID login, you are prompted for the login name and password.
     To use an existing database, you are prompted for its name. If the specified database already contains Oracle GoldenGate Veridata objects, you are
prompted to specify whether to drop and recreate them (removes existing objects) or to use them (preserves existing objects).

17. Veridata Repository, database option: Use any combination of these options to specify how the Oracle GoldenGate Veridata repository will be configured.

- User ID: Oracle GoldenGate Veridata requires a user ID to connect to the repository. Specify whether to create a new user ID for this purpose or to use an existing user ID login. Use an existing user ID if you already created one for Oracle GoldenGate Veridata, or if you are reinstalling Oracle GoldenGate Veridata to a different location from a previous one and want to retain the old repository. An existing user ID must have DML and DDL privileges for the Oracle GoldenGate Veridata repository.

- Database: Specify whether to create a new database for the repository or use an existing one.

18. (Windows only) Windows Services: It is recommended that you install Oracle GoldenGate Veridata as a Windows service. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up.

- Before you elect to install as a service, you must identify the user that will start the service. This logon will be granted full access to Oracle GoldenGate Veridata files. You can install the service to log on as the LocalSystem user, or you can specify a different user. If you specify a different user, that user must be granted LogonAsService privilege or the service will not start.

- When that is done, either accept the default service name or change it as required. If there are, or will be, other instances of this software on the local system, make certain that each service name is unique.

To prevent installation as a service, deselect the check box.

19. (Windows only) Windows Services User: If you elected to install Oracle GoldenGate Veridata as a service, this screen is displayed. Select Logon as Local System for Oracle GoldenGate Veridata to log on as LocalSystem when the service is started. If you select Logon as a specific user instead, enter the logon information of the user you wish to start the service.

Note: The user that starts the Windows service must have LogonAsService rights. If you specify a user that does not have these rights, the service will not start.

20. Start after install: Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.

21. (Windows only) Start Menu Folder: To add Oracle GoldenGate Veridata to the Programs list of the Start menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list. To make the shortcut available to everyone who uses this system, select Create Shortcuts for all users; otherwise, it will only be available to the current user.

To exclude Oracle GoldenGate Veridata from the Start menu, select Don't create a Start Menu folder. If you are installing this software as a service, it can be started from the installation folder or the Services control panel.
22. Information: Review your installation selections. Click **Back** to make changes or click **Next** to begin the installation.

23. Completing the Oracle GoldenGate Veridata Setup Wizard: To view the Oracle GoldenGate Veridata help system after the installation is done, accept the default. To prevent the help system from launching, clear the check box. The Oracle GoldenGate Veridata help system contains an introduction to the software, information about specific functions and tasks, and a detailed tour of the web interface. This tour can help both new and experienced users get started more quickly with a better understanding of the work flow and tools.

24. Click **Finish** to close the installer.
This chapter explains how to install a new, clean copy of the Oracle GoldenGate Veridata software for use with an Oracle database repository.

To upgrade an existing installation, see Chapter 12, "Upgrading Oracle GoldenGate Veridata."

This chapter includes the following sections:
Section 6.1, "Preinstallation System Check"
Section 6.2, "Installing the Oracle GoldenGate Veridata Software"

### 6.1 Preinstallation System Check

This check is only required if you are installing the server component to use the TNSNAMES connection method

1. Run the operating system's command line interface (UNIX/Linux command shell or Windows command console).
2. Execute one of the following commands.
   - **Windows**: Execute outside the Oracle installation bin directory:
     
     ```
     tnsping Oracle_instance
     ```
   - **UNIX/Linux**: Execute as follows:
     
     ```
     $ORACLE_HOME/bin/tnsping instance
     ```

   An OK on the last line of the output confirms that the Oracle installation files and DLLs are installed correctly. Any other result indicates a problem, which must be fixed before proceeding with these steps.

3. After a successful tnsping command, execute one of the following commands as an administrator user.
   - **Windows**:
     
     ```
     sqlplus admin_user/admin_user_password@instance
     ```
   - **UNIX/Linux**:
     
     ```
     sqlplus admin_user/admin_user_password@instance
     ```

   The result should be a SQL*Plus command prompt. If you do not receive the SQL*Plus command prompt, it means that you cannot connect to the database by using a connect string. This problem must be fixed before running the installer.
6.2 Installing the Oracle GoldenGate Veridata Software

1. Start the database that you are using for the Oracle GoldenGate Veridata repository.

2. Close all applications that use this database.

3. Follow the steps in Appendix A, "Downloading Oracle GoldenGate Veridata" to obtain the software.

4. Unzip the downloaded .zip file using WinZip or a similar compression program. If installing on a UNIX or Linux system, unzip the file on a Windows system and then FTP the file to the UNIX or Linux system in binary mode.

5. Run the installer.
   - As a wizard: Double click the GoldenGate_Veridata_platformrelease program (Windows) or use a GUI such as X Window (UNIX and Linux) and then follow the installation steps. The Windows wizard is shown in this documentation and contains additional steps not included in the UNIX or Linux versions.
   - As a shell process: From the command console of the operating system, issue the following command, and then follow the installation steps. The text prompts are similar to those shown on the wizard screens. To accept the default when using the command console, click Enter.
     - Windows:
       ```
       Start /wait GoldenGate_Veridata_platformrelease.exe -c
       ```
     - UNIX and Linux:
       ```
       GoldenGate_Veridata_platformrelease.sh -c
       ```

6. Welcome: Click Next to start the installation.

7. Destination Directory: Select an installation directory. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.

8. Data Location: Select a location for the user data. The data directory is where Oracle GoldenGate Veridata stores reports and other files that contain information about out-of-sync rows. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. This directory will contain data from the tables that are compared, so it should reside on a secure system.

9. Web Server Configuration: Accept the default shutdown and HTTP ports or specify different ones. The shutdown port is used by Oracle GoldenGate Veridata to shut down the server processes. The HTTP port is used by the embedded Tomcat web server application to connect to the Oracle GoldenGate Veridata Web component.

10. Create Veridata Users: Specify which of the following methods you want to use to create Oracle GoldenGate Veridata users:
    - Use the Tomcat Web Service administrative tool to manually create users after the installation.
- Have the installation program import users from an existing Oracle GoldenGate Veridata installation. You will need to know the Oracle GoldenGate Veridata administrator user for that installation.

Whichever option you select, you can add users as needed after the installation using the Tomcat Web Service administrative tool.

11. Import Veridata Web: This screen is displayed if you elect to import users from an existing Oracle GoldenGate Veridata installation. Enter the installation directory and the administrator’s login user name and password.

12. Web Server Admin User: Specify an initial Oracle GoldenGate Veridata administrator user. The administrator is the most privileged role in Oracle GoldenGate Veridata and can perform all configuration, execution, and monitoring functions. Keep a record of this login. It is required to create other users after installation.

13. Veridata Repository: Select Oracle as the database that will be the Oracle GoldenGate Veridata repository.

14. One of the following screens appears:

   - Oracle Database Home Location: This screen appears if there is only one Oracle instance installed, or if you are installing on a UNIX system where the ORACLE_HOME environment variable is defined. Verify that this Oracle installation location is the one that belongs to the database that will be the Oracle GoldenGate Veridata repository. If this is not the correct Oracle Home, click Cancel to exit the installer. Consult the database documentation for how to set the Oracle Home. After correctly setting the Oracle Home, run the installer again.

   - Oracle Database Configuration: If this screen appears, the installer was unable to locate a tnsnames.ora file. You can bypass the tnsnames.ora file by selecting EZCONNECT as the connection method, or you can click Browse to point the installer to the correct tnsnames.ora file location.

15. Database Driver Configuration: Specify the SQL*Net connection information to use.

   Note: Depending on whether you forced tnsnames or EZCONNECT on the previous screen (if it appeared in your case), you may be permitted to select only one of these options.

   Select Use TNS names to have Oracle GoldenGate Veridata get connection information from the tnsnames.ora file. The drop-down list displays all Oracle instances that are listed in the TNSNAMES.ORA file that is associate with the previously selected ORACLE_HOME. Make certain that the entry for this instance in the file is correct and that this same service name is configured in the listener.ora file. Make certain that TNSNAMES is one of the values in the NAMES.DIRECTORY_PATH parameter in the sqlnet.ora file on the database system. Many installation and connection problems can be traced to incorrect connection identifiers in these files.

   Note: On a UNIX system, if the installer is run from the command line instead of from a GUI, it may not display all of the Oracle instances in TNSNAMES.ORA, but only the first Oracle instance in the list. Type the correct instance name if it is not displayed.
Select **Use EZCONNECT** to have Oracle GoldenGate Veridata connect by using EZCONNECT. Then, supply the name or IP address of the database server (domain-qualified name, if required by the operating system), the database listening port number (default is 1521), and the instance name of the database. If connection errors occur, make certain that EZCONNECT is specified for the NAMES.DIRECTORY_PATH parameter in the sqlnet.ora file on the database system.

16. Veridata Repository, database option: Oracle GoldenGate Veridata requires a database user ID or schema to own and connect to the repository. Specify whether to **create a new user ID** for this purpose or to **use an existing user ID**. Use an existing user ID or schema if you already created one for Oracle GoldenGate Veridata, or if you are reinstalling Oracle GoldenGate Veridata to a different location from a previous one and want to retain the old repository. An existing user ID must have DML and DDL privileges for the Oracle GoldenGate Veridata repository.

17. Logon Screens: The next screens vary, depending on your previous selection for the repository configuration.

   - To create a new user:
     Veridata Repository Admin: Supply the login credentials of an existing database user that has DBA or system administrator privileges to create the Oracle GoldenGate Veridata user and schema.
     Click **Next** and then supply login credentials for the new Oracle GoldenGate Veridata user. The installer creates the new user with the necessary privileges to operate as Oracle GoldenGate Veridata.

   - To use an existing user:
     Supply the login credentials of the existing user, and then click **Next**.
     If the specified schema already contains Oracle GoldenGate Veridata objects, you are prompted to specify whether to drop and recreate it (removes existing objects) or to use it (preserves existing objects). Click **Next** after making a selection.

18. (New user only) Veridata Oracle Tablespace: Specify default and temporary tablespaces for the Oracle GoldenGate Veridata repository.

19. (Windows only) Windows Services: It is recommended that you install Oracle GoldenGate Veridata as a Windows service. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up.

   - Before you elect to install as a service, you must identify the user that will start the service. This logon will be granted full access to Oracle GoldenGate Veridata files. You can install the service to log on as the LocalSystem user, or you can specify a different user. If you specify a different user, that user **must** be granted LogonAsService privilege or the service will not start.

   - When that is done, either accept the default service name or change it as required. If there are, or will be, other instances of this software on the local system, make certain that each service name is unique.

To prevent installation as a service, deselect the check box.

20. (Windows only) Windows Services User: If you elected to install Oracle GoldenGate Veridata as a service, this screen is displayed. Select **Logon as Local System** for Oracle GoldenGate Veridata to log on as LocalSystem when the service
is started. If you select **Logon as a specific user** instead, enter the logon information of the user you wish to start the service.

---

**Note:** The user that starts the Windows service must have **LogonAsService** rights. If you specify a user that does not have these rights, the service will not start.

---

21. **Start after install:** Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.

22. **(Windows only) Start Menu Folder:** To add Oracle GoldenGate Veridata to the **Programs** list of the **Start** menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list. To make the shortcut available to everyone who uses this system, select **Create Shortcuts for all users**; otherwise, it will only be available to the current user. To exclude Oracle GoldenGate Veridata from the **Start** menu, select **Don't create a Start Menu folder**. If you are installing this software as a service, it can be started from the installation folder or the Services control panel.

23. **Information:** Review your installation selections. Click **Back** to make changes or click **Next** to begin the installation.

24. **Completing the Oracle GoldenGate Veridata Setup Wizard:** To view the Oracle GoldenGate Veridata help system after the installation is done, accept the default. To prevent the help system from launching, clear the check box. The Oracle GoldenGate Veridata help system contains an introduction to the software, information about specific functions and tasks, and a detailed tour of the web interface. This tour can help both new and experienced users get started more quickly with a better understanding of the work flow and tools.

25. Click **Finish** to close the installer.
This chapter explains how to install a new, clean copy of the Oracle GoldenGate Veridata software for use with a SQL Server database repository.

To upgrade an existing installation, see Chapter 12, "Upgrading Oracle GoldenGate Veridata."

This chapter includes the following sections:

Section 7.1, "Installing the Oracle GoldenGate Veridata Software"

### 7.1 Installing the Oracle GoldenGate Veridata Software

1. Start the database that you are using for the Oracle GoldenGate Veridata repository.
2. Close all applications that use this database.
3. Follow the steps in Section A, "Downloading Oracle GoldenGate Veridata" to obtain the software.
4. Unzip the downloaded .zip file using WinZip or a similar compression program.
5. Run the installer.
   - As a wizard: Double click the `GoldenGate_Veridata_platformrelease` program and then follow the installation steps. The wizard is shown in this documentation.
   - As a shell process: From the command console of the operating system, issue the following command, and then follow the installation steps. The text prompts are similar to those shown on the wizard screens. To accept the default when using the command console, press Enter.
     ```
     Start/wait GoldenGate_Veridata_platformrelease.exe -c
     ```
6. Welcome: Click Next to start the installation.
7. Destination Directory: Select an installation directory. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.
8. Data Location: Select a location for the user data. The data directory is where Oracle GoldenGate Veridata stores reports and other files that contain information about out-of-sync rows. You can accept the default (to be created during installation), type a path name, or click Browse to go to a different directory. This
directory will contain data from the tables that are compared, so it should reside on a secure system.

9. Web Server Configuration: Accept the default shutdown and HTTP ports or specify different ones. The shutdown port is used by Oracle GoldenGate Veridata to shut down the server processes. The HTTP port is used by the embedded Tomcat web server application to connect to the Oracle GoldenGate Veridata Web component.

10. Create Veridata Users: Specify which of the following methods you want to use to create Oracle GoldenGate Veridata users:

   ■ Use the Tomcat Web Service administrative tool to manually create users after the installation.
   ■ Have the installation program import users from an existing Oracle GoldenGate Veridata installation. You will need to know the Oracle GoldenGate Veridata administrator user for that installation.

Whichever option you select, you can add users as needed after the installation using the Tomcat Web Service administrative tool.

11. Import Veridata Web: This screen is displayed if you elect to import users from an existing Oracle GoldenGate Veridata installation. Enter the installation directory and the administrator’s login user name and password.

12. Web Server Admin User: Specify an initial Oracle GoldenGate Veridata administrator user. The administrator is the most privileged role in Oracle GoldenGate Veridata and can perform all configuration, execution, and monitoring functions. Keep a record of this login. It is required to create other users after installation.

13. Veridata Repository: Select SQL Server as the database that will be the Oracle GoldenGate Veridata repository.

14. Database Driver Configuration: Specify how Oracle GoldenGate Veridata connects to the SQL Server database. Click the Choose from predefined Data Sources radio button to enable a list of predefined ODBC data sources (DSNs). Click the Manually define a SQL Server connection radio button to open the following fields to define the new JDBC connection:

   ■ Host: The name of the computer on which SQL Server is running
   ■ Port: (optional) The port on which SQL Server is listening
   ■ Instance: (optional) The name of the SQL Server instance
   ■ Use Integrated Security: Check this box to use Windows SQL Server Integrated Security to log on

Contact your system administrator or consult the documentation for your operating system if you are unable to create a DSN. For basic instructions on creating the connection outside of the installer, see Appendix C, “Configuring an ODBC connection for a SQL Server Repository”.

15. Confirm the DSN connection information.

16. Veridata Repository, database option: Use any combination of these options to specify how the Oracle GoldenGate Veridata repository will be configured.

   ■ Login: Oracle GoldenGate Veridata requires a login to connect to the repository. Specify whether to create a new login for this purpose or to use an existing login. Use an existing login if you already created one for Oracle GoldenGate Veridata, or if you are reinstalling Oracle GoldenGate Veridata to
Installing the Oracle GoldenGate Veridata Software

Installing Using a SQL Server Repository

17. Login Screens: The next screens vary, depending on your previous selection for the repository configuration.

- **Veridata Repository Admin:**
  To create a new login or database (or both), you are prompted for the credentials of an existing database login that can be used to create the object(s). The login will be used only for this purpose and must have administrator or DBA privileges.

- **Veridata Repository:**
  To create a new login, you are prompted to give it a name and password.
  To create a new database, you are prompted to give it a name.
  To use an existing login, you are prompted for the login name and password.
  To use an existing database, you are prompted for its name. If the specified database already contains Oracle GoldenGate Veridata objects, you are prompted to specify whether to drop and recreate them (removes existing objects) or to use them (preserves existing objects).

18. Windows Services: It is recommended that you install Oracle GoldenGate Veridata as a Windows service. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up.

- Before you elect to install as a service, you must identify the user that will start the service. This logon will be granted full access to Oracle GoldenGate Veridata files. You can install the service to log on as the **LocalSystem** user, or you can specify a different user. If you specify a different user, that user **must** be granted **LogonAsService** privilege or the service will not start.

- When that is done, either accept the default service name or change it as required. If there are, or will be, other instances of this software on the local system, make certain that each service name is unique.

To prevent installation as a service, deselect the check box.

19. Windows Services User: If you elected to install Oracle GoldenGate Veridata as a service, this screen is displayed. Select **Logon as Local System** for Oracle GoldenGate Veridata to log on as **LocalSystem** when the service is started. If you select **Logon as a specific user** instead, enter the logon information of the user you wish to start the service.

**Note:** The user that starts the Windows service must have **LogonAsService** rights. If you specify a user that does not have these rights, the service will not start.

20. Start after install: Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.
21. Start Menu Folder: To add Oracle GoldenGate Veridata to the **Programs** list of the **Start** menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list. To make the shortcut available to everyone who uses this system, select **Create Shortcuts for all users**; otherwise, it will only be available to the current user.

To exclude Oracle GoldenGate Veridata from the **Start** menu, select **Don't create a Start Menu folder**. If you are installing this software as a service, it can be started from the installation folder or the Services control panel.

22. Information: Review your installation selections. Click **Back** to make changes or click **Next** to begin the installation.

23. Completing the Oracle GoldenGate Veridata Setup Wizard: To view the Oracle GoldenGate Veridata help system after the installation is done, accept the default. To prevent the help system from launching, clear the check box. The Oracle GoldenGate Veridata help system contains an introduction to the software, information about specific functions and tasks, and a detailed tour of the web interface. This tour can help both new and experienced users get started more quickly with a better understanding of the work flow and tools.

24. Click **Finish** to close the installer.

25. The SQL Server repository must be running before the Oracle GoldenGate Veridata Server Windows service can start. You can add the database as a startup dependency to the server service. Adding a dependency to a service requires a system administrator to modify its Registry entry. For more information, consult the Microsoft documentation or website.
This chapter explains how to set security for Oracle GoldenGate Veridata.

This chapter includes the following sections:

Section 8.1, "Oracle GoldenGate Veridata Security Overview"
Section 8.2, "Securing the Oracle GoldenGate Veridata Files"
Section 8.3, "Securing access to Oracle GoldenGate Veridata Web"
Section 8.4, "Maintaining Passwords"

8.1 Oracle GoldenGate Veridata Security Overview

When using Oracle GoldenGate Veridata, you will be selecting, viewing and storing data values from the tables or files of your business applications. Care must be taken to protect access to the following components:

- The files, programs, and directories in the Oracle GoldenGate Veridata installation directories
- Data files that contain the results of data comparisons
- The Oracle GoldenGate Veridata Web User Interface, where data values can be viewed

8.2 Securing the Oracle GoldenGate Veridata Files

This section describes how to secure your business data and control access to the Oracle GoldenGate Veridata installation directories and user interface.

8.2.1 Controlling Access to the Installation Directories

Standard operating system permissions apply to the programs, files, and directories within the Oracle GoldenGate Veridata Server and Web User Interface, and Oracle GoldenGate Veridata Agent installation directories. You should adjust the permissions for these objects based on your business security rules.

8.2.2 Securing files that Contain Business Data

Oracle GoldenGate Veridata Server creates data files that will contain sensitive application data. By default, these files reside in the shared/data directory within the Oracle GoldenGate Veridata Server installation directory, but the person who installed the software might have installed them in a different location. All of the sub-directories within that directory contain files that may reflect business data.
The types of files that contain sensitive data are:

- The comparison report (rpt sub-directory)
- The out-of-sync report (oosxml and oos sub-directories)

These files inherit the same file permissions as those of the user that runs the Oracle GoldenGate Veridata Server installation program. Do not change the permissions, or Oracle GoldenGate Veridata may be unable to maintain them. These files should be kept just as secure as you would keep your business data. Users of Oracle GoldenGate Veridata Web do not require access to these files because they see the same information through the client interface.

### 8.3 Securing access to Oracle GoldenGate Veridata Web

You can assign security roles to the users of Oracle GoldenGate Veridata to control their access to the functions that are performed by the software, some of which expose selected data values from the database. These roles are:

- **Administrator**: The administrator role is the highest-level security role in Oracle GoldenGate Veridata. This role can perform all of the functions that configure, execute, and monitor Oracle GoldenGate Veridata.

- **PowerUser**: The power user role is the second-highest role in Oracle GoldenGate Veridata. This role can perform all of the functions that configure, execute, and monitor Oracle GoldenGate Veridata from the Oracle GoldenGate Veridata Web User Interface, but this role cannot perform any configuration functions for the Oracle GoldenGate Veridata Server.

- **ReportViewer**: The report viewer role cannot perform functions that configure Oracle GoldenGate Veridata or execute jobs. This role can only view configuration and job information, and view comparison reports.

- **DetailReportViewer**: The detail report viewer role cannot perform any functions that configure Oracle GoldenGate Veridata or execute jobs. This role can only view configuration and job information, and view comparison reports and out-of-sync report information through the Oracle GoldenGate Veridata Web User Interface or at the file level.

Security is controlled through the Administration Tool of the Apache Tomcat Web Server. From this interface, a user with the administrator role can:

- Create a user and assign it a security role.
- Create user groups and assign them security roles. Users can be added to these groups without being given a security role. A user inherits the role of its group.
- Create a user and assign it a security role, and then add that user to a group. The user inherits the role of its group and keeps its individual role.

---

**Note**: You should back up the `conf` directory prior to changing the server configuration or Tomcat users.

---

**To open the Apache Tomcat Web Server Administration Tool**

1. Connect to the Apache Tomcat Web Server from a browser by typing the following address:

   "http://hostname:port/admin"

   Where:
Maintaining Passwords

hostname is the name or IP address of the system where the Oracle GoldenGate Veridata server and web components are hosted, and port is the port number assigned to Oracle GoldenGate Veridata Server (default is 8830).

2. Log on to the Apache Tomcat Web Server Administration Tool as an Oracle GoldenGate Veridata administrator user. A default administrator user was created during the installation of Oracle GoldenGate Veridata.

3. In the navigation pane, click to expand User Definition. From this node, all user resources are managed.

To create a group
1. Under User Definition, click Groups. Existing groups are displayed and can be edited by clicking their names.

2. From the Group Actions list, select Create New Group.

3. Under Group Properties, type a name for the group (no spaces, case sensitive) in the Group Name box and (optional) a description in the Description box.

4. Under Role Name, select the check box next to the role you want to assign to the group. You can select any role that is listed in Section 8.3, "Securing access to Oracle GoldenGate Veridata Web".

5. Click Save to save the group.

6. When finished using the Apache Tomcat Web Server Administration Tool, click Commit Changes to save the changes to the repository. To make any more changes after you commit the changes, you must log in again.

To create or edit a user
1. Under User Definition, click Users.

2. To edit a user, click its name. To add a new user, select Create New User from the User Actions list.

3. Under User Properties, type:
   - User Name: A name for the user (no spaces, case-sensitive)
   - Password: A password for the user (no spaces, case-sensitive).
   - Full Name: (Optional) the name of the person who is this user.

4. To assign the user to a group, click the check box next to the name of the group under Group Name. Linking a user to a group is optional. The user inherits the default role of the group.

5. To assign a role to this user, click the check box next to the name of the role under Role Name. You can select any role that is listed in Section 8.3, "Securing access to Oracle GoldenGate Veridata Web."

6. Click Save to save the user.

7. When finished using the Apache Tomcat Web Server Administration Tool, click Commit Changes to save the changes to the repository. To make any more changes after you commit the changes, you must log in again.

8.4 Maintaining Passwords

You can change the passwords that allow users access to the Veridata Web User Interface and those that access the repository.
8.4.1 Changing User Passwords

User passwords can be changed in the Oracle GoldenGate Veridata Web User Interface by selecting the Change Password option of the Options/Settings menu item. Refer to the online Help for more information.

8.4.2 Changing the Repository Password

A valid repository database password must be stored in Oracle GoldenGate Veridata to allow it to access the database. This repository database password is initially set based on user entries during the installation. It is stored in an Oracle Wallet created by the installation program.

If the repository database password changes after installation, the vericom program is used to change the corresponding password stored in Oracle GoldenGate Veridata.

1. Before you begin you must know the changed password that is currently valid for the database of the repository.
2. Navigate to the Oracle GoldenGate Veridata installation directory.
3. Enter one of the following commands to request a change to the repository password currently stored in Oracle GoldenGate Veridata:
   - On Windows:
     Shell> vericom.bat -reset_password [password]
   - On UNIX or Linux:
     Shell> vericom.sh -reset_password [password]
4. If you do not enter a password on the command line, you will be prompted to enter one once vericom starts.
5. After you enter the password, vericom connects to the database to verify the password is valid.
6. To activate the change, bring down the Oracle GoldenGate Veridata Web User Interface and restart.

See Section 10.2 for more information on running vericom.
This chapter explains how to run the Oracle GoldenGate Veridata programs, such as the agents and Java components.

This chapter includes the following sections:

Section 9.1, "Starting and Stopping the C-agent and Manager"
Section 9.2, "Starting and Stopping the Java-Based Components"
Section 9.3, "Reloading Log Information"
Section 9.4, "Connecting to Oracle GoldenGate Veridata Web"

9.1 Starting and Stopping the C-agent and Manager

The C-agent starts automatically at the request of Oracle GoldenGate Veridata Server when initiating comparisons. However, for Oracle GoldenGate Veridata Agent to function correctly, the following must be running:

- The database to which the agent is linked.
- The Manager process for the C-agent.

Although the agent process itself is an automatic process, you can stop the Manager process that controls the agent. Stopping Manager prevents Oracle GoldenGate Veridata Server from being able to start a new agent process, but it does not stop agents that are already running.

To control the C-agent Manager on all platforms

1. From the Oracle GoldenGate Veridata Agent installation location, run GGSCI.
2. In GGSCI, issue the appropriate command as follows to stop or start the Manager.

   START MANAGER
   Or...
   STOP MANAGER

To control the C-agent Manager as a Windows service

If Manager is installed as a Windows service, it either starts at system boot time or must be started manually, depending on how the service is configured. The installation default is to start automatically at system boot time. To start the service manually, use the START MANAGER command in GGSCI or start the service in the Services control panel applet.
To change the startup behavior of the service, right click the name in the Services control panel, and then select Properties.

When Manager is installed as a Windows service, you can stop it in the Services control panel applet or with the STOP MANAGER command in GGSCI.

### 9.2 Starting and Stopping the Java-Based Components

The Oracle GoldenGate Veridata Server and Oracle GoldenGate Veridata Web components are Java-based programs. The Oracle GoldenGate Veridata Agent component is also available as a Java program for all platforms except NonStop.

Note: Before starting the server and web processes, start the repository database.

#### 9.2.1 Controlling the Java-Based Components from the Command Line

To control the agent component, change directories to its installation directory and issue the appropriate command as follows:

<table>
<thead>
<tr>
<th>UNIX and Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent.sh {start</td>
<td>run)</td>
</tr>
<tr>
<td>Or...</td>
<td>Or...</td>
</tr>
<tr>
<td>agent.sh stop</td>
<td>agent.bat stop</td>
</tr>
</tbody>
</table>

Where:

- **run** starts the agent in the same command window from which it is launched.
- **start** starts the agent in a separate command window.

Note: The run option is useful for diagnosing errors that happen during the startup process before the agent error logging is configured. When the run option is used, messages written stdout and stderr appear in the command window. The agent normally logs its messages to the log file, so only operating system messages and logging system errors are written to stderr. When the start option is used, messages written to stdout and stderr are discarded.

To control the web component, change directories to the web/bin directory of the installation directory, and issue the appropriate command as follows:

<table>
<thead>
<tr>
<th>UNIX and Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>veridata_web.sh start</td>
<td>veridata_web.bat start</td>
</tr>
<tr>
<td>Or...</td>
<td>Or...</td>
</tr>
<tr>
<td>veridata_web.sh stop</td>
<td>veridata_web.bat stop</td>
</tr>
</tbody>
</table>

Configure the host to start and stop the processes automatically. Contact your system administrator if you need assistance.
9.2.2 Controlling the Java-Based Components as Windows Services

An Oracle GoldenGate Veridata Windows service either starts at system boot time or must be started manually, depending on how it is configured. The installation default is to start automatically at system boot time. Windows services are controlled from the Services control panel applet or the Start menu. To change the startup behavior of a service, right click the name in the Services control panel, and then select Properties.

When installed as a service, an Oracle GoldenGate Veridata Java component can be stopped from the Services control panel, from the Start menu, or from the command console.

To control the Java components from the Windows Start Menu

1. In the Start menu, select Programs, and then navigate to the folder that contains the Oracle GoldenGate Veridata shortcuts.
   - To start the web component, select Application, then Start Veridata Web
   - To stop the web component, select Application, then Stop Veridata Web.
   - To start the agent component, select Application, then Start Veridata Agent
   - To stop the agent component, select Application, then Stop Veridata Agent

9.3 Reloading Log Information

You can reload logging information from the log4j.properties file to a running agent. The changes in the log4j.properties file are put into effect on the agent. The agent must be running for this command to work.

Use the following command:

<table>
<thead>
<tr>
<th>UNIX and Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent.sh reloadlog</td>
<td>agent.bat reloadlog</td>
</tr>
</tbody>
</table>

9.4 Connecting to Oracle GoldenGate Veridata Web

You can connect to Oracle GoldenGate Veridata Web in either of the following ways.

- If there is a Start menu shortcut for Oracle GoldenGate Veridata on the system where the server and web components are installed, navigate to it and select Launch Oracle GoldenGate Veridata Web.
- Run a web browser on any system in your network and type the following address:
  
  http://hostname:port/veridata

Where:

hostname is the name of the system where Oracle GoldenGate Veridata Server is installed and port is the port number where it is running (default is 8830). Use localhost as the host name if connecting on the system that is local to the server installation.

Examples:

- http://localhost:8830/veridata
- http://sysa:8830/veridata
The Oracle GoldenGate Veridata Web login page is displayed upon successful connection. Log in with your user name and password. For full instructions on using Oracle GoldenGate Veridata Web, see the online help.
This chapter explains how to use the vericom command line interface to run comparisons.

This chapter includes the following sections:

Section 10.1, "Overview of the Command Line Interface"
Section 10.2, "Running vericom"
Section 10.3, "Vericom exit statuses"
Section 10.4, "Vericom Output Examples"

10.1 Overview of the Command Line Interface

You can use the vericom tool of Oracle GoldenGate Veridata to execute certain comparison tasks from the command shell of the operating system. The vericom tool runs the Oracle GoldenGate Veridata Command Line Interface and enables you to handle these activities with automated programs.

You can:

■ Run an entire job or a specific compare pair of a job.

Note: You cannot run a group individually.

■ Change the password that Veridata uses to connect to the repository
■ Set tracing (only under guidance of an Oracle Support analyst)

For specific compare pairs, you can:

■ Review previous out-of-sync results
■ Generate out-of-sync XML from the previous run
■ Override the same profile and row partition settings that are possible from the web interface

Comparisons also can be run from the Oracle GoldenGate Veridata Web interface. This interface provides greater control in configuring the objects to be compared and for controlling runtime parameter settings.
10.2 Running vericom

The vericom program can be run by anyone who has the correct operating system permissions to run it.

1. On the system where the Oracle GoldenGate Veridata is installed, run the command shell of the operating system.
2. Navigate to the Oracle GoldenGate Veridata installation directory.
3. Use the following syntax to run the vericom program.

Syntax
vericom{.bat|.sh} required_input [optional_input]

Required Input
One of the following are required; otherwise an error is returned. Enter only one option.
- help | -helprun |
{-version | -v} |
{-job | -j} job |
- reset_password [password]

If -version, -v, -help, or -helprun are specified, they take precedence over any other flag specified.

Optional Input
This is the optional input:
[ -g group -c compare_pair ]
[ -nw ]
[ -rP profile ]
[ -rR ]
[ -r0 ]
[ -rN threads ]
[ -rD seconds ]
[ -rC | +rC ]
[ -rOb | -rOx | -rO2 | -rO0 ]
[ -r0s records ]
[ -rT1 ]
[ -rTc ]
[ -rTMs trace_number ]
[ -pS source_partition_name | -pSg source_sql_predicate | -pSA1 source_ascii_start_key | -pSA2 source_ascii_end_key | -pSH1 source_hex_start_key | -pSH2 source_hex_end_key ]
[ -pT target_partition_name | -pTq target_sql_predicate | -pTA1 target_ascii_start_key | -pTA2 target_ascii_end_key | -pTH1 target_hex_start_key | -pTH2 target_hex_end_key ]
[ -pq sql_predicate ]
[ -rd0 | -rdN run_ID ]
### Table 10–1  Vericom Runtime Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-help</code></td>
<td>Displays the <code>vericom</code> syntax components and their descriptions.</td>
</tr>
<tr>
<td><code>-helprun</code></td>
<td>Displays run-related syntax components and their descriptions.</td>
</tr>
<tr>
<td>`(-version</td>
<td>-v)`</td>
</tr>
<tr>
<td>`(-job</td>
<td>-j) job`</td>
</tr>
<tr>
<td><code>-reset_password [password]</code></td>
<td>Requests a change to the database repository password that is stored in Oracle GoldenGate Veridata.</td>
</tr>
<tr>
<td><code>-g group</code></td>
<td>Specifies a group and compare pair. For group and compare_pair, specify the names that were assigned when these objects were created in Oracle GoldenGate Veridata Web.</td>
</tr>
<tr>
<td><code>-c compare_pair</code></td>
<td>Specifies a group and compare pair. For group and compare_pair, specify the names that were assigned when these objects were created in Oracle GoldenGate Veridata Web.</td>
</tr>
<tr>
<td><code>-nw</code></td>
<td>Directs <code>vericom</code> not to wait for the job to finish before returning the prompt. Instead, <code>vericom</code> returns immediately after starting a job.</td>
</tr>
<tr>
<td><code>-rP profile</code></td>
<td>Overrides the profile that is defined for a job. For profile, specify the name that was assigned when the profile was created in Oracle GoldenGate Veridata Web.</td>
</tr>
<tr>
<td><code>-rR</code></td>
<td>A run override option. Compares only those rows that were out-of-sync in the previous run, based on the information that is stored in the out-of-sync file. The results identify which rows were brought back into synchronization by replication or another method.</td>
</tr>
<tr>
<td><code>-rO</code></td>
<td>A run override option. Generates an OOSXML file based on the out-of-sync file from the previous run. It generates XML for every row that is in the file. You can use the XML to view the out-of-sync information in an XML editor or for other purposes.</td>
</tr>
<tr>
<td><code>-rN threads</code></td>
<td>Specifies the number of concurrent comparison threads to use. You can use as many threads as there are processors on the server system. This option overrides the default job profile and has no effect if a job is not run with <code>-j</code> or if just one comparison is run by using <code>-j</code> with <code>-g</code> and <code>-c</code>.</td>
</tr>
<tr>
<td><code>-rD seconds</code></td>
<td>Delays the confirmation step by the specified number of seconds to account for replication lag. Delaying the confirmation step reduces the number of false out-of-sync results that occur because an updated source value was not replicated fast enough. This option overrides the default job profile and has no effect if the <code>-rR</code> option is used.</td>
</tr>
</tbody>
</table>
### Controls whether or not the confirmation step (confirm OOS) is performed in the job.

- `-rC` skips the confirmation step. You can skip the confirmation step if activity on the source tables is stopped or if replication is not continuously updating the target table(s).
- `+rC` includes the confirmation step.

These options override the default job profile and are mutually exclusive. They have no effect unless `-j` is used.

### Controls the kind of file that is produced for the out-of-sync report.

- `-rOb` generates binary format that is compatible with the Oracle GoldenGate Veridata Web browser.
- `-rOx` generates output in XML.
- `-rO2` generates both binary and XML output.
- `-rO0` suppresses out-of-sync output.

These options override the default job profile and are mutually exclusive. They have no effect if `-rR` is used.

### Limits the number of out-of-sync rows that are written to a chunk of the OOSXML file. Writing the file in chunks prevents it from becoming too large for the system to manage and allows periodic archiving or purging. The current file is closed when the specified number of rows is written, and a new file is opened. This option overrides the default job profile and has no effect if `-rR` is used.

### Turns on tracing of Oracle GoldenGate Veridata Agent for the initial comparison step. Do not use without the guidance of an Oracle support analyst.

### Turns on tracing of Oracle GoldenGate Veridata Agent for the confirmation step. Do not use without the guidance of an Oracle support analyst.

### Turns on tracing for Oracle GoldenGate Veridata Server. `trace_number` is a bitmask of server execution trace flags. The higher the level, the more detailed the trace data. Do not use without the guidance of an Oracle support analyst.
Running Comparisons from the Command Line

Table 10-1  (Cont.) Vericom Runtime Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-pS source_partition_name</td>
<td>Runs the comparison using an existing source row partition or using an override partition that is defined by partition criteria. These options are mutually exclusive. They are valid only if comparing one compare pair (-j with -g and -c) and are ignored otherwise.</td>
</tr>
<tr>
<td>-pSq source_sql_predicate</td>
<td>Specifies an existing source partition that is already defined and stored in the repository. The partition name is not validated and is passed directly to Veridata Server. There will be an error if the specified partition does not exist.</td>
</tr>
<tr>
<td>-pSA1 source_ascii_start_key</td>
<td>Specifies an ASCII key as the starting key value of a partition that overrides an existing source partition for an Enscribe file.</td>
</tr>
<tr>
<td>-pSA2 source_ascii_end_key</td>
<td>Specifies an ASCII key as the ending key value of a partition that overrides an existing source partition for an Enscribe file.</td>
</tr>
<tr>
<td>-pSH1 source_hex_start_key</td>
<td>Specifies a hexadecimal key as the starting key value of a partition that overrides an existing source partition for an Enscribe file.</td>
</tr>
<tr>
<td>-pSH2 source_hex_end_key</td>
<td>Specifies a hexadecimal key as the ending key value of a partition that overrides an existing source partition for an Enscribe file.</td>
</tr>
<tr>
<td>-pT target_partition_name</td>
<td>These options specify target partitions and have the same rules as the corresponding options that specify source partitions.</td>
</tr>
<tr>
<td>-pTq target_sql_predicate</td>
<td>Specifies a SQL predicate to be used for both the source and target SQL tables, as an override to existing partitions. This option has the same rules as -pSq source_sql_predicate and -pTq target_sql_predicate.</td>
</tr>
</tbody>
</table>
10.3 Vericom exit statuses

Vericom exits with one of the following statuses. This examples shown are for a UNIX or Linux system.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command executed successfully. If a job was run, it finished with all rows in-sync. If -nw was specified, the exit status is 0 if the job started successfully.</td>
</tr>
<tr>
<td>1</td>
<td>Invalid vericom syntax was used. For example, the following are invalid: vericom.sh -helptun (Typographical error.) vericom.sh -j -g group1 (The name of the job is missing.)</td>
</tr>
<tr>
<td>2</td>
<td>Vericom could not find connection information by looking in veridata.loc or veridata.cfg, and connection parameters were not specified with command arguments.</td>
</tr>
<tr>
<td>3</td>
<td>Provides more granularity for input errors that involve flags that run comparisons. For example, the following mistakes will cause this error: vericom.sh -j job1 -c address=address In the preceding example, the -g group input is missing. It is required with -j if -c is used. vericom.sh -j job1 -g group1 -rd0 In the preceding example, the -rd0 flag requires -c because delta processing applies at the compare pair level.</td>
</tr>
<tr>
<td>4</td>
<td>The job ran successfully, but there were rows that had a comparison status of something other than in-sync.</td>
</tr>
<tr>
<td>5</td>
<td>There was a communication error with Oracle GoldenGate Veridata Server.</td>
</tr>
</tbody>
</table>

10.4 Vericom Output Examples

To view the results of a comparison that is run with vericom, you can use Oracle GoldenGate Veridata Web to view the comparison report (see Section 1.3, "Viewing Comparison Results"), and you can view the output that is returned by vericom to the terminal. If a run finishes successfully, statistics for the job are displayed.
Example 1
The following example shows a run on a Windows system of TestJob without specifying -w. The process exits with status 0, and finished job statistics are not displayed.

C:\veridata\server\bin>vericom.bat -j TestJob
Connecting to: localhost:9177
Run ID: (2256, 0, 0)
C:\veridata\server\bin> if errorlevel 0 echo EXITED 0 STATUS
EXITED 0 STATUS

Example 2
The following example shows a run of TestJob with -w specified. The process exits with status 4 because one of the compare pairs had a validation error. Finished job statistics are displayed.

C:\veridata\server\bin>vericom.bat -j TestJob -w
Connecting to: localhost:9177
Run ID: (2257, 0, 0)
Job Start Time: 2008-03-21 22:48:05
Job Report Filename: C:\\testjunit\\rpt\\TestJob\\00002257\\TestJob.rpt
Number of Compare Pairs: 3
Number of Compare Pairs With Errors: 1
Number of Compare Pairs With OOS: 1
Number of Compare Pairs With No OOS: 1
Number of Compare Pairs Cancelled: 0
Job Completion Status: WITH ERRORS
C:\veridata\server\bin> if errorlevel 4 echo EXITED 4 STATUS
EXITED 4 STATUS

Example 3
The following example shows a run of compare pair TABLE9=TABLE9 in job TestJob with -w specified. The process exits with status 0 because the tables are in-sync. Finished job statistics are displayed.

C:\veridata\server\bin>vericom.bat -j TestJob -g TestGroup -c TABLE9=TABLE9 -w
Connecting to: localhost:9177
Run ID: (2258, 0, 0)
Job Start Time: 2008-03-21 22:51:08
Job Stop Time: 2008-03-21 22:51:11
Job Report Filename: C:\\veridata\\data\\rpt\\TestJob\\00002258\\TestJob.rpt
Number of Compare Pairs: 1
Number of Compare Pairs With Errors: 0
Number of Compare Pairs With OOS: 0
Number of Compare Pairs With No OOS: 1
Number of Compare Pairs Cancelled: 0
Compare Pair Report Filename: C:\veridata\data\rpt\TestJob\00002258\TestGroup\CP_TABLE9=TABLE9.rpt
Number of Rows Compared: 21
Number of Rows In Sync: 21
Number of Rows With Errors: 0
Number of Rows Out Of Sync: 0
Number of Inserts Out Of Sync: 0
Number of Deletes Out Of Sync: 0
Number of Updates Out Of Sync: 0
Compare Pair OOSXML Directory: C:\veridata\data\oosxml\TestJob\00002258\TestGroup
Compare Pair OOSXML Filename:
Job Completion Status: IN SYNC
C:\veridata\server\bin> if errorlevel 0 echo EXITED 0 STATUS
EXITED 0 STATUS

On UNIX systems, the exit status of the previously executed command is in the special variable '$?' if you are using SH or KSH shells. If you are using the CSH shell, the exit status of the previously executed command is in the special variable '$status'. 
This chapter describes parameters that adjust different aspects of the sort memory configuration when using server-side sorting.

This chapter includes the following sections:

Section 11.1, "Overview of the Server Memory"
Section 11.2, "Estimating Memory Usage"
Section 11.3, "How to Set a Parameter"
Section 11.4, "Parameter Descriptions"

11.1 Overview of the Server Memory

Oracle GoldenGate Veridata Server uses virtual memory in the following ways:

- **Server memory for basic operation.** This is the amount of virtual memory that the Veridata server and web components need simply to operate. It stores object pools, database access libraries, and other information. This is usually about 200 MB.

- **Sort memory.** This is the memory that is used when server-side sorting is used. The virtual memory for sorting is allocated for the entire comparison, not per thread. The rows are read from the agent and submitted to be sorted. The sorting occurs in a thread that is separate from the thread that reads from the agent, and the sort may use more threads to work in parallel. Once all of the rows from the agent are submitted to the sort process, the server process retrieves the sorted rows from the sort for comparison.

- **Row hash queue memory.** This is the memory that buffers data between the agent processes, the sort process, and the server process. A comparison that uses database sorting requires a single queue each for the source and target. Each queue has a capacity of 20 MG. The memory usage by the queues is affected by the relative speed of the comparison and by the data coming from the agent. The relative speed between the two agents also affects the memory usage. A larger differential in speed increases the amount of memory that is used, because the queue needs to buffer the data.

- **MOOS queue memory.** This is the memory that holds potentially out-of-sync records between the initial comparison and confirmation steps of a comparison. The size of the MOOS queue is limited to 50K of records. Memory usage is also dependent on the width of each record.

- **IPC buffer memory.** This is the memory that is used to exchange messages between the server and the agent.
■ **Scratch runtime transient memory.** This is virtual memory space.

The amount of memory that can be used by the sort process cannot be greater than the minimum of:

■ System physical memory
■ Available memory in swap
■ Java boot option \(-Xmx\) maximum memory setting

### 11.2 Estimating Memory Usage

The maximum amount of memory available to Oracle GoldenGate Veridata is specified by the Java boot option \(-Xmx\). When server-side sorting is used, a large portion of this memory is reserved for sorting during comparisons. This reserved amount is controlled by the `server.max_sort_memory` configuration parameter.

When a comparison is run, two buffers are allocated from the reserved sort memory. Each of these is equal to the size specified as **Maximum Memory Usage (MB)**. To access this setting click the **Edit** option from the Profile Configuration screen, then **Sorting Method** from the Profile settings categories.

**To Estimate Memory based on the Number of Concurrent Comparisons**

The maximum amount of memory that can be used for any comparison is set by the parameter `server.max_comparison_sort_memory`. The \(-Xmx\) Java boot option should be set large enough to allow the desired number of concurrent comparisons.

The maximum number of concurrent comparisons is defined by the `server.max.concurrent_comparison_threads` configuration parameter. Therefore the maximum amount of sort memory can be as large as:

\[
\text{server.max_comparison_sort_memory} \times \text{server.max.concurrent_comparison_threads}
\]

For example, if you set `server.max.concurrent_comparison_threads` to allow 10 concurrent comparisons and leave `server.max_comparison_sort_memory` set to the default value of 100 MB, you will need 1 GB of available memory.

**To Estimate the Amount of Memory Used per Row**

Refer to Section 2.4.2, "Disk and Memory Requirements for the Server Component" for the calculation to estimate the amount of memory used per row.

### 11.3 How to Set a Parameter

To set a parameter, edit its entry in the `veridata.cfg` file. This file is stored in the `shared/conf` directory within the Oracle GoldenGate Veridata Server installation directory.

---

**Note:** You should back up the `conf` directory prior to changing the server configuration or Tomcat users.

---

Open an Oracle service request before changing these parameters. For more information, go to [http://support.oracle.com](http://support.oracle.com).
11.4 Parameter Descriptions

This section describes the parameters that can be set in the `veridata.cfg` file.

11.4.1 server.max_sort_memory

Sets the maximum amount of sort virtual memory that is available to all running comparisons that use server-side sorting. The default amount is the Java boot option `-Xmx` maximum memory setting less the 200 MB needed for basic tasks. You can limit this amount to make more memory available for the Oracle GoldenGate Web User Interface.

If a comparison cannot get enough virtual memory when it starts, it fails or tries again depending on the setting of the `server.sort_waiting_threshold` parameter. If a comparison does get enough virtual memory, the currently available sort virtual memory gets decremented by the amount that the comparison reserves. When a comparison completes, it increments the amount of available sort virtual memory by the amount of sort virtual memory that it had reserved.

**Syntax**

```
server.max_sort_memory {default | number{M | m}}
```

- `default` allows Oracle GoldenGate Veridata to define a maximum value that is dependent on the operating system.
- `number{M | m}` specifies a value in megabytes. For example, `1000M` means a limit of 1000 megabytes. If this number exceeds the amount of available memory, the value will be reduced to the amount of available memory.

11.4.2 server.max_concurrent_comparison_threads

Sets the maximum number of concurrent comparisons that can be executed. In general, the amount configured by the server is the optimal value, given the machines resources. You can lower this number to reduce the impact of the server on your system. When this limit is reached, no new comparisons will start until an active comparison completes.

**Syntax**

```
server.max_concurrent_comparison_threads {default | number}
```

- `default` allows Oracle GoldenGate Veridata to compute the maximum number of concurrent threads based on the `server.max_comparison_sort_memory` and available resources. The default value is the `server.max_sort_memory` divided by `server.max_comparison_sort_memory`.
- `number` is a positive integer that sets the maximum number of concurrent comparison threads.

11.4.3 server.max_comparison_sort_memory

Sets the maximum amount of virtual memory that can be reserved by a single comparison that is using server sorting. A comparison that tries to reserve more than this amount will fail, even if it is configured to retry.

The amount set for this property limits the amount that can be set in the Oracle GoldenGate Web User Interface Profile.
If the value of this parameter is changed, it might cause the concurrent comparisons that use server-side sorting to fail or retry, depending on the sorting parameters defined in the profile that is being used.

**Syntax**

server.max_comparison_sort_memory {default | number M | m}

- **default** is 100 MB.
- **number**{M | m} specifies a value in megabytes. For example, 600M sets the limit for any comparison to be 600 megabytes.

### 11.4.4 server.sort_waiting_threshold

Sets a threshold above which new jobs will fail if there are too many comparisons waiting to retry a virtual memory reservation.

**Syntax**

server.sort_waiting_threshold {default | number}

- **default** is zero; if memory cannot be allocated the job will not be allowed to run.
- **number** sets the number of comparisons that can be waiting before new jobs will fail. For example, a value of 100 means that if there are 100 comparisons waiting to retry a memory reservation, and there is a job that is ready to begin execution, this job will not be allowed to run.
This chapter describes how to upgrade to a new release of the Oracle GoldenGate Veridata Server and Web User Interface components.

This chapter includes the following sections:

Section 12.1, "Upgrading the Server and Web User Interface"
Section 12.2, "Upgrading to a New Release of the Agent"

12.1 Upgrading the Server and Web User Interface

You must upgrade all components of Oracle GoldenGate Veridata. The use of different releases of the server and agent components is not supported.

12.1.1 Upgrade Support Overview

The upgrade procedure varies depending on your current version:

- **Upgrades from version 1 to later versions**: There is no upgrade path from version 1 to version 2 and later versions. You must recreate your configuration objects in the new release. You can install the new release, and then run the old and new releases side-by-side, so that you can refer to the existing configurations as you recreate them in the new installation.

- **Upgrades from version 2 and later versions using a new repository**: To change the database repository (to move from one instance of Oracle to another, for example, or to move from MySQL to SQL Server), follow the instructions in this manual to install the server and web components as a new installation. Oracle GoldenGate Veridata objects will be installed into the database that you specify during the installation, including the same tables for the user data that exist in the original repository. After the installation, you can use any migration tool to move the existing data from the old repository into the one.

- **Upgrades from version 2 and later versions using the existing repository**: Use the following procedure, Section 12.1.2, "Server and Web Upgrade Instructions for All Platforms". When installing a new version of Oracle GoldenGate Veridata Server and Oracle GoldenGate Veridata Web into an existing directory, the Oracle GoldenGate Veridata installer detects the existing installation and prompts you to upgrade or quit. The upgrade path retains the current Oracle GoldenGate Veridata database repository, without an option to drop any objects. Users can log on after the upgrade and continue to work with their existing environment.
12.1.2 Server and Web Upgrade Instructions for All Platforms

1. Start the Oracle GoldenGate Veridata repository database.
2. Stop all running comparisons or wait until they are finished.
3. Close all Oracle GoldenGate Veridata Web client sessions.
4. Shut down Oracle GoldenGate Veridata Server and Oracle GoldenGate Veridata Web User Interface. (For more information, see Section 9.2, “Starting and Stopping the Java-Based Components”)
5. Run the Oracle GoldenGate Veridata installer:
   - UNIX and Linux: Run the `GoldenGate_Veridata_platformrelease.sh` file from X-windows or an X-windows emulation program.
   - Windows: Run the `GoldenGate_Veridata_platformrelease.exe` file.

   The following steps are the same on Windows, UNIX, and Linux. These instructions show the Windows version. Some prompts appear only in the Windows version.

6. Welcome: Click Next to start the installation.
7. Select Destination Directory: Select the Oracle GoldenGate Veridata installation that you want to upgrade.
8. Click Upgrade to upgrade the existing installation (and preserve the repository). If you do not want to upgrade this installation or preserve the repository, click Quit.
9. Veridata Repository: The installer displays the encrypted version of the last known password for the repository owner. Either accept the default or type the correct one.
10. (Windows only) Windows Services User: If LocalSystem is the user that starts the existing Oracle GoldenGate Veridata, this window will be enabled. It gives you the option to identify a new login to start the service or continue as LocalSystem. If you specify a different user, that user must be granted LogonAsService privilege or the service will not start.
11. Start after Install: Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.
12. Information: Review your selections and either click Cancel to start over or Next to start the upgrade.
13. To view the Oracle GoldenGate Veridata help system after the installation, accept the default and click Finish.

12.2 Upgrading to a New Release of the Agent

In making a decision to use a Java agent versus a C-agent be aware that:

---

**Note:** An attempt to upgrade to 11.2.1 fails if your version 3 Oracle GoldenGate Veridata was not upgraded to release 3.0.0.11 and your Tomcat version is earlier than 5.5.35. The error explains that the Catalina release (one of those earlier than 5.5.35) cannot be upgraded, so you must install to a new location.
Upgrading to a New Release of the Agent

- As of Oracle GoldenGate Veridata release 3.0, you can use a Java agent for Oracle, SQL Server, DB2 LUW or z/OS, and Teradata. As of release 11.2.1, you can use a Java agent for Sybase ASE.

- An 11.2.1 release of the C-agent is available for NonStop SQL/MP.

- The release 3.0 C-agent can be used for the Oracle database for Oracle GoldenGate Veridata release 11.2.1.

The instructions provided are for upgrading a C-agent from Veridata 2.x or later or a Java agent from 3.x or later to a newer release. They assume that:

- You are upgrading the same agent on the same machine.

- You are keeping the existing file path names of the installation.

- If upgrading on a Windows system, you are keeping the existing Manager service name, if one exists.

12.2.1 Upgrading a Java Agent to a Newer Release

The instructions for upgrading a Java agent depend on whether you are using a Windows or a UNIX/Linux platform.

---

**Note:** New versions of the Java agent use different JDBC drivers, so you will need to follow the Windows’s step 6 or UNIX step 3 to verify the JDBC driver versions before starting the new agent.

---

**Upgrading a Java Agent on a Windows Platform**

To upgrade a Java agent on a Windows platform, double-click the `GoldenGate_Veridata_platformrelease` program and then follow the steps explained in this section.

1. Welcome: Click **Next** to start the upgrade.

2. Select Destination Directory: Select the Oracle GoldenGate Veridata installation that you want to upgrade.

3. Click **Upgrade** to upgrade the existing installation (and preserve the repository). If you do not want to upgrade this installation or preserve the repository, click **Quit**. If you click **Upgrade** and there is an existing agent running, you will be asked whether the agent should be stopped. Click **OK** to stop the agent.

4. Start after install: You need to confirm that the driver is correct before the agent is started, so you should not automatically start the agent after the installation is finished.

5. After the installation, click **Finish**.

6. Make certain that the value of the `server.jdbcDriver` property in the `agent.properties` file matches the `agent.properties.sample` file suggested value for the new agent. For example, connections to Oracle databases now require `odbc6.jar` instead of `ojdbc5.jar`, so the property in the `agent.properties` file should be `server.jdbcDriver=odbc6.jar`.

7. Start the new Java agent as explained in Section 9.2, "Starting and Stopping the Java-Based Components."

**Upgrading a Java Agent on a UNIX or Linux Platform**

Follow these steps to upgrade a Java agent on a UNIX or Linux Platform:
1. Stop the existing Java agent.
   
   ```shell
   agent.sh stop
   ```

2. Extract the Oracle GoldenGate Veridata media pack upgrade zip file to the installation directory. This will put the agent in place.

3. Make certain that the value of the `server.jdbcDriver` property in the `agent.properties` file matches the `agent.properties.sample` file suggested value for the new agent. For example, connections to Oracle databases now require `odbc6.jar` instead of `ojdbc5.jar`, so the property in the `agent.properties` file should be `server.jdbcDriver=ojdbc6.jar`.

4. Start the new Java agent as explained in Section 9.2, "Starting and Stopping the Java-Based Components."

### 12.2.2 Upgrading from a C-Agent to a Java Agent

There is no direct upgrade path from the C-agent to the Java-based agent. To use a Java agent:

- Follow the instructions in Chapter 3, "Installing Oracle GoldenGate Veridata Java Agent."

- You can specify the same port that was used by the existing agent Manager. If you use a different port number, make certain to notify the users of Oracle GoldenGate Veridata Web, so that they can update their connection configurations. If you use the same port, Oracle GoldenGate Veridata must be restarted so that there is no cached information from the C-agent.

### 12.2.3 Upgrading a C-Agent to a Newer Release

**To upgrade a C-agent on Windows, UNIX and Linux**

1. Wait for all jobs that use this agent to finish running.

2. From the agent installation location, run GGSCI. The default location is `GoldenGate_Veridata/agent`.

3. In GGSCI, issue the following command to stop the Manager process.
   
   ```
   STOP MANAGER
   ```

   Shutting down the Manager process prevents new agent processes from being started by the server component, if it is still running.

4. Extract the new Oracle GoldenGate Veridata Agent files to the same directory where the current agent is installed. The default relative path that is extracted is `veridata/agent`. You might need to change the extraction path if your naming conventions are different. Overwrite the existing files in the `agent` directory with the extracted files.

5. Run GGSCI from the agent directory.

6. In GGSCI, issue the following command to start the Manager process.

   ```
   START MANAGER
   ```

7. In GGSCI, issue the following command to verify that the installation succeeded. The command should return "Manager is running."

   ```
   INFO MANAGER
   ```
8. (Windows) In the Services control panel, verify that the correct service for this agent started.

**To upgrade a C-agent on NonStop**

1. Wait for all user jobs that use this agent to finish running.
2. From the agent installation location, run GGSCI. The default location is `GoldenGate_Veridata/agent`.
3. In GGSCI, issue the following command to stop the Manager process.
   ```
   STOP MANAGER
   ```

   Shutting down the Manager process prevents new agent processes from being started by the server component, if it is still running.

4. Transfer the new Oracle GoldenGate Veridata Agent files in binary mode to the volume and subvolume on the NonStop Server where you want to upgrade the agent.
5. Alter the `VERUNPAK` macro to be an edit file by issuing the following TACL command.
   ```
   FUP ALTER VERUNPAK, CODE 101
   ```
6. Run the `VERUNPAK` macro by issuing the following TACL command.
   ```
   RUN VERUNPAK
   ```
7. At the prompt, verify the location of the agent that you want to upgrade. Type `Y` to confirm the location shown or `N` to select another location.
   ```
   Installing GoldenGate at $DATA.GoldenGate Veridata
   Is this correct? (Y/N) y
   UNPAK - File decompression program - T1255G06 - (2002-05-06)
   Archive version: 1
   File Mode RESTORE Program - T9074G07 (15JAN2002)
   Copyright Tandem Computers Incorporated 1981-2002
   Summary Information
   Files restored = 7 Files not restored = 0
   GoldenGate Veridata for NonStop Installation
   Installs the GoldenGate Veridata Product
   Enter X at any prompt to quit.
   ```
8. You are prompted for a SQL catalog for the agent to use. Type the catalog name or type `X` for no catalog.
   ```
   SQL Catalog for Compilation (X for no catalog)? $data.cpscat
   SQL compiling VERIAGT
   GoldenGate Veridata Installation Complete.
   ```
9. Run GGSCI from the agent directory.
10. In GGSCI, issue the following command to start the Manager process.
    ```
    START MANAGER
    ```
11. In GGSCI, issue the following command to verify that the installation succeeded. The command should return "Manager is running."
    ```
    INFO MANAGER
    ```
12. If tables have remote partitions, copy VSNSERV to the remote nodes.

12.2.3.1 Copying VSNSERV to remote nodes

If your tables have partitions on remote nodes, you will need to place a copy of the VSNSERV module on each of those nodes.

If all of the remote nodes are the same hardware type, you can use a copy of the VSNSERV that is in the Oracle GoldenGate Veridata agent subvolume. Otherwise, you might need to download the correct agent build for that hardware type. It will include the correct VSNSERV.

To place the VSNSERV on each node, you can do either of the following:

- Install the entire Oracle GoldenGate Veridata Agent package on each of the remote nodes, even though the agent itself will not be running on them.
- Copy the VSNSERV object to each of the remote nodes. To use this option, take the following steps.

To copy VSNSERV to remote nodes

1. Copy the appropriate VSNSERV program to each of the remote nodes.
2. Log onto each remote node as a super user.
3. Issue the following commands on each remote node:
   
   FUP GIVE vsnserv, SUPER.SUPER
   FUP SECURE vsnserv, "NNNN", PROGID

   - The first command sets the VSNSERV owner as SUPER.SUPER.
   - The second command sets security and PROGID to run as SUPER.SUPER.
4. This upgrade procedure assumes that the location of VSNSERV on each remote node did not change from the original installation. If the location did change, you must change the HOST parameter(s) in the GLOBALS file. See Section 4.4.3, "Creating a GLOBALS File".
Uninstalling Oracle GoldenGate Veridata

This chapter describes how to uninstall the Oracle GoldenGate Veridata components. This chapter includes the following sections:

Section 13.1, "Uninstalling the Server and Web Components (all platforms)"
Section 13.2, "Uninstalling the Java Agent from a Windows System"
Section 13.3, "Uninstalling the Java Agent from a UNIX or Linux System"
Section 13.4, "Uninstalling the C-agent (all platforms)"

13.1 Uninstalling the Server and Web Components (all platforms)

1. Log on to the system as a user who has privileges to uninstall files.
2. Stop all running comparisons, or wait until they are finished.
3. Stop all Oracle GoldenGate Veridata Web client sessions.
4. Stop Oracle GoldenGate Veridata Server. See Section 9.2, "Starting and Stopping the Java-Based Components."
5. If you intend to drop the Oracle GoldenGate Veridata database repository, start the repository database.
6. Run the uninstall program from the Oracle GoldenGate Veridata directory:
   - UNIX and Linux: Using an X-windows or an X-windows emulation program, run uninstall.sh from the Oracle GoldenGate Veridata installation directory.
   - Windows: From the Start menu, go to the Oracle GoldenGate Veridata program folder, and select Oracle GoldenGate Veridata Uninstaller. Alternatively, you can run the uninstall.exe executable file from the installation directory.

   The uninstaller operates in the same way on Windows, UNIX, and Linux.
7. By default, the Oracle GoldenGate Veridata repository is not removed, so that user data is preserved. To drop the Oracle GoldenGate Veridata repository, select:
   - (Oracle) Yes, drop the user/schema. Supply the login of the database user or schema that owns the repository.
   - (Other databases) Yes, drop the database and Yes, drop the user or Yes, drop the login (whichever applies to the database type). Supply authentication credentials that have privileges to drop database objects. In the case of Microsoft SQL Server, you can use integrated login if the data source configuration accepts it.
8. Click Next to confirm and start the uninstall processing.
9. Click Finish to exit the uninstaller program.
10. (Optional) Remove the Oracle GoldenGate Veridata installation directory. The \textit{uninstall} program does not remove files in this directory that were created by other programs. Logs are an example of such files.

13.2 Uninstalling the Java Agent from a Windows System

1. Log on to the system as a user who has privileges to uninstall files.
2. Stop all running comparisons, or wait until they are finished.
3. Stop the agent process (see Section 9.2, “Starting and Stopping the Java-Based Components”).
4. From the \textit{Start} menu, go to the Oracle GoldenGate Veridata Agent program folder, and then select \textbf{Oracle GoldenGate Veridata Agent Uninstaller}. Alternatively, you can run the \texttt{uninstall_veridata_agent.exe} executable file from the installation directory.
5. Click \textbf{Next} on the welcome screen to start the removal process.
6. One of two messages is displayed, depending on whether the agent is still running. A running agent could indicate that server processes still are using this agent. Click \textbf{Next} to continue with the uninstallation process or click \textbf{Quit} to cancel it. If you continue with the uninstallation, the status is displayed.
7. Click \textbf{Finish} to exit the uninstaller program.
8. (Optional) Remove the Oracle GoldenGate VeridataAgent installation directory. The \textit{uninstall} program does not remove files in this directory that were created by other programs. Logs are an example of such files.

13.3 Uninstalling the Java Agent from a UNIX or Linux System

1. Log on to the system as a user who has privileges to uninstall files.
2. Stop all running comparisons, or wait until they are finished.
3. Stop the agent process (see Section 9.2, “Starting and Stopping the Java-Based Components”).
4. Remove the files in the agent installation directory.

13.4 Uninstalling the C-agent (all platforms)

1. Log on to the system as a user who has privileges to uninstall files.
2. Stop all running comparisons, or wait until they are finished.
3. From the agent installation location, run GGSCI. The default location is \texttt{GoldenGate_Veridata/agent}.
4. In GGSCI, issue the following command to stop the Manager process.
   \begin{verbatim}
   STOP MANAGER
   \end{verbatim}
   Shutting down the Manager process prevents new agent processes from being started by the server component, if it is still running.
5. (Optional) Remove the files that are in the installation location.
6. (Optional, NonStop) Purge any copies of the file named $*.*.VSNSEerv.$*.*.VSNSEerv.
This appendix describes how to download the Oracle GoldenGate Veridata files. Follow these steps to download:

- Navigate to [http://edelivery.oracle.com](http://edelivery.oracle.com).
- On the Welcome page:
  - Click **Sign In/Register**.
- On the Export Validation page:
  - Enter your identification information.
  - Click **Yes** to accept the Trial License Agreement (even if you have a permanent license).
  - Click **Yes** to accept the Export Restrictions.
  - Click **Continue**.
- On the Media Pack Search page:
  - Select the Oracle Fusion Middleware Product Pack.
  - Select the platform on which you will be installing the software.
  - Click **Go**.
- In the Results List:
  - Select the Media Pack that you want.
  - Click **Continue**.
- On the Download page:
  - Click **Download** for each component that you want. Follow the automatic download process to transfer the mediapack.zip file to your system.

**Note:** Before installing the software, review the release notes for any new features, new requirements, or bug fixes that affect your current configuration.
Changing Oracle GoldenGate Veridata Web Port Numbers

This appendix describes how to change Oracle GoldenGate Veridata Web port numbers.

Ports are specified in the configuration file of the Tomcat web server application that was installed with Oracle GoldenGate Veridata. The name of this file is as follows (example shows UNIX file structure):

Veridata_install_directory/Oracle GoldenGate Veridata/server/web/conf/server.xml

- The first line in the file is

```xml
<Server port="8820" shutdown="SHUTDOWN" debug="0">
```

This is the port that the web server listens on for a shutdown message. You can change this port as needed.

- About halfway into the script, there are the following lines that specify ports:

```xml
<Service name="Tomcat-Standalone">
  <Connector className="org.apache.coyote.tomcat4.CoyoteConnector" port="8830" minProcessors='5' maxProcessors='75'
    enableLookups='true' redirectPort='8443'
    acceptCount='100' debug='0' connectionTimeout='20000'
    useURIValidationHack='false' disableUploadTimeout='true' />
```

  - The first port number is the http port that users type on the address line when starting Oracle GoldenGate Veridata Web from a browser. You can change this port.
  - The second port is used internally. Do not change this port unless it conflicts with another program.
Configuring an ODBC connection for a SQL Server Repository

This appendix describes how to define a connection to an SQL Server repository. To use a SQL Server database as a repository, you must establish a system data source name (DSN). A DSN stores information about how to connect to the database.

---

**Note:** You can use these same steps to edit or verify an existing DSN.

---

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

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

3. Do one of the following:
   - To create a new DSN, click **Add**.
   - To verify an existing DSN, select the name from the **System Data Sources** list and then click **Configure**.

4. (New data source only) Under **Create New Data Source**, select either the **SQL Server** or **SQL Native Client** driver.

5. Click **Finish**. A new wizard is displayed.

6. Supply or verify the following:
   - **Name**: Can be of your choosing.
   - **Description**: (Optional) Type a description of this data source.
   - **Which SQL Server do you want to connect to**: Select (or type) the name of the SQL Server instance. To create the DSN for a named instance, specify the host name and instance name in this format: `host\instance`. To create a DSN for
the default instance, just the host name is sufficient. If the instance is on the local machine, you can select the localhost option.

7. Click Next.

8. Click the Client Configuration button.

9. In the Edit Network Library Configuration popup:
   - Enter a server alias name (optional).
   - Deselect the Dynamically determine port box, and then enter the name of a port number to use (the default of 1433 is acceptable). Oracle GoldenGate Veridata requires the use of a static port.
   - Under Network Libraries, make certain that TCP/IP is selected (default).

10. Click OK.

11. Based on your SQL Server configuration, select either With Windows NT authentication to require a network login or select With SQL Server authentication using a login to require database credentials. If you select SQL Server authentication, supply a login and password to test the DSN. This is for test purposes, only. This is not necessarily the user that will be used by Oracle GoldenGate Veridata.

12. Click Next.

13. You can leave all of the Microsoft SQL Server DSN Configuration settings to their default values.

14. Click Next.

15. You can leave all of the Create New Data Source to SQL Server settings to their default values.

16. Click Finish.

17. Click Test Data Source to test the connection.

18. Close the confirmation box and the Create a New Data Source box.
Glossary

agent
See Oracle GoldenGate Veridata Agent.

client
An interface to Oracle GoldenGate Veridata Server, from which you configure or execute comparison work. Oracle GoldenGate Veridata provides a GUI client named Oracle GoldenGate Veridata Web and a command client named vericom, which provides a subset of the functions that can be performed from the web interface.

column mapping
An Oracle GoldenGate Veridata Web configuration object that establishes the correct structural relationship between the source and target objects in a compare pair.

compare group
See group.

compare pair
A configuration object in Oracle GoldenGate Veridata Web that associates one source table or file with one target table or file for the purpose of comparing their data. The source table typically contains data that is known or presumed to be up-to-date and accurate. The target table is a different table which contains identical data, such as a backup, that is synchronized with that of the source by some mechanism, such as replication.

comparison format
A standardized format to which Oracle GoldenGate Veridata transforms the native data type of a column. The use of standardized formats enables Oracle GoldenGate Veridata to support heterogeneous comparisons (those between different types of databases) where source and target data types are different, but similar.

comparison report
A report that is created at the conclusion of a comparison that contains information about the objects that were compared, the number of rows that are out-of-sync, and statistics about performance.

confirmation step or confirm-out-of-sync (COOS) step
A follow-up comparison that is performed after the initial comparison step to confirm a row’s status as in-sync or perpetually out of synchronization. The confirmation step waits until after a specified replication latency threshold has expired, to give a replication mechanism time to post any change that was in-flight.
connection
A configuration object in Oracle GoldenGate Veridata Web that stores information about how the Oracle GoldenGate Veridata Server process will connect to a Veridata Agent process. It consists of a DNS host name or IP address, a Manager port number, and a data source type.

data source
A set of stored data, such as a relational or hierarchical database.

delta processing
A performance feature that Oracle GoldenGate Veridata supports on the NonStop platform by which Oracle GoldenGate Veridata detects which data blocks in the database were modified since a previous comparison and only compares the rows in those blocks. Rows in unchanged blocks are skipped.

digital signature
A message signed with a private key that can be verified by a person or process that has the public key, thereby proving that the message is unchanged and authentic.

GGSCI
The command line interface to the Oracle GoldenGate replication software, which also is used as the command interface to the Manager process of the Oracle GoldenGate Veridata Agent.

GLOBALS file
A parameter file that is required on some platforms where Oracle GoldenGate Veridata Agent is installed. It stores parameters that may be required in some installation conditions.

group
A configuration object in Oracle GoldenGate Veridata Web that is a logical container for one or more compare pairs (sets of related source and target tables) and their associated data source connections.

hash
A method of converting data into a unique numeric value (hash value) by substituting or transposing the data. If two hash values are different, then the degree of certainty that the two original input values are identical is extremely high (but not absolute). To be absolutely certain that values are identical, you can use a literal comparison.

in-flight
One of the possible states of a row during a comparison. The target row was out-of-sync in the initial comparison step, but was since updated and still do not match those of the source.

in-sync
One of the possible states of a row during comparison. The target row values match the source values that were retrieved during the initial comparison step.

initial comparison step, or row-hash step
The first step in a comparison, in which rows are retrieved from the source and target tables with a query and then compared. Target rows that appear to be out-of-sync are
stored in a maybe out-of-sync (MOOS) queue in memory. The rows will be assessed further by the confirmation step.

**install**
A program that is used on Windows platforms to install Oracle GoldenGate Veridata programs as a Windows service.

**job**
A unit of work that is linked to one or more compare groups that are processed together.

**key, or key columns**
A unique identifier that Oracle GoldenGate Veridata uses to order rows for comparison. This could be a primary key, a unique index, or a user-defined identifier.

**latency**
The difference in time between when a change is made to source data and when that change is reflected in the target data.

**literal comparison**
A comparison that is performed value for value, as opposed to a comparison that uses a hash.

**Manager**
Coordinator program of the Oracle GoldenGate Veridata Agent component. It coordinates requests sent to the agent process by the Oracle GoldenGate Veridata Server process.

**Maybe-out-of-sync queue, or MOOS queue**
A section of memory that stores rows that were determined to be out-of-sync in the initial comparison step, until they can be confirmed as in-sync or out-of-sync in the confirmation step.

**Oracle GoldenGate Veridata Agent**
Runs on the system where a source or target database resides. It fetches and returns blocks of rows, determines whether rows are out-of-sync, returns column-level detail, and executes requests on the database. The agent is started at the request of the Oracle GoldenGate Veridata Server and is controlled by the Oracle GoldenGate Veridata Manager process.

**Oracle GoldenGate Veridata Server**
The comparison engine that coordinates Oracle GoldenGate Veridata tasks, compares data, confirms out-of-sync data, and produces reports. Row sorting also can be performed by the server if the server-side sorting option is selected.

**out-of-sync**
The data values in a source row are not identical to those of the corresponding target row.

**out-of-sync report**
A report that is created at the conclusion of a comparison that provides summary and column-level details about out-of-sync rows and the operations that are needed to bring them back into synchronization.
partition
A selection criteria that is defined within Oracle GoldenGate Veridata Web or the vericom interface, which enables comparisons to be performed against a subset of the rows in a table or file.

persistently out-of-sync
One of the possible states of a row during comparison. During the confirmation step, the target values have not changed since the initial comparison step and still are not the same as those of the source.

profile
A configuration object in Oracle GoldenGate Veridata Web that is a stored set of global processing parameters, each containing unique settings for a specific purpose.

replication
The process of duplicating sets of data to one or more other locations in near real-time for such purposes as recovery or decision support.

repository
A collection of database objects that persists configuration information created by Oracle GoldenGate Veridata Web users to disk, saving it permanently as a user environment.

row partition
See partition.

row-hash step
See initial comparison step.

server
See Oracle GoldenGate Veridata Server.

server-side sorting
The sorting of source and target rows for comparison when performed by Oracle GoldenGate Veridata Server.

source
The database or location that is trusted to contain the most accurate and current version of data in an environment that contains replica versions of that data.

target
A database or location that contains a secondary or replica set of data that corresponds to a primary, or source, set of data.

validation
Performed by Oracle GoldenGate Veridata to test source and target table structures for compatibility.

vericom
The command line interface to Oracle GoldenGate Veridata.
A
ADDEVENTS Windows service option, 4-4
ADDSERVICE Windows service option, 4-4
administration tool
using, 8-2
administrator
specifying user as, 5-3, 6-3, 7-2
Administrator role, 8-2
agent component
about, 1-2
installing, 3-1, 4-1
starting, 9-1
stopping, 9-1
system requirements, 2-1
uninstalling, 13-2
upgrading, 12-2
AUTOSTART Windows service option, 4-4

B
browser
connecting to server from, 9-3
supported kinds, 2-10

C
category.dll, 4-4
client
connecting to, 9-3
system requirements, 2-9
clustered keys, 2-10
command line
about, 1-3
running comparisons from, 10-1
compare pairs
about, 1-3
comparison
configuring in web interface, 1-3
results of
viewing, 1-5, 1-6
running from command line, 10-1
comparison report, 1-5, 1-6
confirmation step
about, 1-5
delaying, 10-3
skipping, 10-4
connections
as configuration object, 1-3
COOS step
about, 1-5
delaying, 10-3
skipping, 10-4

D
data
how compared, 1-4
See also rows
data source name (DSN)
for SQL Server repository, C-1
database
as repository, 1-2
specifying for repository, 5-3, 6-3, 7-2
verifying, 5-1, 6-1
delta processing
about, 1-4
enabling, 10-6
DetailReportViewer role, 8-2
directory
installation
agent, 3-3, 4-2, 4-5
server, 3-3, 5-2, 6-2, 7-1
user data location, 5-2, 6-2, 7-1
disk space
agent component, 2-3
server component, 2-5
dYNAMICPORTLIST parameter, 4-2, 4-5, 4-8

E
Enscribe
agent
installing, 4-5
agent permissions, 2-4
agent supported, 2-3

F
Firefox, 2-10
G
GGSMGR default Manager name, 4-3
ggsmgr.dll, 4-4
GGSPREFIX parameter, 4-7
GLOBALS file, 4-3, 4-7
groups
   about, 1-3
Guardian
   installing agent on, 2-3

H
hash
   rows, 1-4
HOST parameter, 4-7
HTTP port, 5-3, 6-2, 7-2

I
in-flight
   definition, 1-5
initial comparison step, 1-4
installing
   agent component, 3-1, 4-1, 4-2, 4-5
   server component, 5-1, 6-1, 7-1
   web component, 5-1, 6-1, 7-1
in-sync
   definition, 1-5
Internet Explorer, 2-10
IPC buffer memory, 11-1

J
Java Runtime Environment, 2-10
JDBC driver
   Teradata, 3-3
jobs
   about, 1-3
   running, 10-1

K
keys
   clustered, 2-10

L
latency
   replication, 1-5
Linux
   installing agent, 3-1, 4-1
   installing server and web components, 5-1, 6-1, 7-1

M
Manager
   about, 1-2
   as Windows service, 4-3
   configuring on NonStop, 4-7
   multiple on same system, 4-3, 4-7
   name
      customizing, 4-3, 4-7
   port number, 4-2
   starting, 9-1
   stopping, 9-1
   upgrading, 12-2
MANUALSTART Windows service option, 4-4
maybe out-of-sync queue (MOOSQ), 1-4
memory
   managing for server component, 11-1
   server requirements, 2-5
MGRSERVNAME parameter, 4-3
MOOS queue
   about, 1-4
   memory use, 11-1
   multi-tier server installation, 2-7
MySQL
   using as repository, 5-1

N
NonStop SQL/MP
   installing agent, 4-5
   using delta processing on, 1-4

O
ODBC connection
   for SQL Server repository, C-1
   ODBCAD32.EXE client, C-1
OOSXML file
   about, 1-5
   based on OOS from previous run, 10-3
   controlling size of, 10-4
Oracle
   agent
      installing, 3-2, 4-1, 4-2
      ORACLE_HOME variable, 6-3
   using as repository, 6-1
Oracle GoldenGate Veridata
   installing, 3-1, 5-1, 6-1
   system requirements, 2-1
   uninstalling, 13-1
   upgrading, 12-1
Oracle GoldenGate Veridata Agent
   about, 1-2
   installing, 3-1, 4-1, 4-2, 4-5
   starting, 9-1
   stopping, 9-1
   uninstalling, 13-2
   upgrading, 12-2
Oracle GoldenGate Veridata repository
   about, 1-2
   system requirements, 2-7
Oracle GoldenGate Veridata Server
   about, 1-2
   configuration parameters, 11-1
   installing, 5-1, 6-1, 7-1
   starting, 9-1
uninstalling, 13-1
upgrading, 12-1
Oracle GoldenGate Veridata Web
  about, 1-2
  connecting to, 9-3
  installing, 7-1
  starting, 9-1
  uninstalling, 13-1
  upgrading, 12-1
out-of-sync (OOS) report
  about, 1-5
  format of, 10-4

P
  parameters
    server, 11-1
  partition
    specifying for rows, 10-5
  partitions on remote NonStop nodes, 4-6
  PASSWORD Windows service option, 4-4
  persistently out of sync
    definition, 1-5
  port
    default, 2-10
    HTTP, 5-3, 6-2, 7-2
    Manager, 4-2
    MySQL database, 5-3
    numbers
      changing, B-1
      Oracle database, 6-4
      shutdown, 5-3, 6-2, 7-2
      SQL Server database, 7-2
      static
        SQL Server, 2-9, C-2
        web component, 9-3
  PowerUser role, 8-2
  profile
    about, 1-3
    overriding, 10-3

R
  RAM requirements
    agent component, 2-3
    server component, 2-5
  removing Oracle GoldenGate Veridata, 13-1
  replication latency, 1-5
  report
    comparison, 1-5, 1-6
  ReportViewer role, 8-2
  repository
    about, 1-2
    owner
      specifying, 5-4, 6-4, 7-2
      system requirements, 2-7
      where to install, 2-7
    roles
      security, 8-1
      row hash queue memory, 11-1
row hash step
  see initial comparison step
rows
  comparing, 10-1
  comparing only previously out-of-sync, 10-3
  how compared, 1-4
  number in OOSXML chunk, 10-4
  specifying partition, 10-5

S
  security
    configuring, 8-1
  server component
    about, 1-2
    configuration parameters, 11-1
    installing, 5-1, 6-1, 7-1
    starting, 9-1
    system requirements, 2-5
    uninstalling, 13-1
    upgrading, 12-1
    server.max_comparison_sort_memory, 11-3
    server.max_concurrent_comparison_threads, 11-3
    server.max_sort_memory, 11-3
    server.sort_waiting_threshold, 11-4
    service
      installing as on Windows, 3-3, 4-2, 5-4, 6-4, 7-3
      shutdown port, 5-3, 6-2, 7-2
    sort
      about, 2-5
      memory configuration, 11-1
      source key range
        specifying as partition, 10-5
      SQL predicate
        specifying, 10-5
      SQL Server
        agent
          installing, 3-2
          using as repository, 7-1
        SQL/MP clustered keys, 2-10
    starting
      Manager, 4-5, 4-8
      Oracle GoldenGate Veridata Agent, 9-1
      Oracle GoldenGate Veridata Server, 9-1
      Oracle GoldenGate Veridata Web, 9-1
    statistics for comparisons, 1-6
    stopping
      Manager, 9-1
      Oracle GoldenGate Veridata Agent, 9-1
      Oracle GoldenGate Veridata Server, 9-1
      Oracle GoldenGate Veridata Web, 9-1
    subset
      of rows, 10-5
    system requirements, 2-1

T
  tables
    comparing, 10-1
  tablespace
specifying, 6-4
Teradata agent
  installing,  3-2
threads
  number to use, 10-3
Tomcat Web Server
  Administration tool, 8-2
  configuration file,  B-1
tracing
  setting, 10-4

U
uninstalling Oracle GoldenGate Veridata,  13-1
UNIX
  installing agent,  3-1, 4-1
  installing server and web components, 5-1, 6-1,
  7-1
upgrading Oracle GoldenGate Veridata, 12-1
USER Windows service option, 4-4
users and groups, 8-1

V
vericom tool, 10-1
VSNSERV module, 4-6

W
web component
  about,  1-2
  connecting to, 9-3
  installing,  5-1, 6-1, 7-1
  system requirements, 2-9
  uninstalling, 13-1
  upgrading, 12-1
Windows
  installing agent, 4-2
  installing server and web components, 5-1, 6-1,
  7-1