Oracle® GoldenGate Veridata

Administration Guide

Version 3.0

October 2009



Administration Guide, Version 3.0

Copyright © 1995, 2009 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface **New in GoldenGate Veridata Version 3.0** 7
- Expanded database support 7
- Oracle enhancements 7
- NSK enhancements 7
- Other enhancements 8

Chapter 1 **About GoldenGate Veridata** 9
- GoldenGate comparison objects 11
- How GoldenGate Veridata compares data 11
  - Initial comparison step 11
  - Confirmation step 12
- Viewing comparison results 13
  - Out-of-sync report 13
  - Comparison report 13

Chapter 2 **GoldenGate Veridata System Requirements** 15
- General information 15
- Supported platforms 15
- GoldenGate Veridata Agent system requirements 15
  - Disk and memory requirements for the agent component 17
  - Database privileges for the agent component 17
- GoldenGate Veridata Server system requirements 18
  - Location for the server component 18
  - Additional programs for the server component 18
  - Disk and memory requirements for the server component 19
  - Repository requirements for the server component 20
- GoldenGate Veridata Web system requirements 22
  - Location for the web component 22
  - Java environment for the web component 22
  - Supported browsers for the web component 22
  - Recommended screen resolution for the web component 23

Port numbers for the web component 23
Security for the web component 23
SQL/MP data exclusions 23
Chapter 3 Installing GoldenGate Veridata Java Agent 24
Installing the Java agent on a UNIX or Linux system 24
Installing the Java agent on a Windows system 26
Chapter 4 Installing GoldenGate Veridata C-Agent 32
Installing the C-agent on a UNIX or Linux system 32
Installing the C-agent on a Windows system 33
Installing the GoldenGate Veridata Agent files 33
Specifying a custom Manager name 34
Installing Manager as a Windows service 34
Configuring Manager 36
Installing the C-agent on a NonStop system 36
Installing the GoldenGate Veridata Agent files 37
Copying VSNSERV to remote nodes 37
Creating a GLOBALS file 38
Configuring Manager 39
Chapter 5 Installing the server and web components using a MySQL repository 41
Preinstallation system check 41
Installing the GoldenGate Veridata software 41
Chapter 6 Installing the server and web components using an Oracle repository 50
Preinstallation system check 50
Installing the GoldenGate Veridata software 51
Chapter 7 Installing the server and web components using a SQL Server repository 60
Preinstallation system check 60
Installing the GoldenGate Veridata software 60
Chapter 8 Configuring GoldenGate Veridata security 70
Securing the GoldenGate Veridata files 70
Controlling access to the installation directories 70
Securing files that contain business data 70
Securing access to GoldenGate Veridata Web 71
Chapter 9 Running the GoldenGate Veridata Programs 73
Starting and stopping the C-agent and Manager 73
Starting and stopping the Java-based components 74

Controlling the Java-based components from the command line 74
Controlling the Java-based components as Windows services 75
Reloading log information 75
Connecting to GoldenGate Veridata Web 75

Chapter 10 Running comparisons from the command line 77

Overview of the command line interface 77
Running vericom 77
Vericom exit statuses 83
Vericom output examples 84

Chapter 11 GoldenGate Veridata Server configuration parameters 86

Overview of the server memory 86
Estimating memory usage 87
How to set a parameter 88
Parameter descriptions 88
server.max_nsort_memory 88
server.max_concurrent_comparison_threads 88
server.max_comparison_nsort_memory 89
server.nsort_waiting_threshold 89
server.nsort_compare_function_pool_size 89
Additional parameters in the web interface 90
Sorting Retry Limit 90
Sorting Retry Interval 90

Chapter 12 Upgrading GoldenGate Veridata 91

Upgrading to a new version of the server and web components 91
Upgrade support overview 91
Server and web upgrade instructions for all platforms 91
Upgrading to a new version of the agent 96
Upgrading from a C-agent to a Java agent 96
Upgrading a C-agent to a newer version 97

Chapter 13 Uninstalling GoldenGate Veridata 100

Uninstalling the server and web components (all platforms) 100
Uninstalling the Java agent from a Windows system 102
Uninstalling the Java agent from a UNIX or Linux system 103
Uninstalling the C-agent (all platforms) 103

Appendix 1 Changing GoldenGate Veridata Web port numbers 105

Appendix 2 Configuring an ODBC connection for a SQL Server repository 106

Glossary of GoldenGate Veridata Terms 112

Index 116

PREFACE

New in GoldenGate Veridata Version 3.0

Expanded database support

- GoldenGate Veridata Server now supports Microsoft SQL Server as a repository database.
- GoldenGate Veridata Agent now supports the following databases
  - Microsoft SQL Server
  - Teradata
- Heterogeneous comparisons now can be performed between the different kinds of databases that are supported by GoldenGate Veridata Agent. The concept of the *comparison format* has been introduced to support heterogeneous data type mapping where source and target data types are different, but similar. During system-generated (automatic) mapping, the comparison format is used, rather than the data type name, to determine whether or not the source and target columns are compatible for comparison. The server process selects the correct format from a set of them that is supported for a given data type and, when applicable, determines the correct precision and scale. A user can override the system selection to specify another supported comparison format. For more information about this feature, see the online help.
- The agent component is now provided in Java code. There is one agent executable for all supported databases. The exception is the NonStop platform: the C-agent is still required for Enscribe and NonStop SQL/MP, and also can be used for Oracle like-to-like comparisons.

Oracle enhancements

- The installation now includes the Oracle client software for use with an Oracle repository.
- Oracle EZCONNECT is now supported for repository connections. The installer prompts for either EZCONNECT or TNSNAMES.

NSK enhancements

- Delta processing now can be enabled or disabled from the Existing Compare Pairs tab of the Compare Pair Configuration page, instead of having to go to the Manual Mapping tab and then select from the drop-down list.

Other enhancements

- The Java agent does not require a GoldenGate Manager process. There is an installer wizard for Windows systems that collects all of the required installation information, including the Windows service. For UNIX systems, the agent is installed in a simple extraction process, with a properties file that requires minor editing to specify some parameter values. (This properties file also underlies the Windows installation.)
- GoldenGate Veridata Web has a new skin that makes it easier for the eye to move around the sections of a page. Included in the improvements are enhanced status icons and a refined navigation bar that does not require the nodes to be expanded.
- Firefox 3.0 is now supported.
- GoldenGate Veridata now supports case-sensitive name mapping while doing pattern mapping of compare pairs. First a case-sensitive match is tried, then a case-insensitive match is tried for any objects that still remain unpaired.
- The out-of-sync report and the Out-of-Sync Rows view in the web interface now display the values of all of the out-of-sync columns that are in a table or file.
- To support the addition of a comparison format mapping function (for specifying a standard datatype to which native datatypes are mapped), the Edit Connection page now has three tabs: Connection Details, where the connection information is stored; Connection Settings, where connection parameters reside; and Connection Properties, where the comparison formats are specified.
- From the Details for Finished Compare Pair page, you can now go directly to the Compare Pair Configuration page for that pair.
- Some columns and sections on some pages have been relocated for better workflow, without changing functionality.

CHAPTER 1

About GoldenGate Veridata

GoldenGate Veridata compares one set of data to another and identifies data that is out of synchronization. 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, GoldenGate Veridata can run concurrently with data transactions and replication, while still producing an accurate comparison report.

GoldenGate Veridata can compare data across the Enscribe, Oracle, NonStop SQL/MP, SQL Server, and Teradata platforms. It will map column data types 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

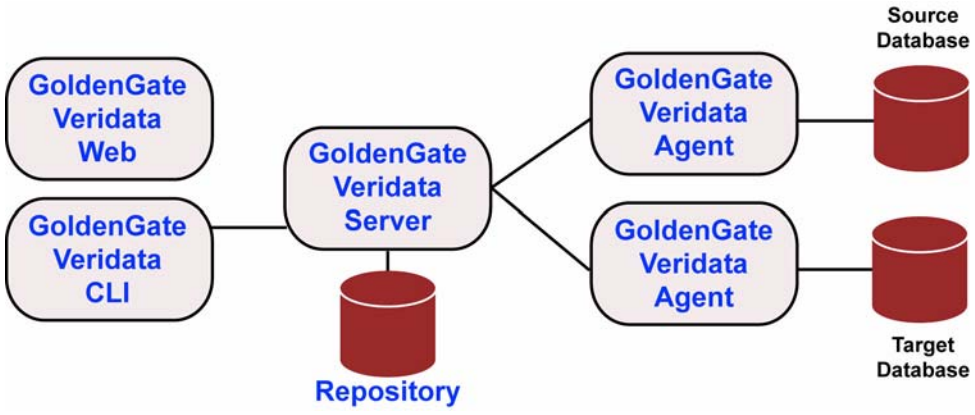


Figure 1 GoldenGate Veridata components

GoldenGate Veridata Server

The GoldenGate Veridata Server performs the following functions:

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

GoldenGate Veridata Web

GoldenGate Veridata Web is a client to GoldenGate Veridata Server. It 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

GoldenGate Veridata Repository

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

GoldenGate Veridata Agent

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

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

GoldenGate provides a Java-based agent for the Oracle, SQL Server, and Teradata databases, and also provides a C-based agent for the Oracle, Enscribe, and NonStop SQL/MP databases.

GoldenGate Manager

The GoldenGate Manager process is part of the C-agent. It controls the GoldenGate Veridata Agent process. A Manager is not used in a Java agent.

GoldenGate Veridata CLI (Command Line Interface)

GoldenGate Veridata includes a command line client to GoldenGate Veridata Server. The program name is vericom. This interface enables you to run comparisons, specify overrides, and stop GoldenGate Veridata Server by using automated programs. Configuration and monitoring functions must be performed through GoldenGate Veridata Web.

GoldenGate comparison objects

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

- Configure datasource connections: GoldenGate Veridata Server connects to 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 a GoldenGate Veridata Agent on that host, and the datasource 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 datasource 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 runtime 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 GoldenGate Veridata includes a default profile that contains settings that apply to most usage scenarios. However, as you gain experience with 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 GoldenGate Veridata online help.

How 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 GoldenGate Veridata Web.

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, 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 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 your organization has a license to use server-side sorting. Delta processing is a performance feature by which 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 controlled by the "Delta processing enabled" controls on the Compare Pair Configuration page and the Run/Execute Job page of GoldenGate Veridata Web. These controls allow 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 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.

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 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.

Viewing comparison results

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

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 GoldenGate Veridata programs. As such, it can be viewed from the 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.

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 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.

CHAPTER 2

GoldenGate Veridata System Requirements

General information

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

Supported platforms

GoldenGate Veridata supports the following databases for comparisons:

- Enscribe
- Oracle
- NonStop SQL/MP
- SQL Server
- Teradata

To view a list of currently supported database versions and operating systems:

1. Go to http://www.goldengate.com.
2. Go to the **Support** link.
3. Log in with your GoldenGate customer email and password.
4. From the Software Downloads page, navigate to the GoldenGate Veridata page.

GoldenGate Veridata Agent system requirements

One 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.

GoldenGate Veridata Java agent

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

- Oracle
- SQL Server
- Teradata

The Java agent connects to the database by using JDBC (Java Database Connectivity). The Java agent enables 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.

The Java agent is installed with the following methods:

- Installation program on Windows
- Compressed file set on UNIX and Linux

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 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.

Table 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 1.5. Java 6 is preferred.
Oracle
◆ Make certain the listener is configured and running.
SQL Server
A static TCP/IP port must be configured and enabled.
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.

GoldenGate Veridata C-agent

GoldenGate 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 optional for the Oracle database. The C-agent may provide better performance when running like-to-like comparisons.
- 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 2 Environment variables for dynamically linked C-agents

Variable	Operating system
PATH=<Oracle libraries>	Microsoft Windows
SHLIB_PATH=<Oracle libraries>	HP-UX
LIBPATH=<Oracle libraries>	IBM AIX
LD_LIBRARY_PATH=<Oracle libraries>	All other supported UNIX platforms

Disk and memory requirements for the agent component

- The agent requires at least 1GB of RAM.
- The disk space requirements for the 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 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 “Disk and memory requirements for the server component” on page 19.

Database privileges for the agent component

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 GoldenGate Veridata Web interface. The following are the database privileges that are required for the database user.

Table 3 Required database privileges for GoldenGate Veridata Agent

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 GoldenGate Veridata to use. ◆ Read access to the DDL dictionaries that you want GoldenGate Veridata to use. ◆ Read access to the Enscribe and SQL/MP tables that will be compared. ◆ Read, write, create, purge permissions for the GoldenGate Veridata report and trace files, and access to the subvolumes where they are installed.
SQL Server
◆ db_datareader or the equivalent on the tables to be compared. ◆ VIEW DEFINITION in the databases to be compared. ◆ The database must allow SQL Server authentication.
Teradata
◆ SELECT privileges on the tables to be compared.

GoldenGate Veridata Server system requirements

Location for the server component

The server and web 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 or web components on a NonStop system. To use GoldenGate Veridata for NonStop databases:

- Install the server and web 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.

Additional programs for the server component

GoldenGate Veridata Server requires a local installation of the Microsoft Visual C++ 2005 Redistributable package. This package is installed by the installer and

should be reinstalled with each new GoldenGate Veridata version. To use an existing package, it must be at least as current as the version that is distributed with GoldenGate Veridata.

Disk and memory requirements for the server component

The server component uses about 30 MB of fixed virtual memory for basic tasks. The remaining virtual memory is used for comparisons. The main consumers of processing resources on the GoldenGate Veridata Server machine are the row sorting operations of the NSort component of the initial comparison step when using server-side sorting. Nsort is a third-party sort/merge library that is built into GoldenGate Veridata.

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:

((number of columns in key + 1) * 4) + 14 + (comparison width of each key column)

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

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

((1 + 1) * 4) + 14 + 8 = 30 bytes of memory for this row

NOTE This assumes that all non-key columns are compared by using a hash, not literally. More space is needed for literal comparisons.

Beyond this allocation, memory is required for storing rows during the second step of processing, the confirmation step. This can be significant 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.

NSort 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 Nsort 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 GoldenGate Veridata application:

- 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 allocate 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 page 86.

Repository requirements for the server component

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 GoldenGate Veridata Web. You can use the following databases as a repository:

- MySQL
- Oracle
- SQL Server

When installing 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 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. 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.

Database privileges

During installation, you will be prompted for a user (or login) and password for 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. It is best practice to dedicate this user to GoldenGate Veridata.

- To have the 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 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.

Table 4 Required database privileges for 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 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 INSERT, UPDATE, DELETE from tables in the database SELECT from tables in the database ALTER SCHEMA of the login's default schema

Database environment

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

Table 5 Required environment components for 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 tablespaces 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 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 page 106.) 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.

GoldenGate Veridata Web system requirements

Location for the web component

See "Location for the server component" on page 18. One installer program installs both GoldenGate Veridata Server and GoldenGate Veridata Web.

Java environment for the web component

The installation package includes the required Java Runtime Environment (JRE) 1.5.x, which is installed into the JRE directory within the main GoldenGate Veridata Server installation. To ensure that the web client runs properly, do not change this JRE environment.

Supported browsers for the web component

- Microsoft Internet Explorer version 6 or higher

- Mozilla Firefox version 1.0.4 and higher

Recommended screen resolution for the web component

Minimum of 1024 x 768

Port numbers for the web component

- The GoldenGate Veridata Web application interacts with the Apache Tomcat web server. Tomcat is installed with GoldenGate Veridata Server on default ports. You can change the port numbers if needed (see Appendix 1 on page 105).
- GoldenGate Veridata Web makes use of the following default ports, which you can change during installation:
  - 8820: Shutdown port
  - 8830: HTTP port
  - 4150: Server communication port (requires a password for server-client communication)
  - Other port numbers are used by default. You can change these port numbers after installation by following the directions in "Changing GoldenGate Veridata Web port numbers" on page 105.

Security for the web component

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

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 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 GoldenGate Veridata.

CHAPTER 3

Installing GoldenGate Veridata Java Agent

These instructions are for installing a new, clean copy of the Java-based GoldenGate Veridata Agent software. To upgrade an existing installation, see Chapter 12 on page 91.

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

- Oracle
- SQL Server
- 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).

Installing the Java agent on a UNIX or Linux system

NOTE Make certain you have reviewed the System Requirements on page 15 before installing GoldenGate Veridata Agent.

1. Create a directory for the agent.
2. Download the GoldenGate Veridata Agent from the GoldenGate support site at http://www.goldengate.com/services/support.html. A valid user name and password is required. Select either the .tar or .zip package. Both contain identical software.
3. Unzip the files with gunzip.

   ```
   gunzip <filename>.tar
   ```

4. Extract the files into the directory.

   ```
   tar -xf <filename>.tar
   ```

5. In the agent directory within the installation directory, make a copy of the agent.properties.sample file, and rename the copy to agent.properties.
6. Open the agent.properties file with a text editor. This file contains comments that help guide your work.

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

Table 6 agent.properties file parameters

Parameter	Value
`server.port`	A local port where the agent listens for communication requests from GoldenGate Veridata Server.
`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=ojdbc5.jar`
`database.transaction.isolation`	Controls the transaction isolation level that is used during the initial comparison step.

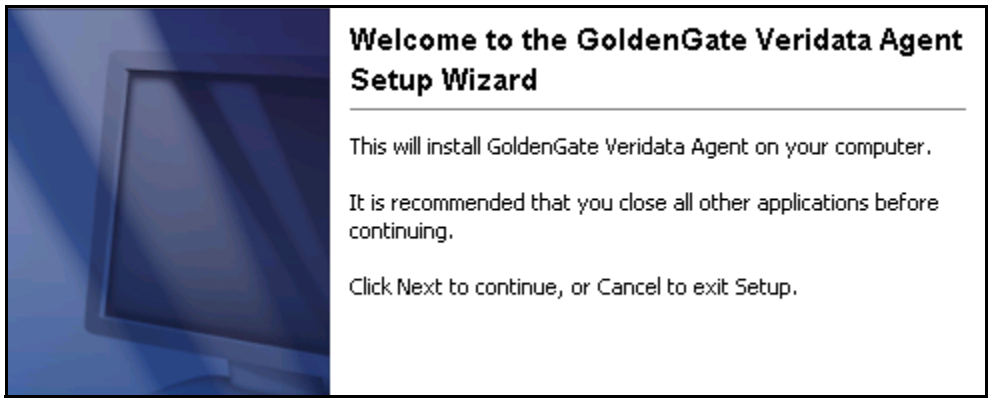
8. Save and close the file.

Installing the Java agent on a Windows system

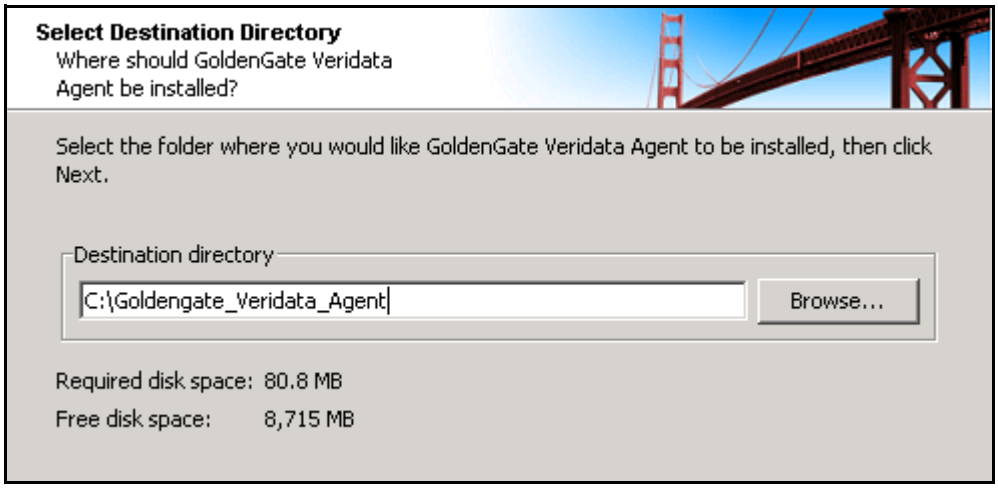
NOTE Make certain you have reviewed the System Requirements on page 15 before installing GoldenGate Veridata Agent.

1. Start the database for which you are installing this agent.
2. Download the GoldenGate_Veridata_Agent_<platform><version>.exe file from the GoldenGate support site at http://www.goldengate.com/services/support.html. A valid user name and password is required.
3. Run the executable in either of these ways:
   - **As a GUI wizard**: Double click the program and then following 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.

     ```
     Start /wait GoldenGate_Veridata_Agent_<platform><version>.exe
     -c
     ```
4. Click **Next** on the first page to start the installation.

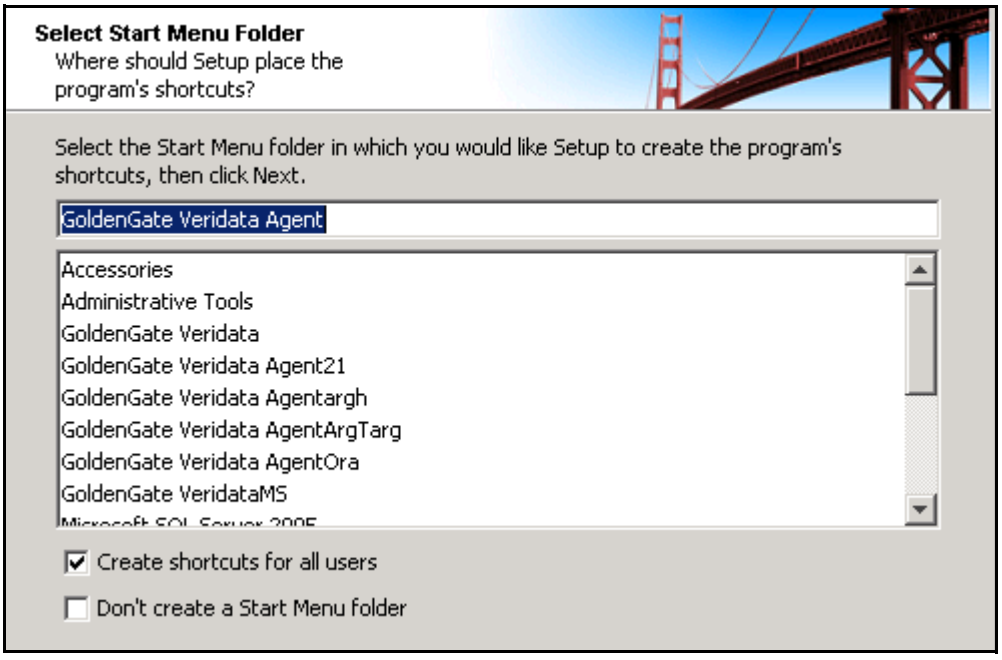


5. Select an installation directory. You can accept the default (to be created during installation), type a path name, or click **Browse** to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.

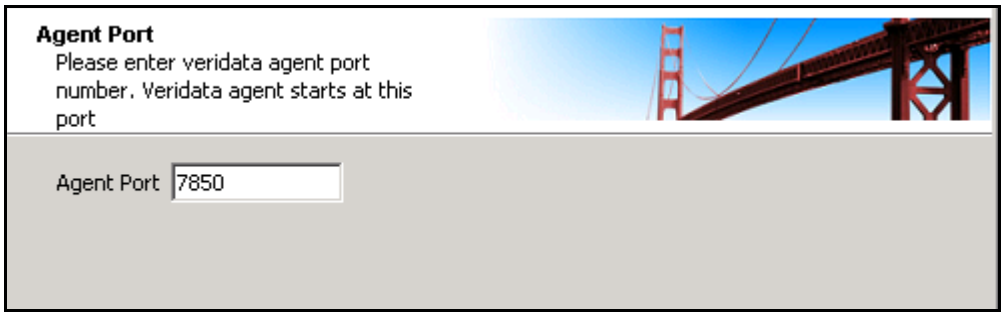


6. To add GoldenGate Veridata Agent to the **Programs** list of the **Start** menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list. To make the shortcut available to everyone who uses this system, select **Create Shortcuts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

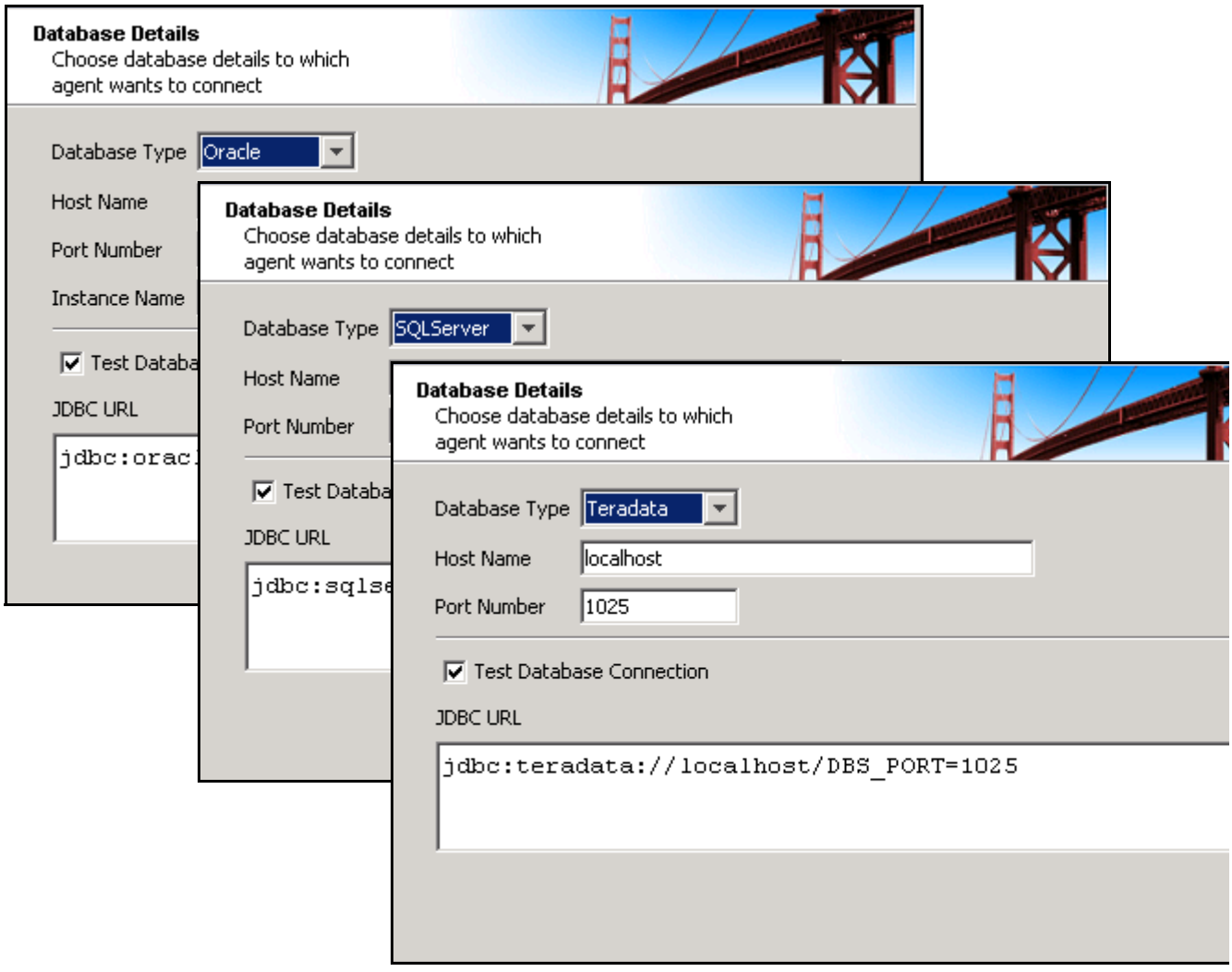
   To exclude GoldenGate Veridata Agent from the **Start** menu, select **Don't create a Start Menu folder**. If you are installing this software as a service, it can be started from the installation folder or the **Services** control panel.



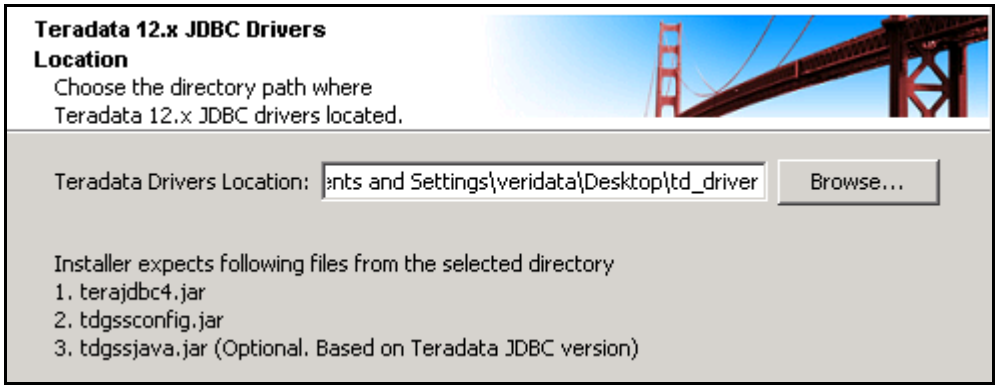
7. Specify a local port for the agent to use for communication with GoldenGate Veridata Server.



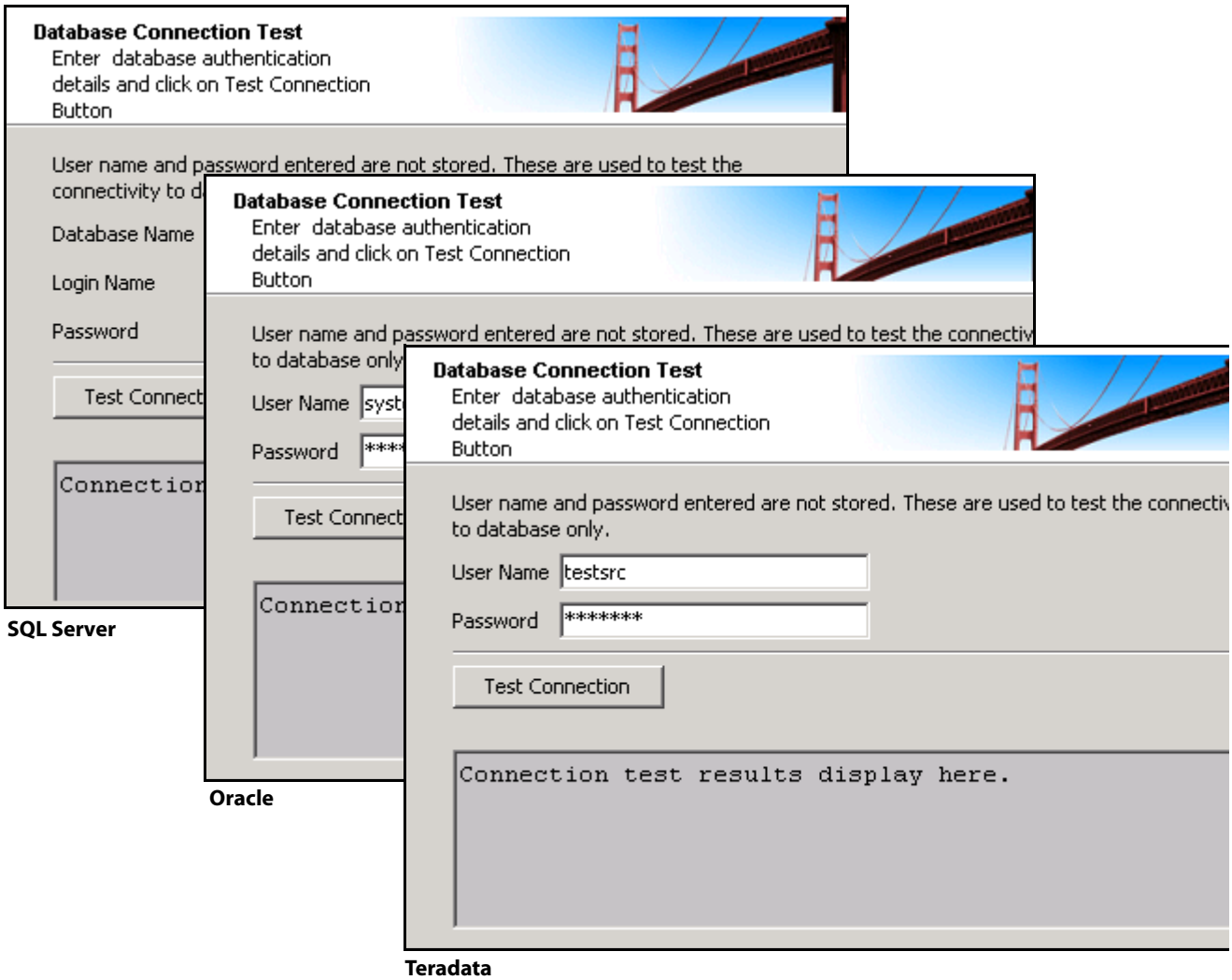
8. Select the database type, and specify the name of the host system, the port number that is used by the database, the name of the database (if applicable), and (if Oracle) the name of the instance to which the agent will connect. Under JDBC URL, the resultant connection string is displayed.



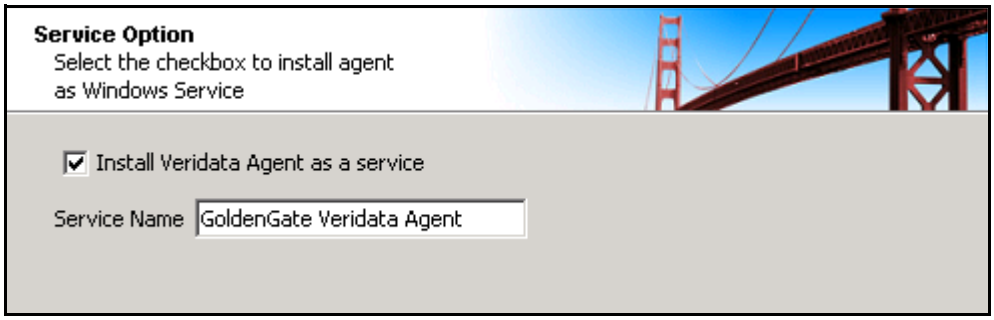
9. (Teradata) Supply the location of the Teradata JDBC driver.



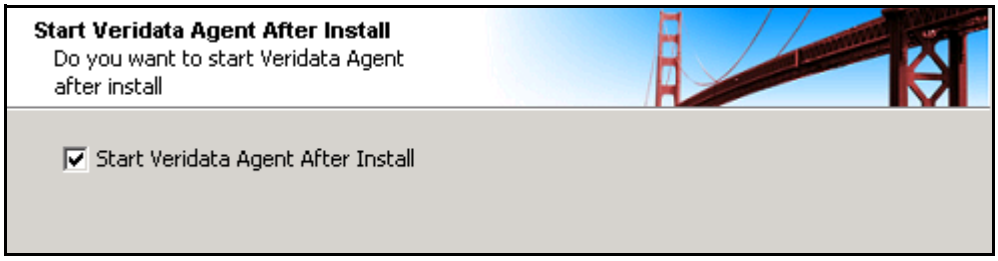
10. If you kept the default of **Test Database Connection**, a test screen prompts for authentication credentials that you created for the agent to use with this database or for an ODBC driver location, depending on the database type. Click **Test Connection**. Any connection errors are displayed in the output area.



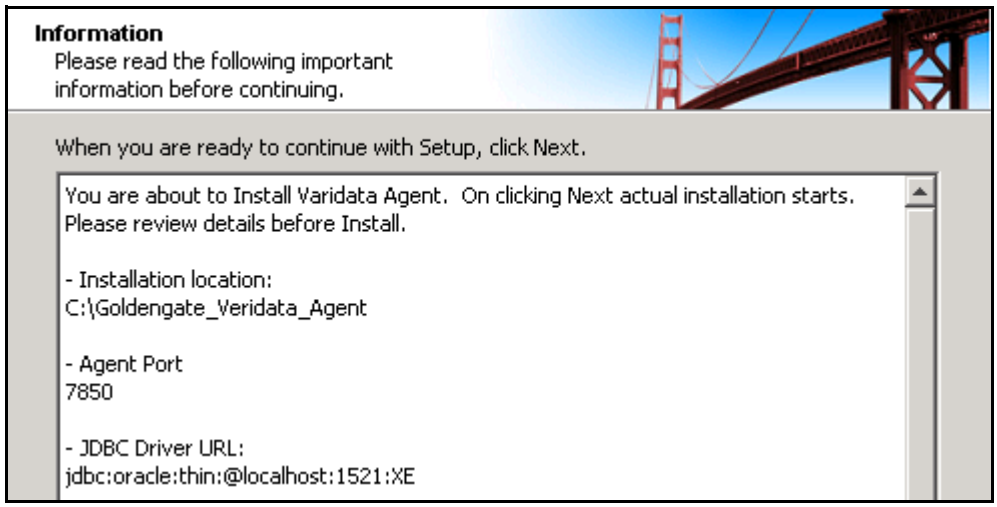
11. It is recommended that you install GoldenGate Veridata Agent as a Windows service. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up. Either accept the default 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.



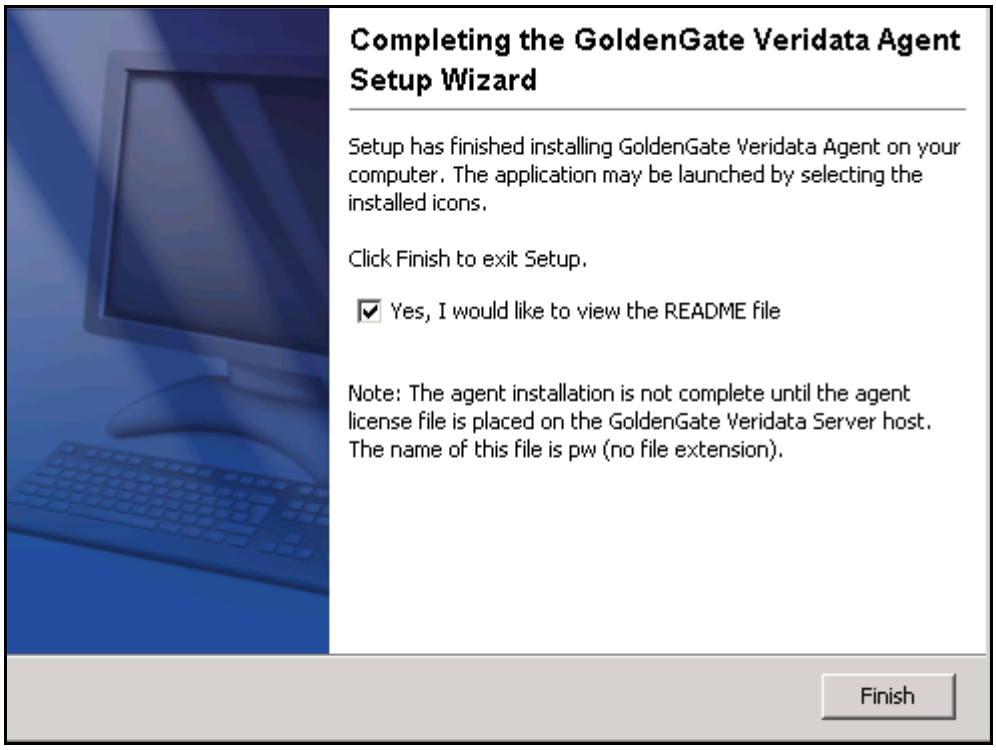
12. Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.



13. Review your installation selections.



14. Click **Back** to make changes or click **Next** to begin the installation.
15. Click **Finish** to exit the installer.



CHAPTER 4

Installing GoldenGate Veridata C-Agent

These instructions are for installing a new, clean copy of the C-based GoldenGate Veridata Agent software. To upgrade an existing installation, see Chapter 12 on page 91..

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

Installing the C-agent on a UNIX or Linux system

1. Create a directory for the agent.
2. Download the GoldenGate Veridata Agent build file for your platform from the GoldenGate support site.
3. FTP the file to the directory that you created.
4. Unzip the files with gunzip.

   ```
   gunzip <filename>.tar
   ```

5. Extract the files into the directory.

   ```
   tar -xf <filename>.tar
   ```

   The extraction creates an installation directory and a subdirectory named agent that contains the software files.
6. From the agent subdirectory, run GGSCI.
7. In GGSCI, issue the following command to create the working directories for GoldenGate Veridata.

   ```
   CREATE SUBDIRS
   ```

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

   ```
   EDIT PARAMS MGR
   ```

9. 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 GoldenGate software.

   ```
   PORT <number>
   ```

10. (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`.

11. Save and close the parameter file. This file is stored in the dirprm directory within the agent subdirectory. Do not move it.
12. 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
    ```

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

    ```
    INFO MGR
    ```

NOTE GoldenGate Veridata cannot be installed as a service on a UNIX or Linux system.

Installing the C-agent on a Windows system

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

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

Installing the GoldenGate Veridata Agent files

1. Create a directory for the agent.
2. Download the GoldenGate Veridata Agent build file for your platform from the GoldenGate support site to the directory you created.
3. Double-click the file to run your default Windows archiving/compression utility, or use any such utility of your choosing.

4. Extract the files 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.
5. From the agent subdirectory, run GGSCI.
6. In GGSCI, issue the following command to create the working directories for the agent.

   ```
   CREATE SUBDIRS
   ```

7. 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 page 35.

Specifying a custom Manager name

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

- you want to use a name for Manager other than the default of GGSMGR.
- there will be multiple Manager processes running as Windows services on this system, such as one for the GoldenGate replication software and one for GoldenGate Veridata. Each Manager on a system must have a unique name. Before proceeding further, verify the names of any local Manager services.

To specify a custom Manager name

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

   ```
   EDIT PARAMS ./GLOBALS
   ```

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

   ```
   MGRSERVNAME <name>
   ```

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

Installing Manager as a Windows service

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

To install Manager as a Windows service

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

   ```
   install <option> [...]
   ```

   Where: <option> is one of the following:

Table 7 INSTALL options

Option	Description
ADDEVENTS	Adds GoldenGate events to the Windows Event Manager. By default, GoldenGate errors are generic. To produce more specific error content, copy the following files from the GoldenGate installation directory to the SYSTEM32 directory. category.dll ggsmsg.dll
ADDSERVICE	Adds Manager as a service by the name specified in the GLOBALS file, if one exists, or by the default of GGSMGR. ADDSERVICE configures the service to run as the Local System account, the standard for most Windows applications because the service can be run independently of user logins and password changes. To run Manager as a specific account, use the USER and PASSWORD options.[1] The service is installed to start at system boot time (see AUTOSTART). To start it after installation, either reboot the system, or start the service manually from the Services applet of the Control Panel.
AUTOSTART	Specifies that the service created with ADDSERVICE is to be started at system boot time. This is the default unless MANUALSTART is used.
MANUALSTART	Specifies that the service created with ADDSERVICE is to be started manually through GGSCI, a script, or the Services applet of the Control Panel. The default is AUTOSTART.
USER <name>	Specifies a domain user account for executing Manager. For <name>, include the domain name, a backward slash, and the user name, for example HEADQT\GGSMGR. By default, the Manager service is installed to use the Local System account.
PASSWORD <password>	Specifies the password for the user specified with USER.

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

Configuring Manager

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

   ```
   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 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
   ```

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 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.

Installing the GoldenGate Veridata Agent files

1. Download the GoldenGate Veridata Agent build file from the GoldenGate support site to a Windows workstation.
2. 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 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 “Copying VSNSERV to remote nodes” and “Creating a GLOBALS file” as necessary for your environment.

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 GoldenGate Veridata agent subvolume. Otherwise, you might need to download the correct agent build for that hardware type from GoldenGate. It will include the correct VSNSERV.

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

- Install the entire 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. 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 GoldenGate Veridata Agent installation directory. See “Creating a GLOBALS file”.

Creating a GLOBALS file

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

- Other Manager processes exist on this system, such as the one used by 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 GoldenGate Veridata agent.
- Partitions for tables that will be compared with 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 SYSINFO on that node.

```
SYSINFO - 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 SYSnn    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.

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 MGRPARM
   ```

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 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
   ```

CHAPTER 5

Installing the server and web components using a MySQL repository

These instructions are for installing a new, clean copy of the GoldenGate Veridata Server software for use with a MySQL database repository. To upgrade an existing installation, see Chapter 12 on page 91.

Preinstallation system check

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 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 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.

   ```
   mysqld is alive.
   ```

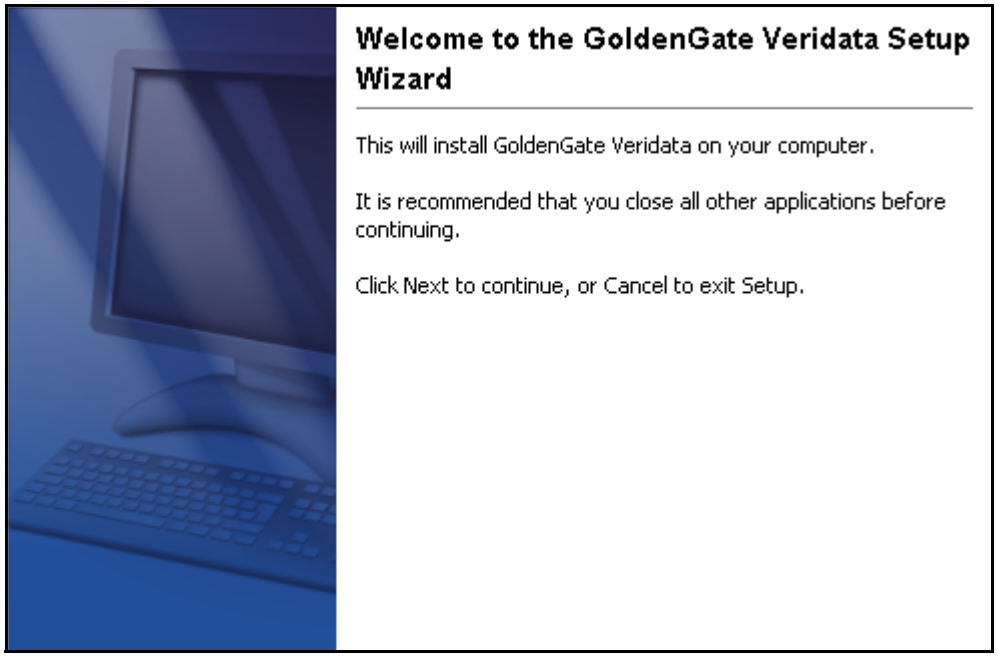
Installing the GoldenGate Veridata software

1. Start the database that you are using for the GoldenGate Veridata repository.
2. Close all applications that use this database.

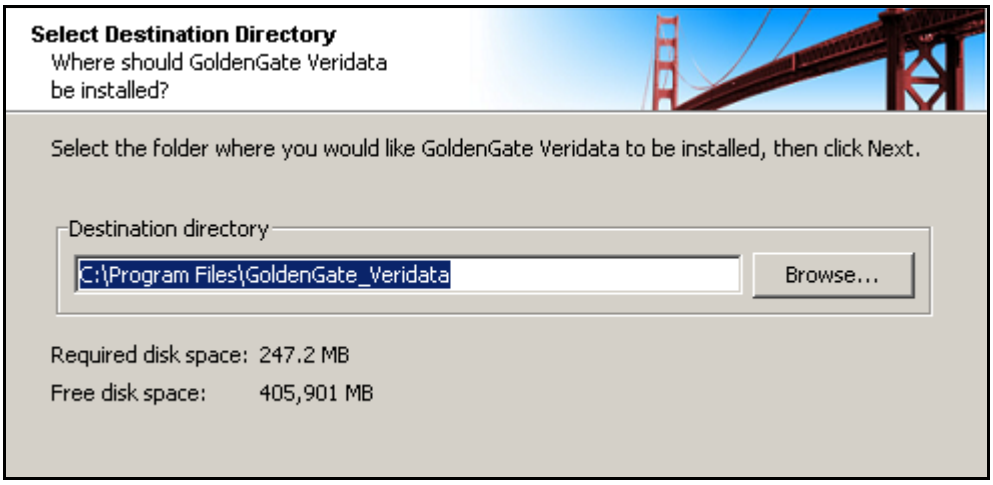
3. Go to the GoldenGate support site at http://www.goldengate.com/services/support.html. A valid user name and password is required.
4. Download the GoldenGate_Veridata_<platform><version>.exe file.
5. Run the installer.
   - **As a wizard**: Double click the program and then following the installation 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_Agent_<platform><version>.exe -c
     ```

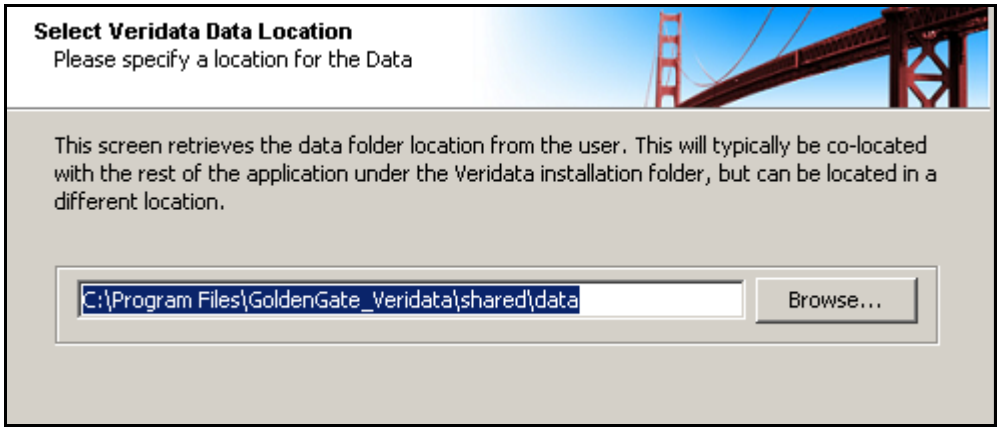
6. Click **Next** to start the installation.



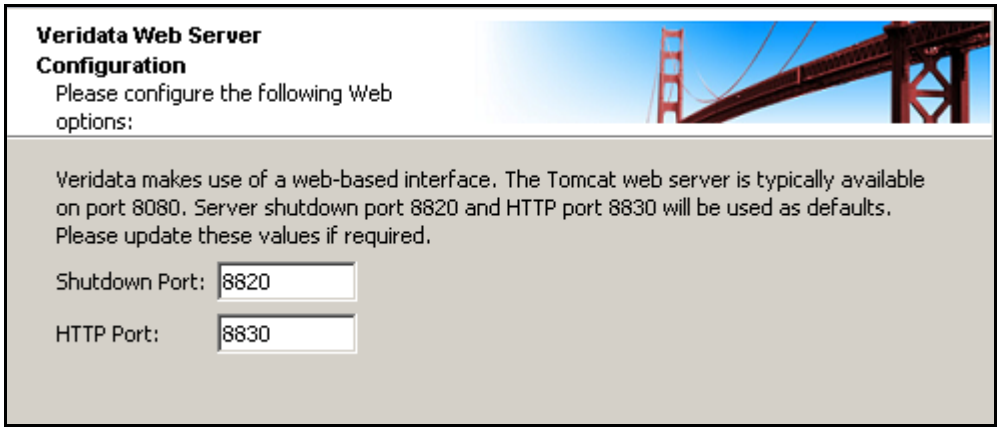
7. Select an installation directory. You can accept the default (to be created during installation), type a path name, or click **Browse** to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.



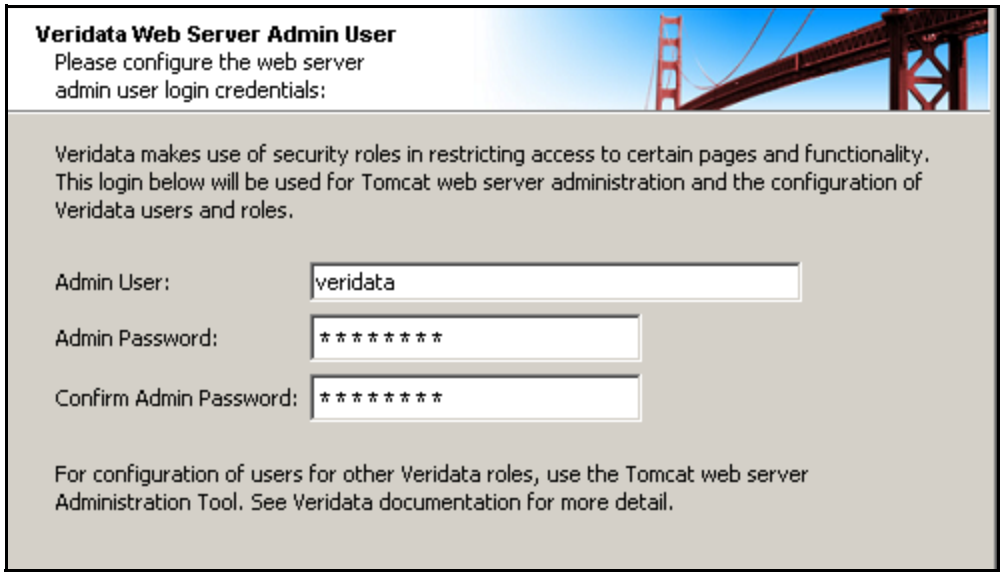
8. Select a location for the user data. The data directory is where GoldenGate Veridata stores reports and other files that contain information about out-of-sync rows. You can accept the default (to be created during installation), type a path name, or click **Browse** to go to a different directory.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



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

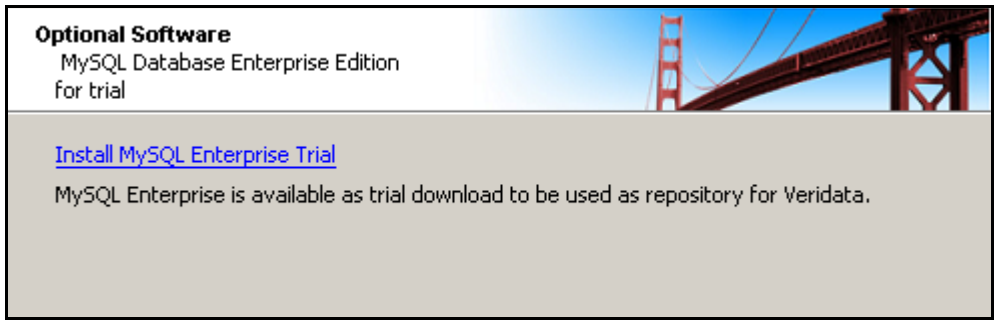


10. Specify an initial GoldenGate Veridata administrator user. The administrator is the most privileged role in GoldenGate Veridata and can perform all configuration, execution, and monitoring functions. Keep a record of this login. It is required to create other users after installation (see page 70).

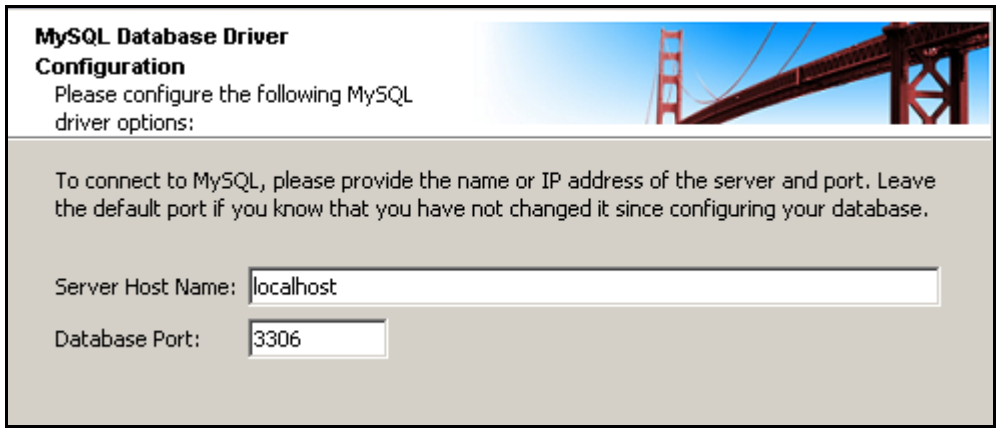


11. Select as the database that will be the GoldenGate Veridata repository.

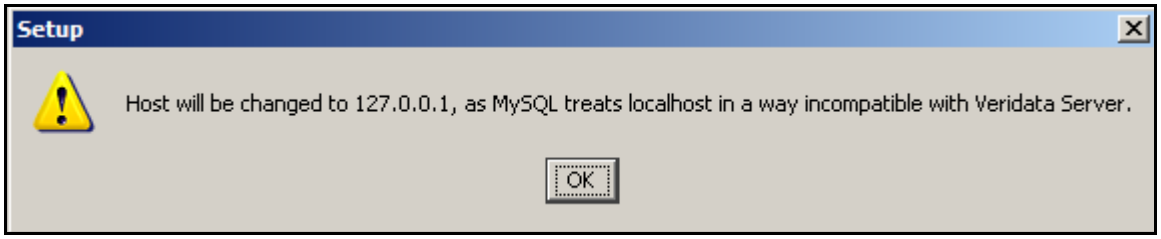
12. (Optional) To download and install a free trial version of MySQL for use as the GoldenGate Veridata repository, click the link to the MySQL website. Install MySQL and start it. You can leave the installer running while you do this.



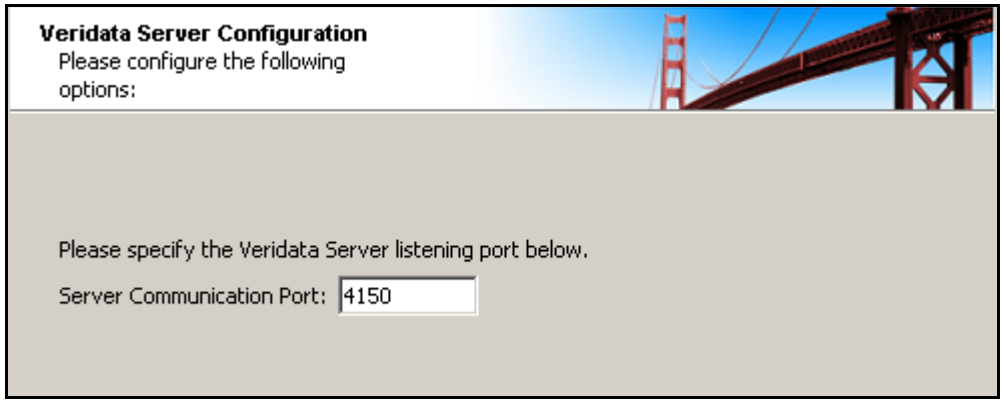
13. Specify the name or IP address of the system where you are installing the GoldenGate Veridata server and web components. Specify the database port number, if it is not the MySQL default port number.



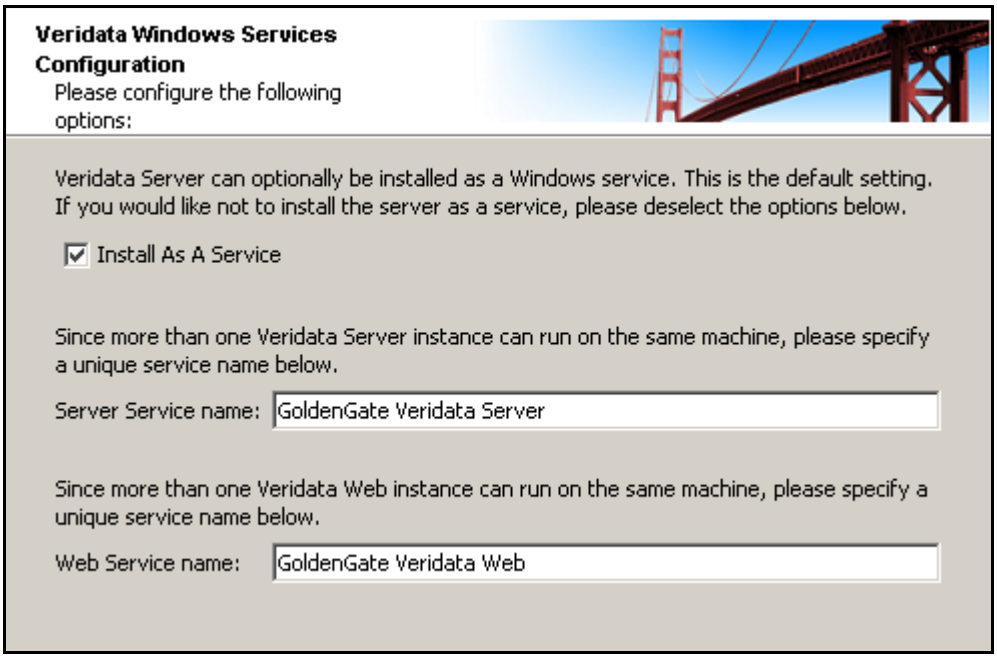
If you use **localhost** as the host name, the installer will convert it to the IP address and alert you with a message.



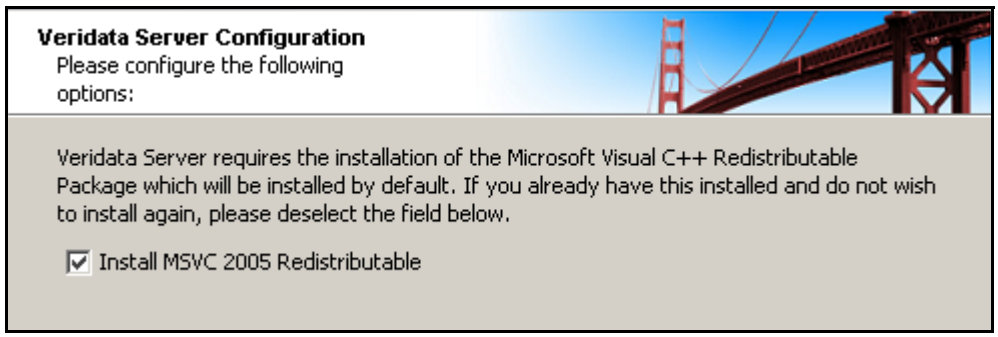
14. Specify a listening port for communication between the GoldenGate Veridata server and web components on this system, or accept the default port number.



15. It is recommended that you install the GoldenGate Veridata server and web components as Windows services. 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 names or change them 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.



16. Accept the default to install the Microsoft Visual C++ Redistributable Package unless the system contains a version of this package that is at least as current as the one shown by the installer.

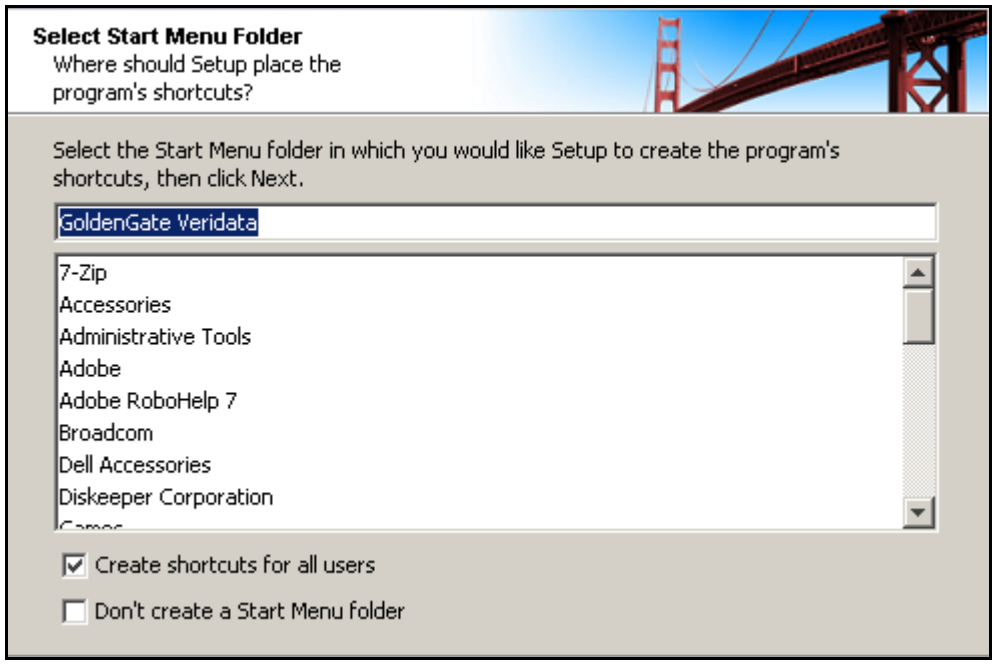


17. Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.



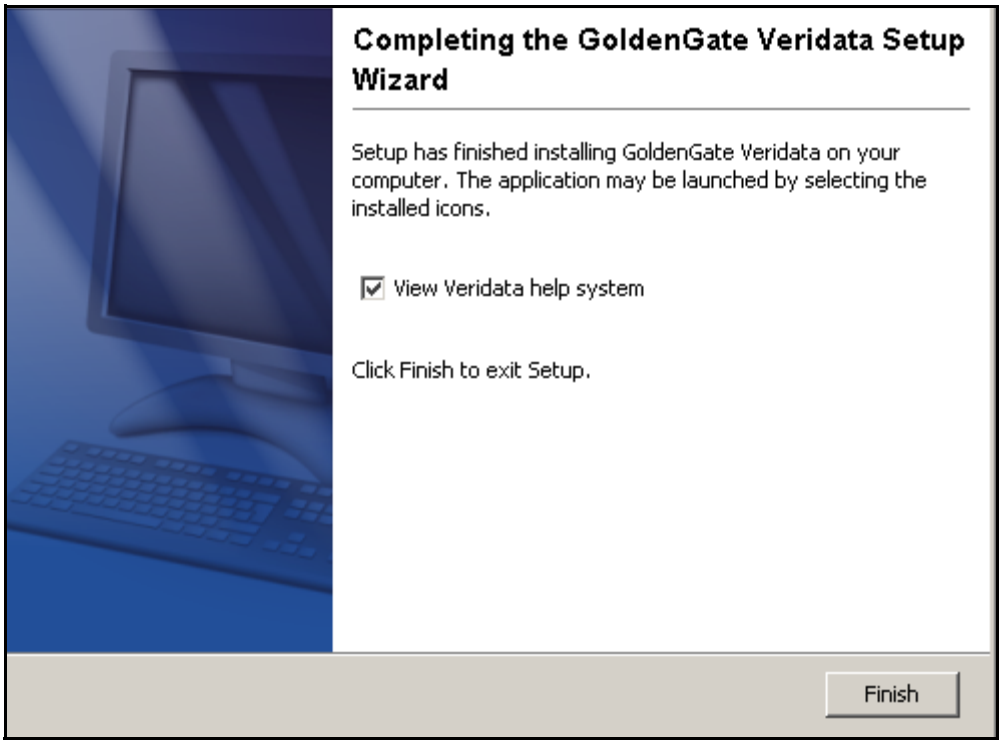
18. To add GoldenGate Veridata to the **Programs** list of the **Start** menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list. To make the shortcut available to everyone who uses this system, select **Create Shortcuts for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

To exclude 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.



19. Review your installation selections. Click **Back** to make changes or click **Next** to begin the installation.
20. To view the GoldenGate Veridata help system after the installation is done, accept the default. To prevent the help system from launching, clear the check box. The 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 GoldenGate users get started more quickly with a better understanding of the work flow and tools..



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

CHAPTER 6

Installing the server and web components using an Oracle repository

These instructions are for installing a new, clean copy of the GoldenGate Veridata Server software for use with an Oracle database repository. To upgrade an existing installation, see Chapter 12 on page 91.

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>
     ```

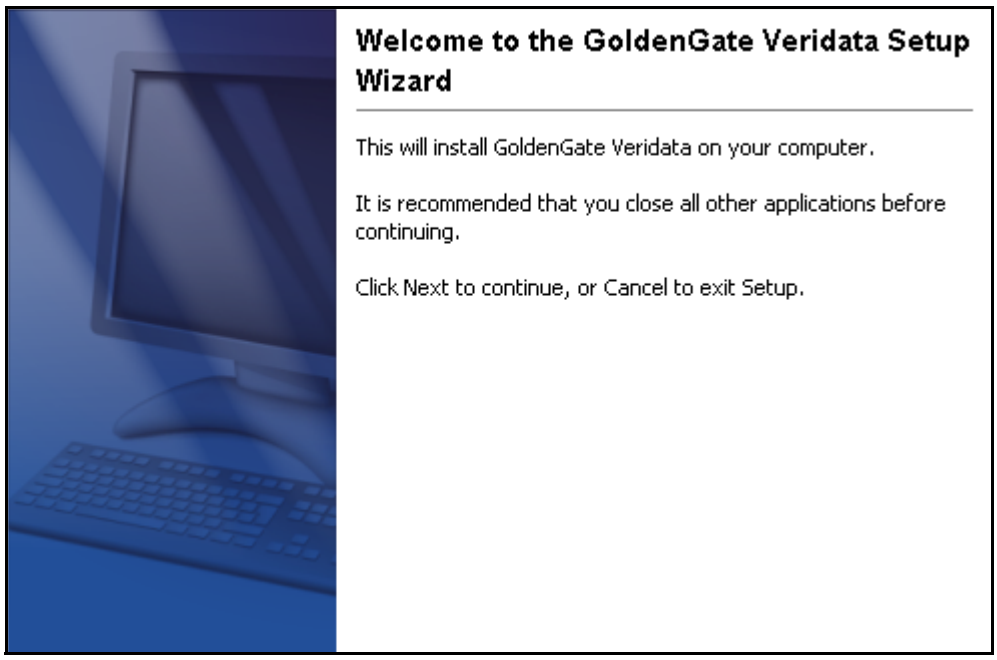
   The result should be a SQL*Plus command prompt. If you do not receive the SQL*Plus command prompt, it means that you cannot connect to the database by using a connect string. This problem must be fixed before running the installer.

Installing the GoldenGate Veridata software

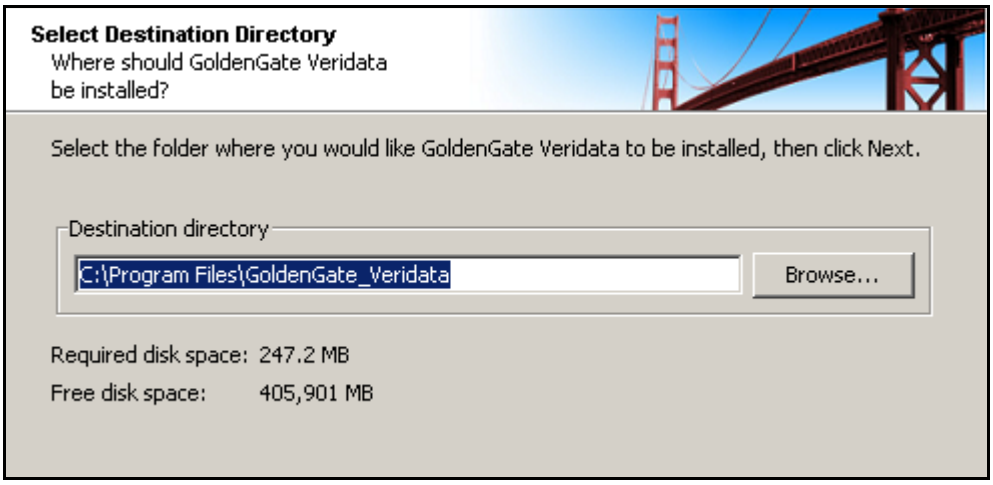
1. Start the database that you are using for the GoldenGate Veridata repository.
2. Close all applications that use this database.
3. Go to the GoldenGate support site at http://www.goldengate.com/services/support.html. A valid user name and password is required.
4. Download the GoldenGate_Veridata_<platform><version>.exe file.
5. Run the installer.
   - **As a wizard**: Double click the program and then following 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_Agent_<platform><version>.exe -c
     ```

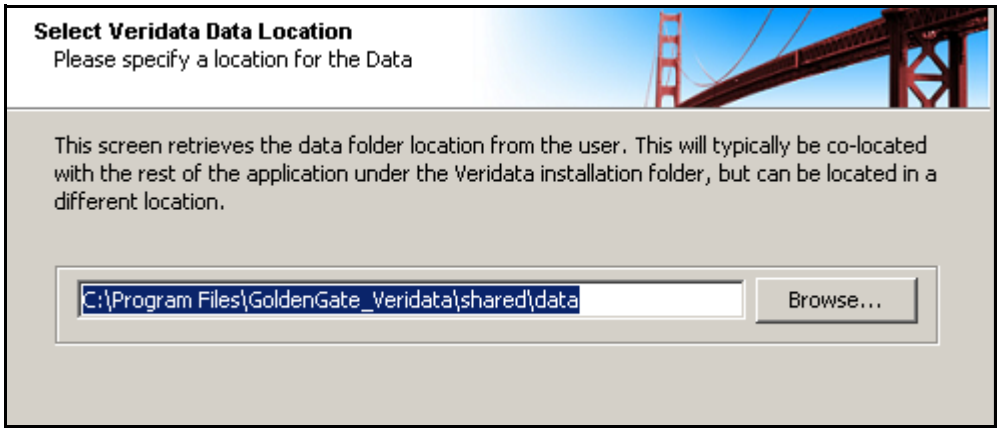
6. Click **Next** to start the installation.



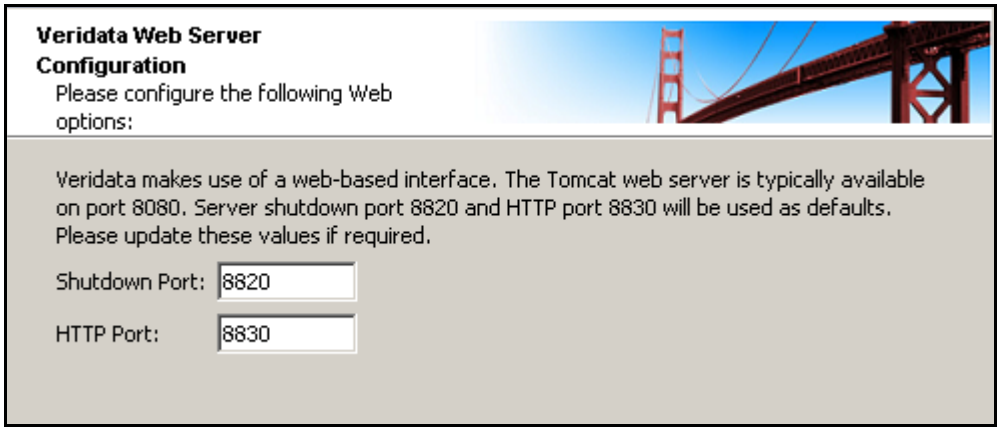
7. Select an installation directory. You can accept the default (to be created during installation), type a path name, or click **Browse** to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.



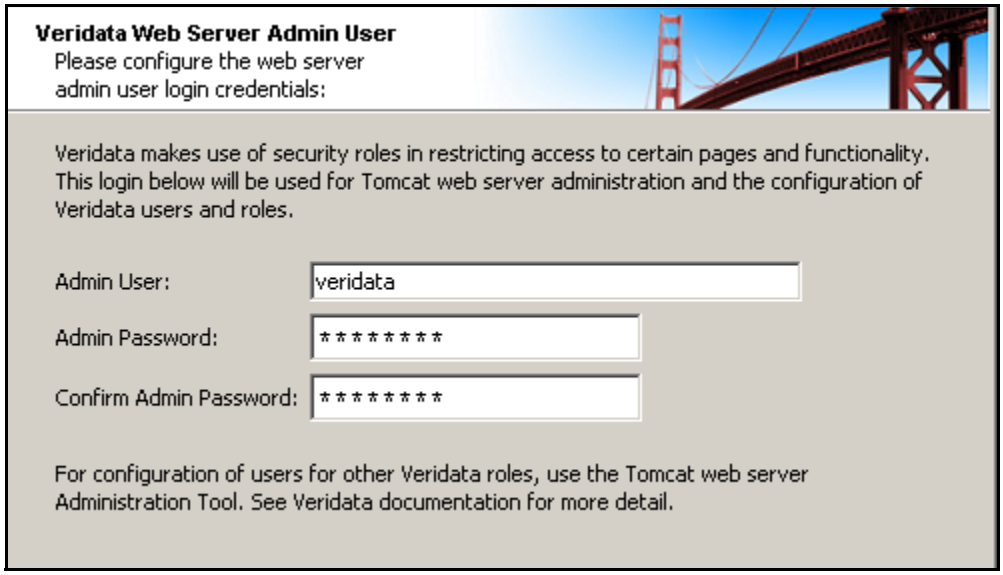
8. Select a location for the user data. The data directory is where GoldenGate Veridata stores reports and other files that contain information about out-of-sync rows. You can accept the default (to be created during installation), type a path name, or click **Browse** to go to a different directory.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



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



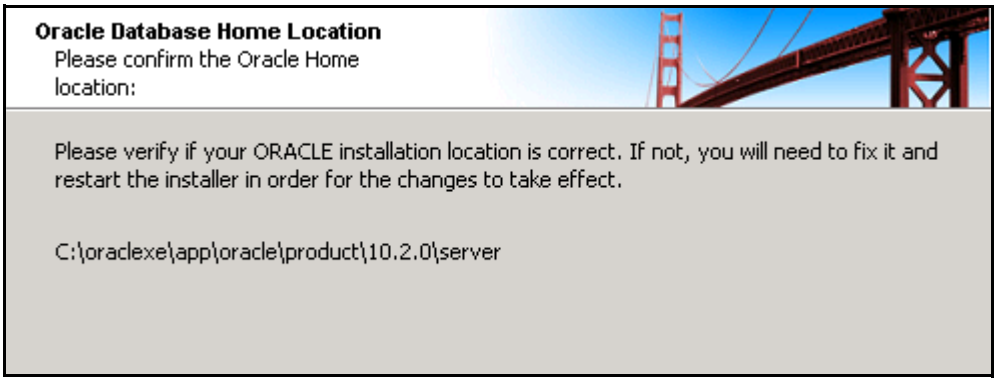
10. Specify an initial GoldenGate Veridata administrator user. The administrator is the most privileged role in GoldenGate Veridata and can perform all configuration, execution, and monitoring functions. Keep a record of this login. It is required to create other users after installation (see page 70).



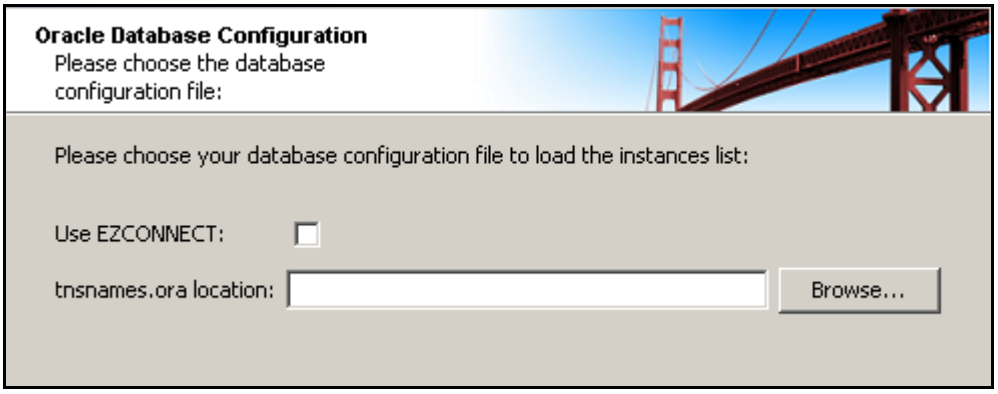
11. Select as the database that will be the GoldenGate Veridata repository.

12. One of the following screens appears:

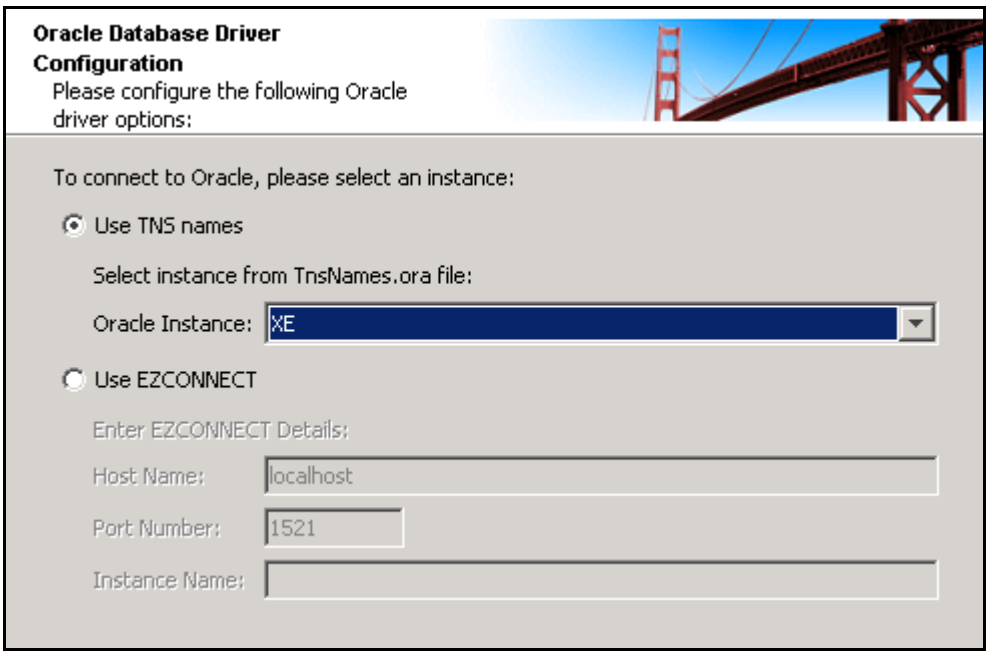
    - This screen appears if there is only one Oracle instance installed, or if you are installing on a UNIX system where the ORACLE_HOME environment variable is defined. Verify that this Oracle installation location is the one that belongs to the database that will be the GoldenGate Veridata repository. 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.



    - 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.



13. Specify the SQL*Net connection information to use.



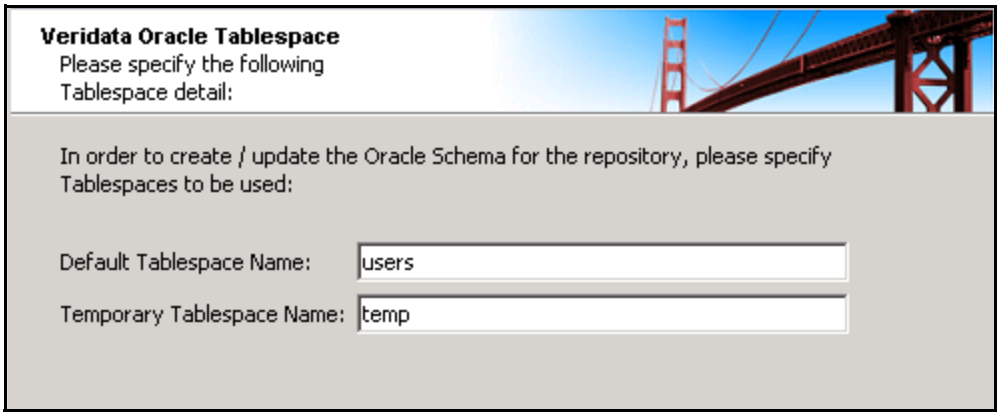
NOTE Depending on whether you forced tnsnames or EZCONNECT on the previous screen (if it appeared in your case), you may be permitted to select only one of these options.

- Select **Use TNS names** to have 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.

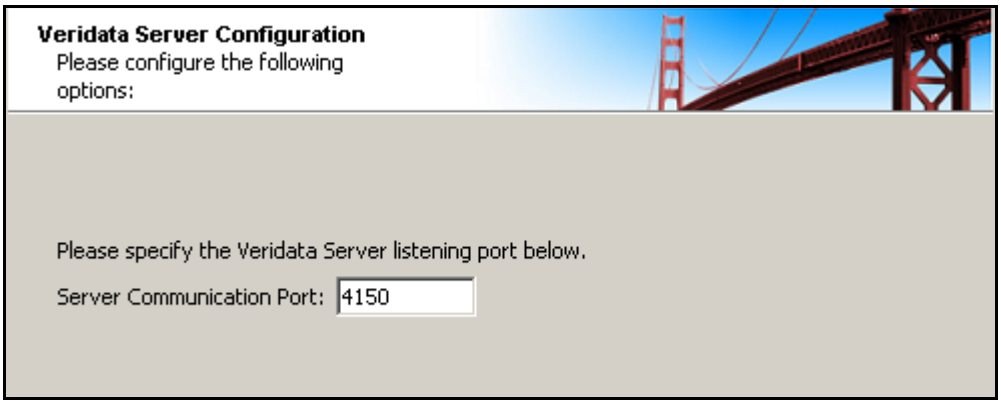
  NOTE On a UNIX system, if the installer is run from the command line instead of from a GUI, it may not display all of the Oracle instances in TNSNAMES.ORA, but only the first Oracle instance in the list. Type the correct instance name if it is not displayed.

- Select **Use EZCONNECT** to have 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.

14. (New user only) Specify default and temporary tablespaces for the GoldenGate Veridata repository.

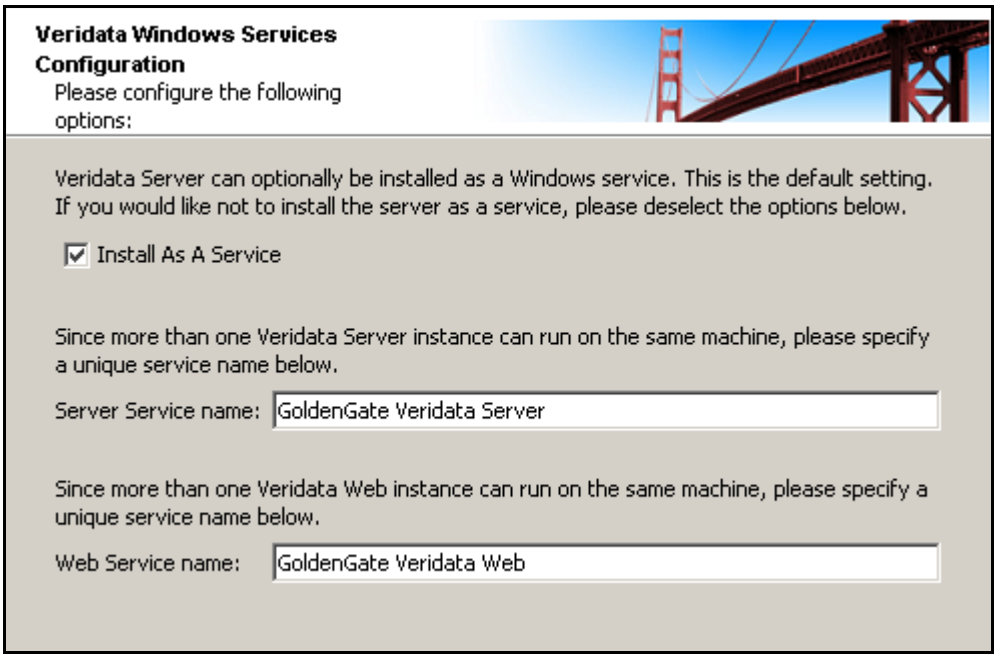


15. Specify a listening port for communication between the GoldenGate Veridata server and web components on this system, or accept the default port number.

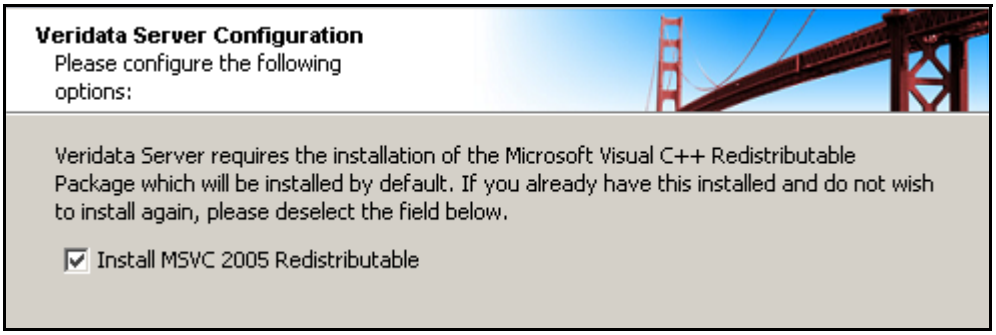


16. It is recommended that you install the GoldenGate Veridata server and web components as Windows services. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up. Either accept the default service

names or change them 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.



17. Accept the default to install the Microsoft Visual C++ Redistributable Package unless the system contains a version of this package that is at least as current as the one shown by the installer.

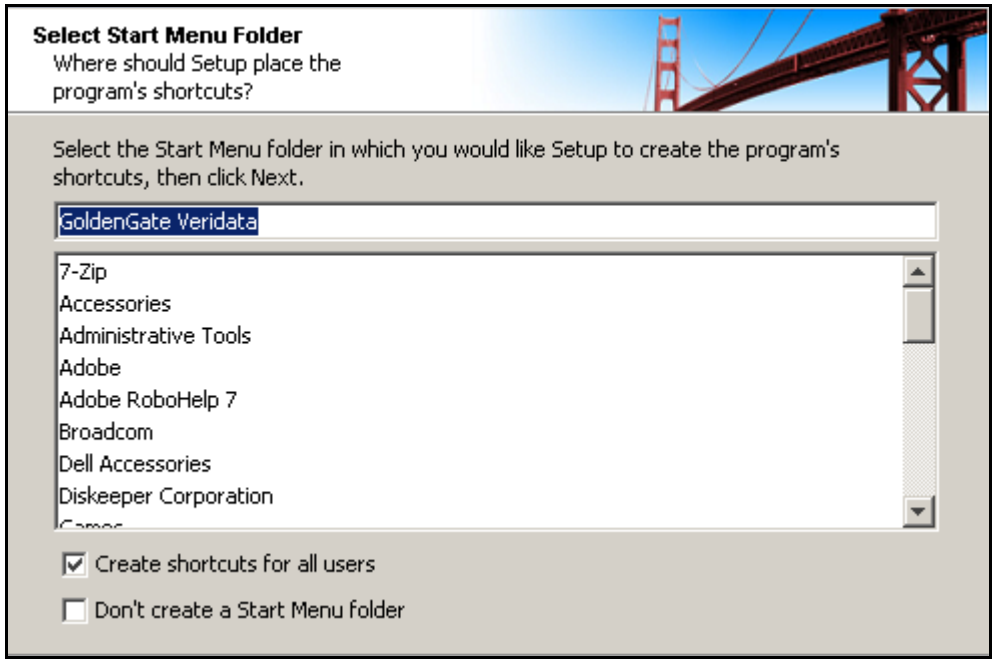


18. Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.



19. To add 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.

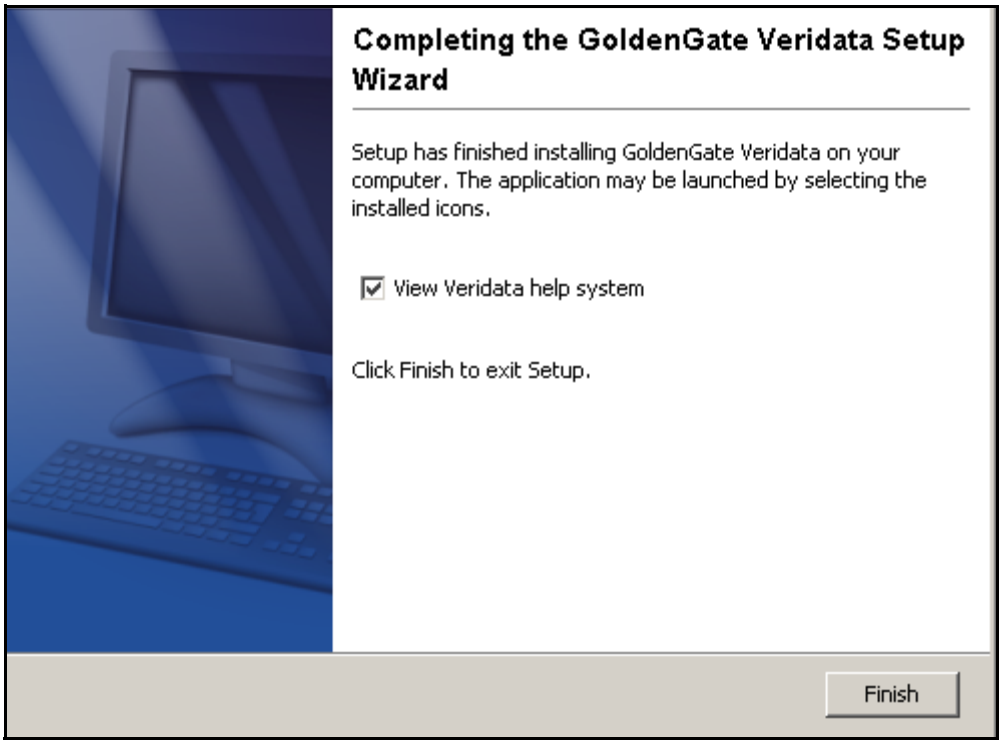
    To exclude GoldenGate Veridata from the **Start** menu, select **Don't create a Start Menu folder**. If you are installing this software as a service, it can be started from the installation folder or the **Services** control panel.



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

21. To view the GoldenGate Veridata help system after the installation is done, accept the default. To prevent the help system from launching, clear the check box. The 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 GoldenGate users get started more quickly with a better understanding of the work flow and tools..



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

CHAPTER 7

Installing the server and web components using a SQL Server repository

These instructions are for installing a new, clean copy of the GoldenGate Veridata Server software for use with a SQL Server database repository. To upgrade an existing installation, see Chapter 12 on page 91.

Preinstallation system check

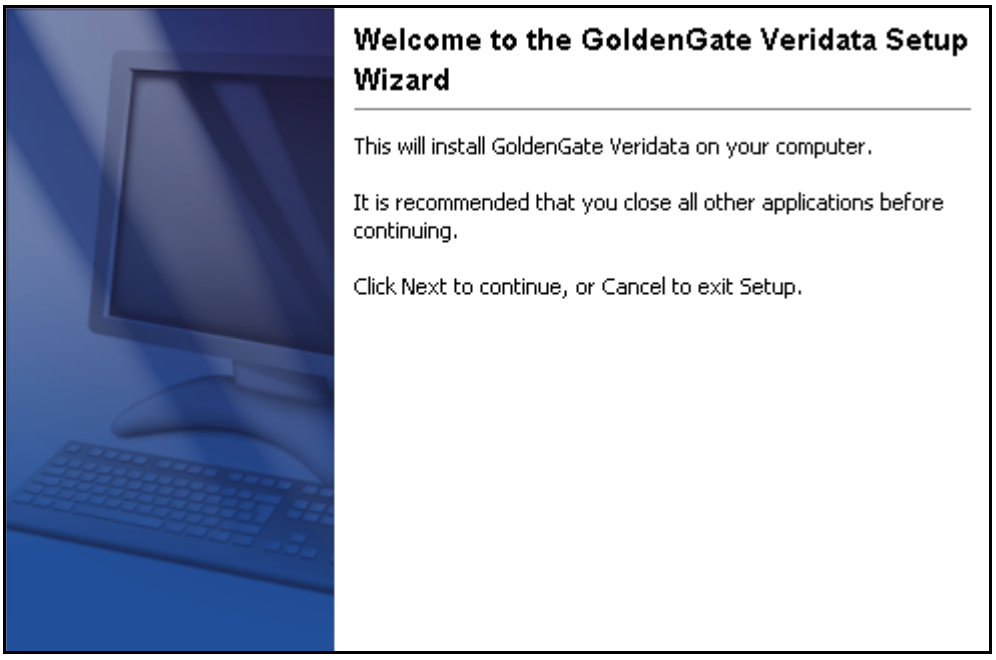
If you have not done so already, create a system data source (DSN) for GoldenGate Veridata to use to connect to the SQL Server instance. See page 106. If you have created a DSN already, you can use the same steps to verify the connection.

Installing the GoldenGate Veridata software

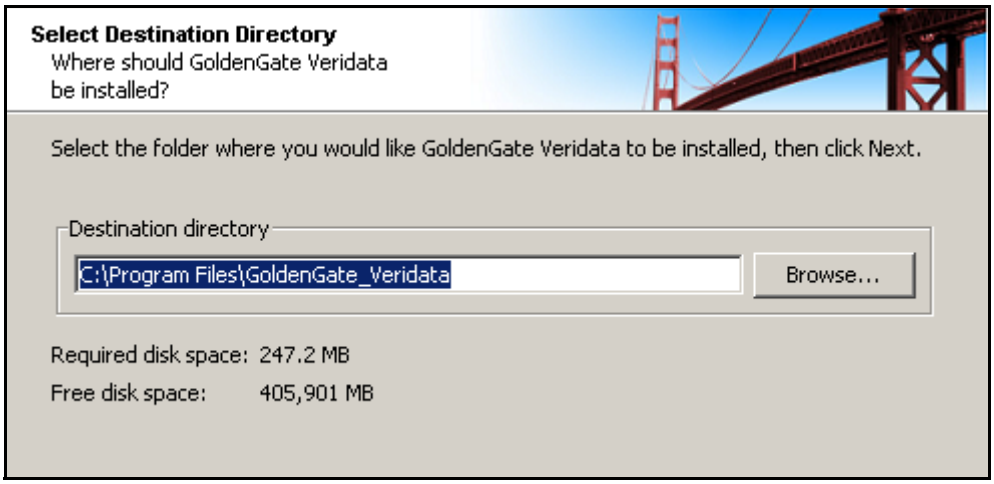
1. Start the database that you are using for the GoldenGate Veridata repository.
2. Close all applications that use this database.
3. Go to the GoldenGate support site at http://www.goldengate.com/services/support.html. A valid user name and password is required.
4. Download the GoldenGate_Veridata_<platform><version>.exe file.
5. Run the installer.
   - **As a wizard**: Double click the program and then following 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_Agent_<platform><version>.exe
     -c
     ```

6. Click **Next** to start the installation.

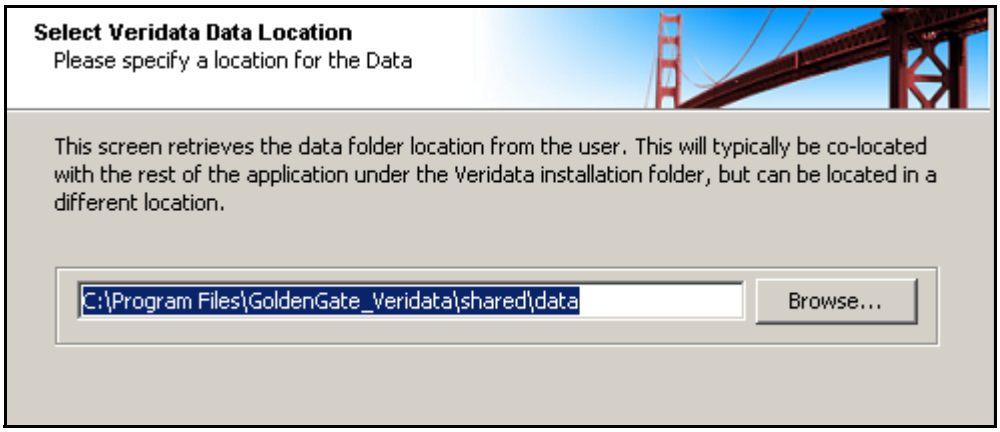


7. Select an installation directory. You can accept the default (to be created during installation), type a path name, or click **Browse** to go to a different directory. If installing more than one instance of this software, select a different directory or sub-directory each time that you run the installer.

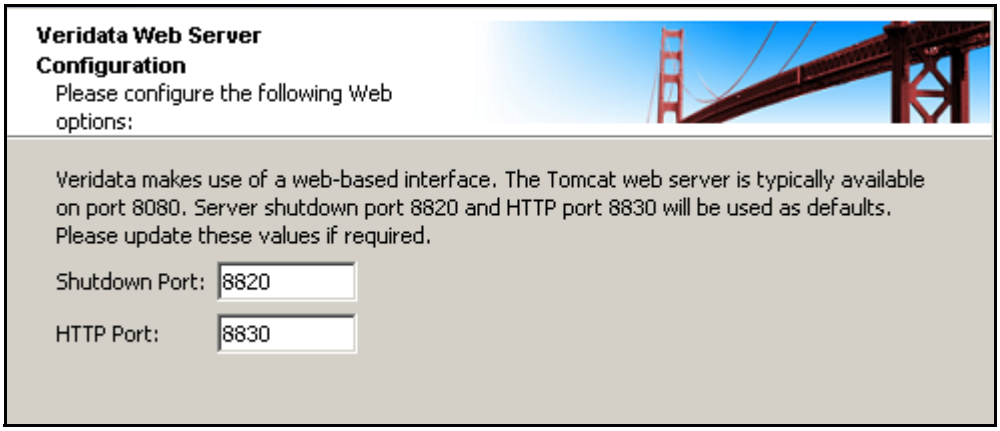


8. Select a location for the user data. The data directory is where GoldenGate Veridata stores reports and other files that contain information about out-of-sync rows. You can accept the default (to be created during installation), type

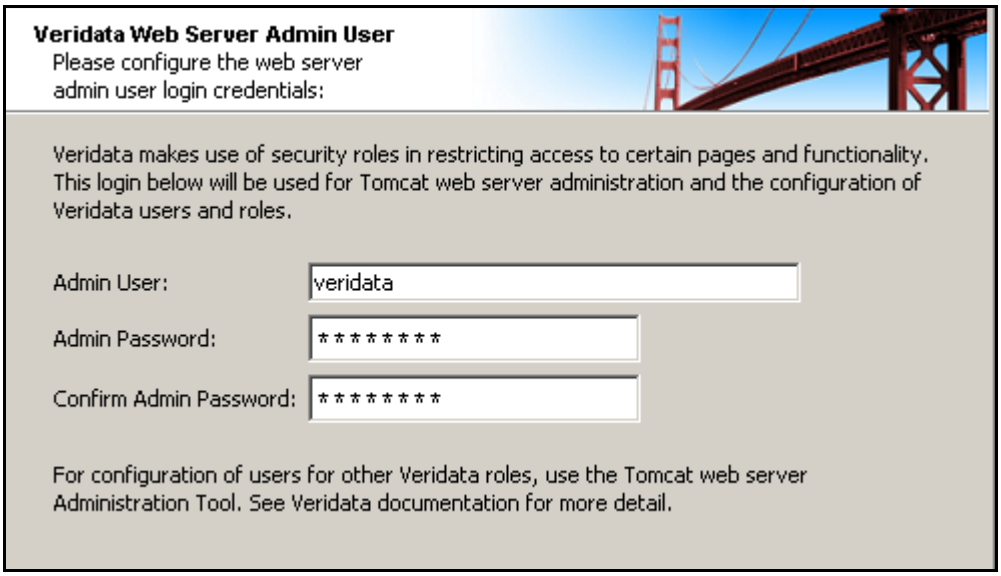
a path name, or click **Browse** to go to a different directory. This directory will contain data from the tables that are compared, so it should reside on a secure system.



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

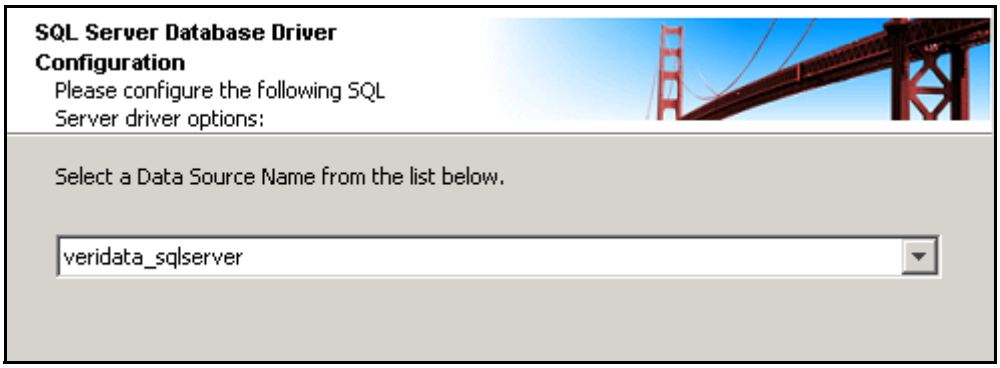


10. Specify an initial GoldenGate Veridata administrator user. The administrator is the most privileged role in GoldenGate Veridata and can perform all configuration, execution, and monitoring functions. Keep a record of this login. It is required to create other users after installation (see page 70).

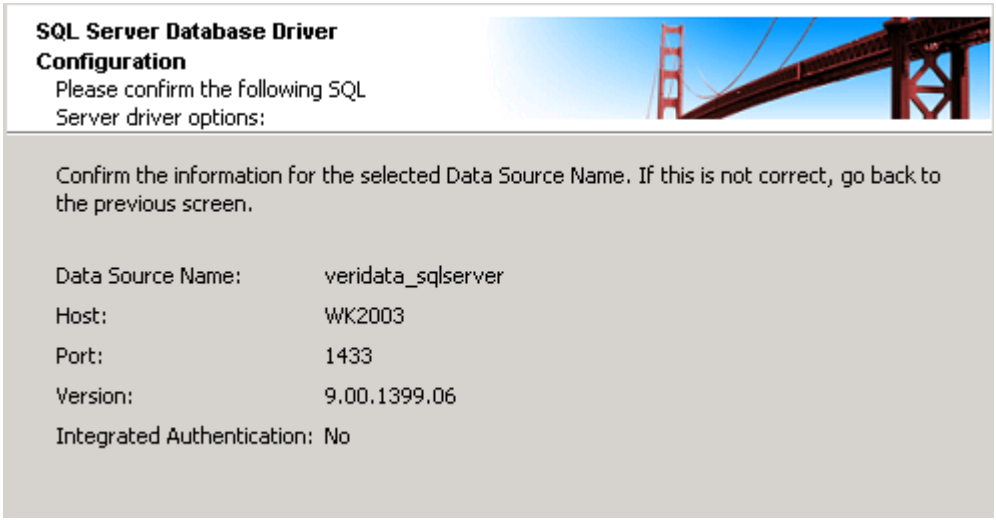


11. Select as the database that will be the GoldenGate Veridata repository.

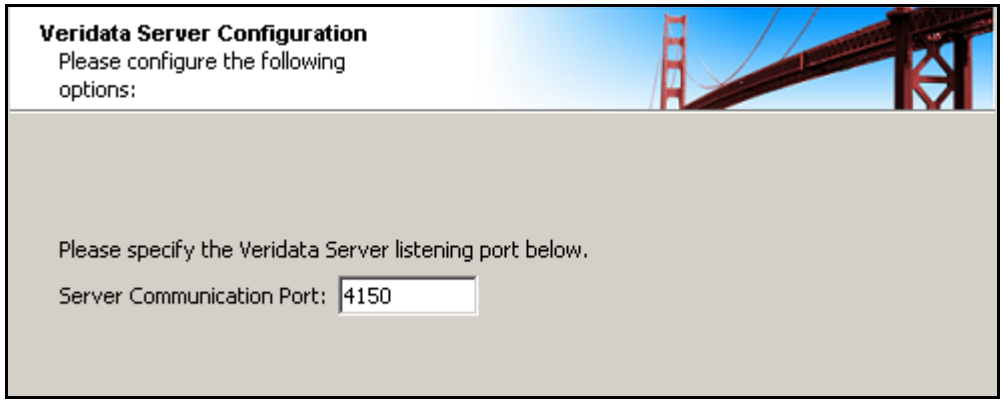
12. Specify the correct SQL Server ODBC data source name (DSN) for GoldenGate Veridata to use to connect to the repository database. If a DSN does not exist, or if no DSNs are displayed on this installer page, you can leave the installer running while you create the DSN, and then resume the installation afterward. Contact your system administrator or consult the documentation for your operating system if you are unable to create a DSN. For basic instructions, see page 106.



13. Confirm the DSN connection information.

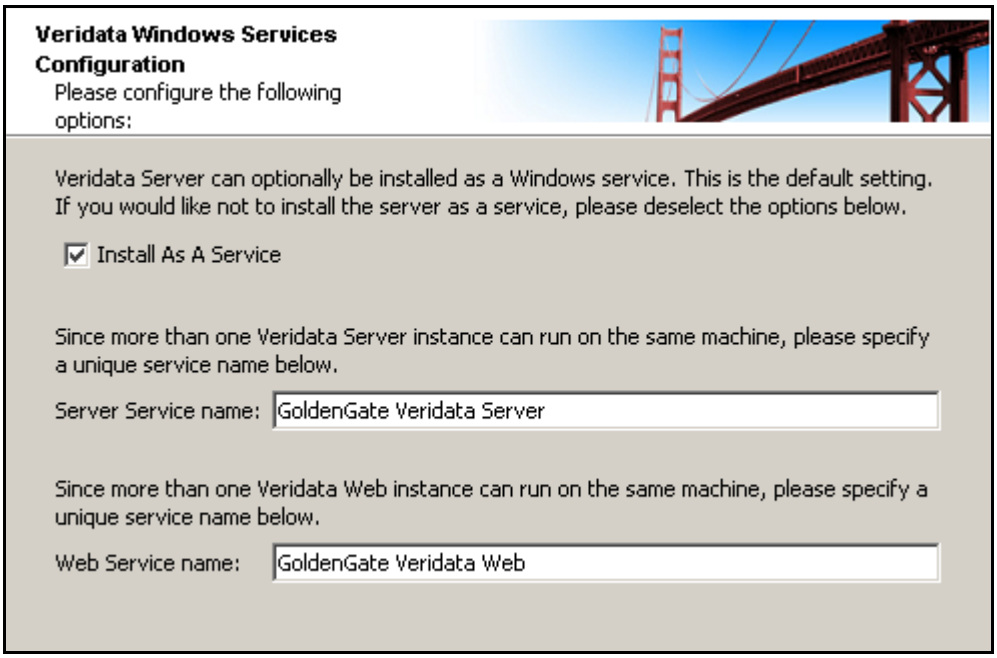


14. Specify a listening port for communication between the GoldenGate Veridata server and web components on this system, or accept the default port number.

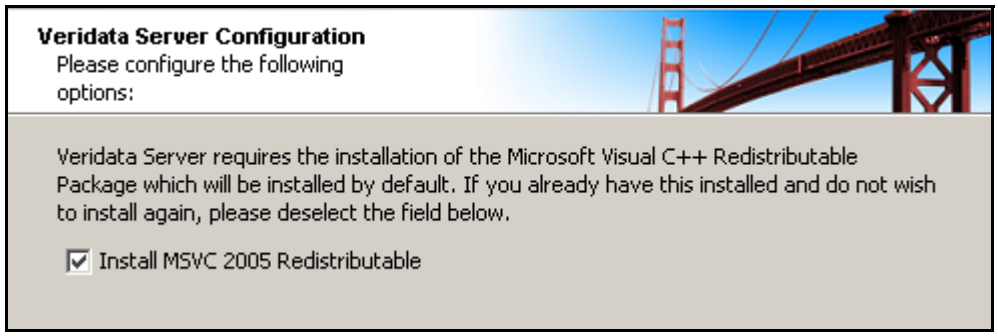


15. It is recommended that you install the GoldenGate Veridata server and web components as Windows services. When installed as a service, software can operate independently of user connections, which can accidentally be terminated, and you can configure it to start manually or at system start-up. Either accept the default service

names or change them 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.



16. Accept the default to install the Microsoft Visual C++ Redistributable Package unless the system contains a version of this package that is at least as current as the one shown by the installer.

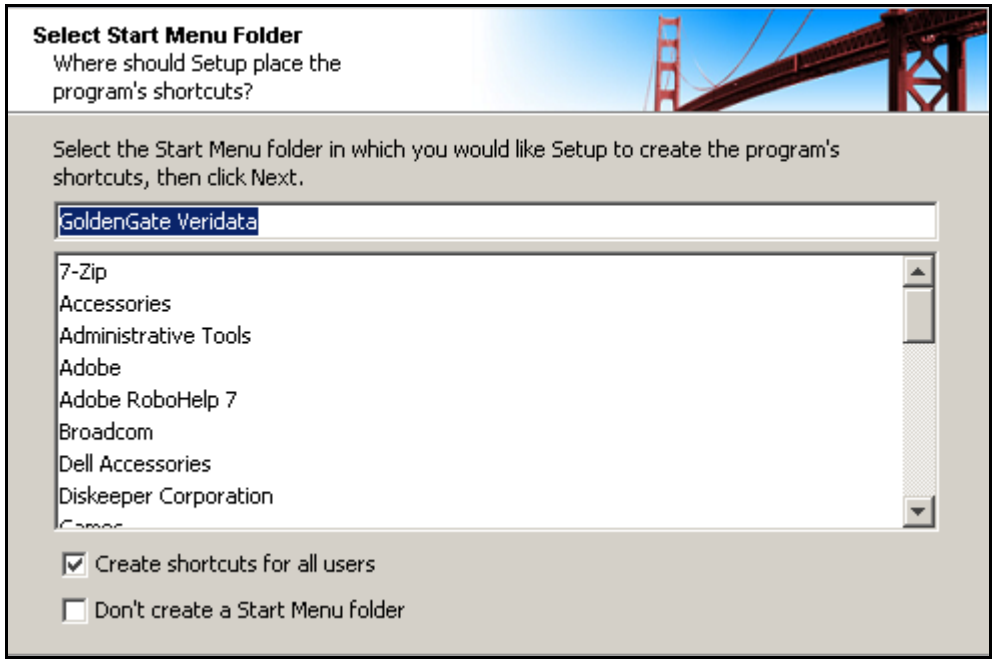


17. Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.



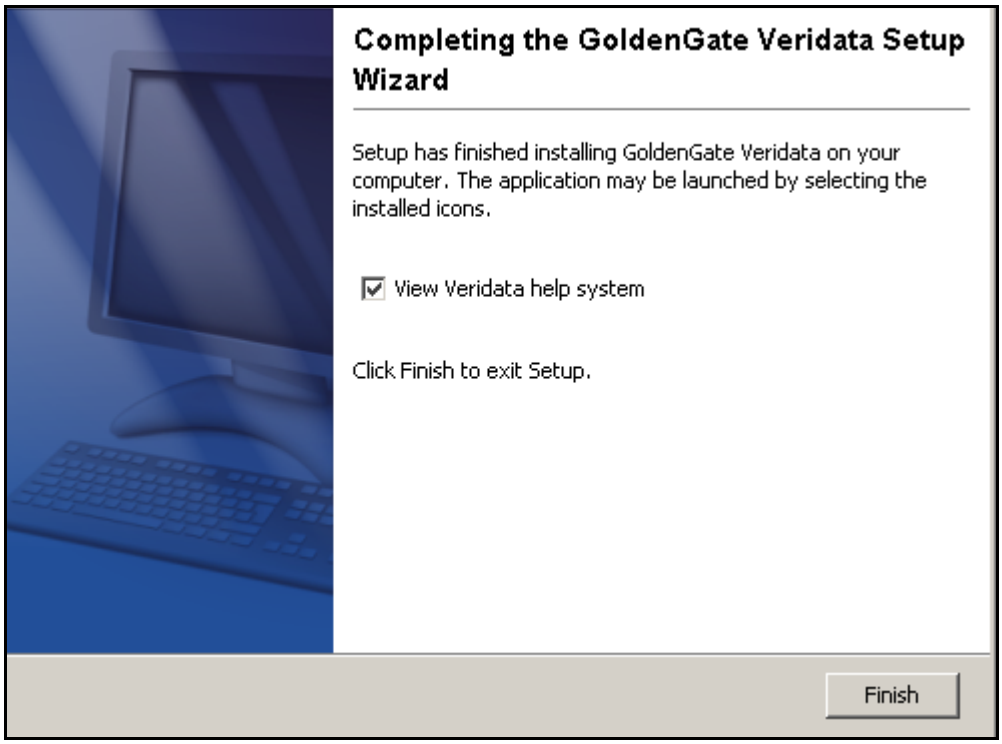
18. To add GoldenGate Veridata to the **Programs** list of the **Start** menu, you can accept the default shortcut name (to be created during installation), type a different one, or select one from the list. To make the shortcut available to everyone who uses this system, select **Create Shortcuts for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

    To exclude 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.



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

20. To view the GoldenGate Veridata help system after the installation is done, accept the default. To prevent the help system from launching, clear the check box. The 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 GoldenGate users get started more quickly with a better understanding of the work flow and tools..



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

22. The SQL Server repository must be running before the GoldenGate Veridata Server Windows service can start. The best practice is to 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.

CHAPTER 8

Configuring GoldenGate Veridata security

When using 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 GoldenGate Veridata installation directories
- Data files that contain the results of data comparisons
- The GoldenGate Veridata Web interface, where data values can be viewed

Securing the GoldenGate Veridata files

Controlling access to the installation directories

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

Securing files that contain business data

GoldenGate Veridata Server creates data files that will contain sensitive application data. By default, these files reside in the shared/data directory within the GoldenGate Veridata Server installation directory, but the person who installed the GoldenGate 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 GoldenGate Veridata Server installation program. Do not change the permissions, or 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 GoldenGate Veridata Web do not require access to these files because they see the same information through the client interface.

Securing access to GoldenGate Veridata Web

You can assign security roles to the users of 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 GoldenGate Veridata. This role can perform all of the functions that configure, execute, and monitor GoldenGate Veridata.
- **PowerUser**: The power user role is the second-highest role in GoldenGate Veridata. This role can perform all of the functions that configure, execute, and monitor GoldenGate Veridata from the GoldenGate Veridata Web interface, but this role cannot perform any configuration functions for the GoldenGate Veridata Server.
- **ReportViewer**: The report viewer role cannot perform functions that configure 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 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 GoldenGate Veridata Web 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.

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

2. Log on to the Apache Tomcat Web Server Administration Tool as a GoldenGate Veridata administrator user. A default administrator user was created during the installation of GoldenGate Veridata.
3. In the navigation pane, 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 "Securing access to 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 "Securing access to 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.

CHAPTER 9

Running the GoldenGate Veridata Programs

You can start the GoldenGate Veridata programs in any order.

Starting and stopping the C-agent and Manager

The C-agent starts automatically at the request of GoldenGate Veridata Server when initiating comparisons. However, for 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 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 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.

Starting and stopping the Java-based components

The GoldenGate Veridata Server and GoldenGate Veridata Web components are Java-based programs. The 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.

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:

UNIX and Linux

```
agent.sh {start | run}
Or...
agent.sh stop
```

Windows

```
agent.bat {start | run}
Or...
agent.bat stop
```

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

UNIX and Linux

```
veridata_server.sh start
Or...
veridata_server.sh stop
```

Windows

```
veridata_server.bat start
Or...
veridata_server.bat stop
```

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

UNIX and Linux

```
veridata_web.sh start
Or...
veridata_web.sh stop
```

Windows

```
veridata_web.bat start
Or...
veridata_web.bat stop
```

Configure the host to start and stop the processes automatically. Contact your

system administrator if you need assistance.

NOTE If you are restarting GoldenGate Veridata Server, you must also restart GoldenGate Veridata Web.

Controlling the Java-based components as Windows services

A 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, a 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 GoldenGate Veridata shortcuts.
   - To start the server component: **Application**>**Start Veridata Server**
   - To stop the server component: **Application**>**Stop Veridata Server**
   - To start the web component: **Application**>**Start Veridata Web**
   - To stop the web component: **Application**>**Stop Veridata Web**.
   - To start the agent component: **Application**>**Start Veridata Agent**
   - To stop the agent component: **Application**>**Stop Veridata Agent**

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:

UNIX and Linux	**Windows**
`agent.sh reloadlog`	`agent.bat reloadlog`

Connecting to GoldenGate Veridata Web

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

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

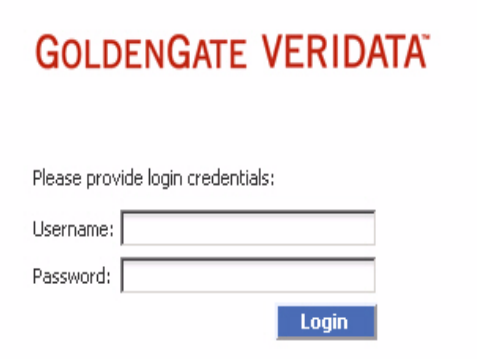
  `http://<hostname>:<port>/veridata`

Where: <hostname> is the name of the system where 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 GoldenGate Veridata Web login page is displayed upon successful connection.



Log in with your user name and password. For full instructions on using GoldenGate Veridata Web, see the online help.

CHAPTER 10

Running comparisons from the command line

Overview of the command line interface

You can use the vericom tool of GoldenGate Veridata to execute certain comparison tasks from the command shell of the operating system. The vericom tool runs the 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.
- Shut down GoldenGate Veridata Server
- Set tracing (only under guidance of GoldenGate Support)

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 GoldenGate Veridata Web interface. This interface provides greater control in configuring the objects to be compared and for controlling runtime parameter settings.

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 GoldenGate Veridata Server is installed, run the command shell of the operating system.
2. Change directories to server/bin within the GoldenGate Veridata Server installation directory.
3. Use the following syntax to run the vericom program.

Syntax `vericom -<required input> -<optional input>`

Required input

One of the following are required; otherwise an error is returned.

```
-shutdown |
-help |
-helprun |
-{version | v} |
-{job | j} <job>
```

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

Optional input

```
[ -g <group> -c <compare_pair> ]
[ -rP <profile> ]
[ -rR ]
| -rO ]
[ -rN <threads> ]
[ -rD <seconds> ]
[ -rC | +rC ]
[ -rOb | -rOx | -rO2 | -rO0 ]
[ -rOs <records> ]
[ -rTi ]
[ -rTc ]
[ -rTs <trace_number> ]
[ -pS <source_partition_name> |
   -pSq <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> ]
[ -w | { -wp <poll_interval> } ]
[ -rd0 | -rdN <run_ID> ]
[ -{url | u} <url> ]
[ -{password | p} <password> ]
```

Table 8 Vericom runtime arguments

Argument	Description
`-shutdown`	Shuts down the GoldenGate Veridata Server processes.
`-help`	Displays the vericom syntax components and their descriptions.
`-helprun`	Displays run-related syntax components and their descriptions.
`-{version | v}`	Displays the version of the GoldenGate Veridata command-line interface that is being used.
`-{job | j} <job>`	Specifies the job to be run. For <job>, specify the name that was assigned when the job was created in GoldenGate Veridata Web.
`-g <group>` `-c <compare_pair>`	Specifies a group and compare pair. For <group> and <compare_pair>, specify the names that were assigned when these objects were created in GoldenGate Veridata Web. ◆ If -g and -c are used, -j must also be used.
`-rP <profile>`	Overrides the profile that is defined for a job. For <profile>, specify the name that was assigned when the profile was created in GoldenGate Veridata Web. ◆ If -rP is used, -j must be used.
`-rR`	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. ◆ Do not use -rR and -rO in the same run.
`-rO`	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. ◆ -rO must be used with -j. ◆ Do not use -rR and -rO in the same run.
`-rN <threads>`	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 -j or if just one comparison is run by using -j with -g and -c.

Table 8 Vericom runtime arguments (continued)

Argument	Description
`-rD <seconds>`	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 -rR option is used.
`-rC | +rC`	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.
`-rOb | -rOx | -rO2 | -rO0`	Controls the kind of file that is produced for the out-of-sync report. ◆ -rOb generates binary format that is compatible with the 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.
`-rOs <records>`	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.
`-rTi`	Turns on tracing of GoldenGate Veridata Agent for the initial comparison step. Do not use without the guidance of a GoldenGate support analyst.
`-rTc`	Turns on tracing of GoldenGate Veridata Agent for the confirmation step. Do not use without the guidance of a GoldenGate support analyst.

Table 8 Vericom runtime arguments (continued)

Argument	Description
`-rTs <trace_number>`	Turns on tracing for 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 a GoldenGate support analyst.
`-pS <source_partition_name> | -pSq <source_sql_predicate> | -pSA1 <source_ascii_start_key> | -pSA2 <source_ascii_end_key> | -pSH1 <source_hex_start_key> | -pSH2 <source_hex_end_key>`	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. **-pS <source_partition_name>** 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. **-pSq <source_sql_predicate>** Specifies a SQL predicate that defines a partition to override an existing source partition for a SQL table. The predicate is the conditional statement that follows the WHERE keyword, for example: LAST_NAME BETWEEN "A" AND "M". Do not include the WHERE keyword. It will be added automatically at runtime. If the predicate contains multiple words, it must be enclosed within quotes to make it a single command argument. The type of quote is dependent on the command shell or interpreter that is being used. If the predicate contains special characters (such as $, *, < in sh/csh or %, < in Windows), they must be properly escaped for that shell or interpreter. **-pSA1 <source_ascii_start_key>** Specifies an ASCII key as the starting key value of a partition that overrides an existing source partition for an Enscribe file. **-pSA2 <source_ascii_end_key>** Specifies an ASCII key as the ending key value of a partition that overrides an existing source partition for an Enscribe file.

Table 8 Vericom runtime arguments (continued)

Argument	Description
	-pSH1 <source_hex_start_key> Specifies a hexadecimal key as the starting key value of a partition that overrides an existing source partition for an Enscribe file. **-pSH2 <source_hex_end_key>** Specifies a hexadecimal key as the ending key value of a partition that overrides an existing source partition for an Enscribe file.
`-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>`	These options specify target partitions and have the same rules as the corresponding options that specify source partitions.
`-pq <sql_predicate>`	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>.
`-w | { -wp <poll_interval> }`	Directs vericom to wait for the job to finish before returning the prompt. Normally, vericom returns immediately after starting a job. ◆ -w waits the default time of one minute. ◆ -wp <poll_interval> waits the specified number of minutes and overrides the default. When either option is used, vericom polls the server until the job finishes, and then the statistics of the run are displayed. If multiple -w or -wp flags are specified, the last one on the command line takes effect.
`-rd0 | -rdN <run_ID>`	Valid for NonStop SQL/MP and Enscribe only. Controls delta processing for a compare pair. ◆ -rd0 disables delta processing for this run. All rows are compared. ◆ -rdN <run_ID> enables delta processing using a previous job run as the basis for the delta. For <run_ID> use the number from the Run ID line at the beginning of the job comparison report. Vericom does not validate the run ID that is supplied. To use these options, you must specify a compare pair with: ◆ -j ◆ -g ◆ -c

Table 8 Vericom runtime arguments (continued)

Argument	Description
`-{url | u} <url>`	Specifies the URL for GoldenGate Veridata Server in the form of <host>:<port>. Required if a run is not executed with veridata/server/bin as the working directory. If -url is not specified, vericom will try to determine the URL from the veridata.loc and veridata.cfg files.
`-{password | p} <password>`	Specifies the password or key of GoldenGate Veridata Server. Required if you do not execute a run with veridata/server/bin as the working directory. ◆ <password> must match the text that is in the veridata/shared/conf/password.txt file. This password is generated by the installer program. If -password is not specified, vericom will try to open password.txt to obtain the password.

Vericom exit statuses

Vericom exits with one of the following statuses.

Status	Description
`0`	The command executed successfully. If a job was run, it finished with all rows in-sync. If -w or -wp was not specified, the exit status is 0 if the job started successfully.
`1`	Invalid vericom syntax was used. For example, the following are invalid: vericom -helptun (Typographical error.) vericom -j -g group1 (The name of the job is missing.)
`2`	Vericom could not find connection information by looking in veridata.loc or veridata.cfg, and connection parameters were not specified with command arguments.
`3`	Provides more granularity for input errors that involve flags that run comparisons. For example, the following mistakes will cause this error: vericom –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 –j job1 –g group1 –rd0 In the preceding example, the -rd0 flag requires –c because delta processing applies at the compare pair level.
`4`	The job ran successfully, but there were rows that had a comparison status of something other than in-sync.

Status	Description
5	There was a communication error with GoldenGate Veridata Server.

Vericom output examples

To view the results of a comparison that is run with vericom, you can use GoldenGate Veridata Web to view the comparison report (see page 13), 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 -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 -j TestJob -w
Connecting to: localhost:9177
Run ID: (2257, 0, 0)
Job Start Time: 2008-03-21 22:48:05
Job Stop Time: 2008-03-21 22:48:20
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 -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'.

CHAPTER 11

GoldenGate Veridata Server configuration parameters

When using server-side sorting, you can use these parameters to adjust different aspects of the Nsort memory configuration. Nsort is the sorting library API that is built into GoldenGate Veridata server.

Overview of the server memory

GoldenGate Veridata Server uses virtual memory in the following ways:

- **Server memory for basic operation**. This is the amount of virtual memory that the server needs simply to operate. It stores object pools, database access libraries, and other information. This is usually about 30 MB.
- **Nsort memory**. This is the memory that is used by Nsort when server-side sorting is used. The virtual memory for Nsort is allocated for the entire comparison, not per thread. The rows are read from the agent and submitted to Nsort to be sorted. The Nsort sorting occurs in a thread that is separate from the thread that reads from the agent, and Nsort may use more threads to work in parallel. Once all of the rows from the agent are submitted to Nsort, the server process retrieves the sorted rows from Nsort for comparison.
- **Row hash queue memory**. This is the memory that buffers data between the agent processes, Nsort, and the server process. A comparison that uses database sorting requires a single queue each for the source and target. A comparison that uses server-side sorting requires two queues each for the source and target: one to buffer data from the agent, and one to buffer data from Nsort. Each queue has a capacity of 40 500k blocks. 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.
- **SOAP messaging memory**. This is the memory that is used to serialize the SOAP messages between the server and the UI.

- **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 Nsort cannot be greater than the minimum of:

- System physical memory
- Available memory in swap
- Current process limit (rlimit)

The Nsort library performs a check for available system virtual memory before allocating memory. The allocation can be affected by other processes that are running on the local system. The memory check is operating-system dependent, and not all checks are performed on all platforms.

When checking, Nsort assumes that it is the only Nsort in the current process. If the server employs multiple Nsort processes, however, this check can be inaccurate. This check also can be inaccurate if the free virtual memory has become fragmented over time. Even if the memory check succeeds, the actual allocation could fail if a contiguous chunk of memory is not available. This causes the comparison to fail. To overcome these uncertainties, use the server configuration parameters.

Estimating memory usage

Here is how GoldenGate Veridata Server uses memory resources on a per-comparison basis and among the overall workload.

Except for fixed virtual memory of 30 MB for basic tasks, GoldenGate Veridata Server uses the remaining virtual memory to perform comparisons.

To estimate the maximum number of concurrent comparisons

{Available virtual memory - 30 MB fixed memory} / {Average memory used per comparison}

Where: Average memory used per comparison assumes the following:

- Half of the comparisons will use server-side sorting (Nsort).
- The producer and consumer queues between the server and agents are fully utilized (500 Kb * 40 per queue). Two queues are needed for database sorting. Four queues are needed for server-side sorting.
- The maximum Nsort memory per comparison is set to the default.
- There is a fixed overhead for threads per comparison equal to 2Kb per thread stack plus two fixed. Database sorting requires four threads. Server-side sorting requires 14 threads.

In general this amount is about 80 MB of virtual memory per comparison.

To estimate the amount of memory used per row

((number of columns in key + 1) * 4) + 14 + size of key columns as ascii

For example, if a key is composed of one, 4-byte integer column, the memory required is:

((1 + 1) * 4) + 14 + 4 = 26 bytes of memory for this row

NOTE This assumes that non-key columns are compared by using a hash, not literally. More space is needed for literal comparisons.

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 GoldenGate Veridata Server installation directory.

- Consult GoldenGate Technical Support before changing these parameters.
- After changing a parameter value, stop and start GoldenGate Veridata Server.

Parameter descriptions

server.max_nsort_memory

Sets the maximum amount of Nsort virtual memory that is available to all running comparisons that use server-side sorting. The default amount assumes that half of the comparisons will use server-side sorting. You can reduce this value to reduce the impact of the server process on your system.

If a comparison cannot get enough virtual memory when it starts, it fails or tries again. If a comparison does get enough virtual memory, the currently available Nsort virtual memory gets decremented by the amount that the comparison reserves. When a comparison completes, it increments the amount of available Nsort virtual memory by the amount of Nsort virtual memory that it had reserved.

Syntax `server.max_nsort_memory {default | <n>{ M | m }}`

- default allows GoldenGate Veridata to define a maximum value that is dependent on the operating system.
- <n>{M|m} specifies a value in megabytes. For example, 1000M means a limit of 1000 megabytes.

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.

Syntax `server.max_concurrent_comparison_threads {default | <n>}`

- default allows GoldenGate Veridata to compute the maximum number of concurrent threads based on average usage and available resources.
- <n> is a positive integer that sets the maximum number of concurrent comparison threads.

server.max_comparison_nsort_memory

Sets the maximum amount of virtual memory that can be reserved by a single comparison that is using Nsort. The default is 50 MB for the source data and 50 MB for the target data. A comparison that tries to reserve more than this amount will fail, even if it is configured to retry.

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_nsort_memory {default | <n>{ M | m }}`

- default specifies that there is no per-comparison limit.
- <n>{M|m} specifies a value in megabytes. For example, 600M sets the limit for any comparison to be 600 megabytes.

server.nsort_waiting_threshold

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

Syntax `server.nsort_waiting_threshold {default | <n>}`

- default allows all jobs to run, regardless of the number of comparisons that are in retry state.
- <n> 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.

server.nsort_compare_function_pool_size

Sets the size of the Nsort compare function pool. When a comparison tries to reserve Nsort virtual memory, it will fail or retry if there are no free Nsort compare functions. If a comparison can reserve the required amount of Nsort virtual memory, it will use an Nsort compare function, thereby reducing the amount of those functions that are available. When the comparison completes, it will return its compare function to the pool.

Syntax `server.nsort_compare_function_pool_size {0 | <n>}`

- 0 allows GoldenGate Veridata to define a default value that is dependent on the operating system.
- <n> is a positive integer that sets the compare function pool to the given size.

Additional parameters in the web interface

You can set these additional parameters within a profile in the GoldenGate Veridata Web interface.

Sorting Retry Limit

Controls the number of times that a comparison is retried if it cannot reserve the amount of virtual memory that it needs. The default of 0 prevents the comparison process from trying to reserve virtual memory if there is not enough available when it begins execution. A value of greater than 0 causes the process to try again to reserve virtual memory if its initial attempt fails.

The repeat attempt is executed after all of the other comparisons in the current job have been executed. If the profile option **Sorting retry interval** is greater than 0, the comparison process waits the specified interval before making a repeat attempt. Valid values are from 0 to 1000.

Sorting Retry Interval

Controls the amount of time that a comparison waits between attempts to reserve virtual memory if the initial attempt fails. Valid values are 0 to 999999999 seconds. A value of 0 means the comparison does not wait between attempts. A value of greater than 0 means the comparison waits for the specified interval before making another attempt. The default value is 600 seconds (10 minutes). Comparisons that fail to reserve virtual memory after the first attempt are retried after all of the other comparisons in that job have been executed.

CHAPTER 12

Upgrading GoldenGate Veridata

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

Upgrading to a new version of the server and web components

Upgrade support overview

- **Upgrades from version 1.x to later versions**: There is no upgrade path from version 1.x to version 2.x or 3.x. You must recreate your configuration objects in the new version. You can install the new version, and then run the old and new versions side-by-side, so that you can refer to the existing configurations as you recreate them in the new version.
- **Upgrades from version 2.x 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. 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.x using the existing repository**: Use the following procedure, "Server and web upgrade instructions for all platforms". When installing a new version of GoldenGate Veridata Server and GoldenGate Veridata Web into an existing directory, the GoldenGate Veridata installer detects the existing installation and prompts you to upgrade or quit. The upgrade path retains the current 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.

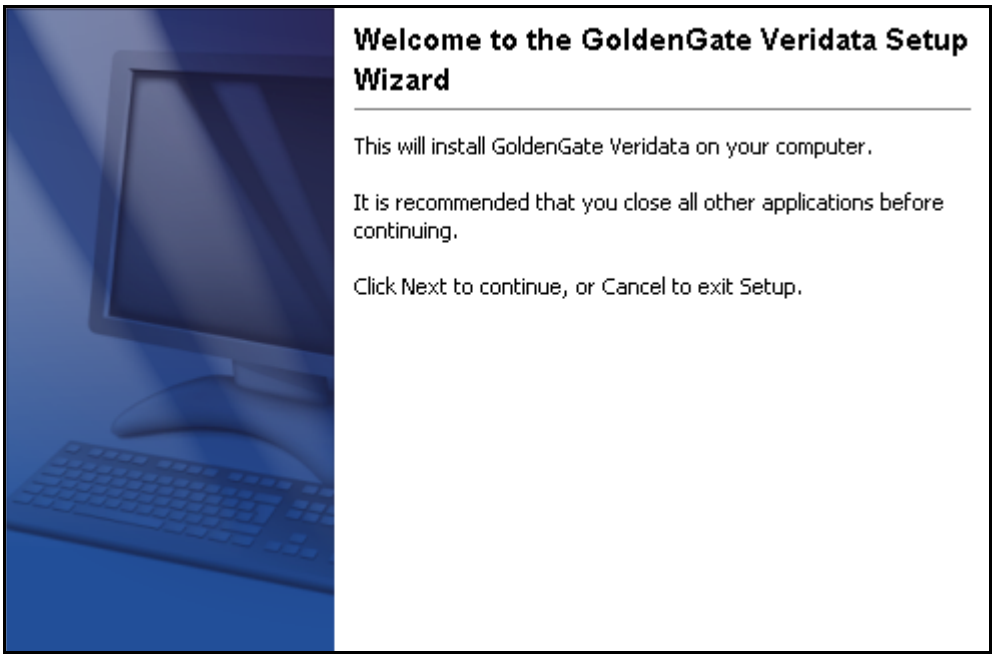
Server and web upgrade instructions for all platforms

1. Start the GoldenGate Veridata repository database.
2. Stop all running comparisons or wait until they are finished.
3. Close all GoldenGate Veridata Web client sessions.
4. Shut down GoldenGate Veridata Server and GoldenGate Veridata Web. (For more information, see page 74.)
5. Run the GoldenGate Veridata installer:

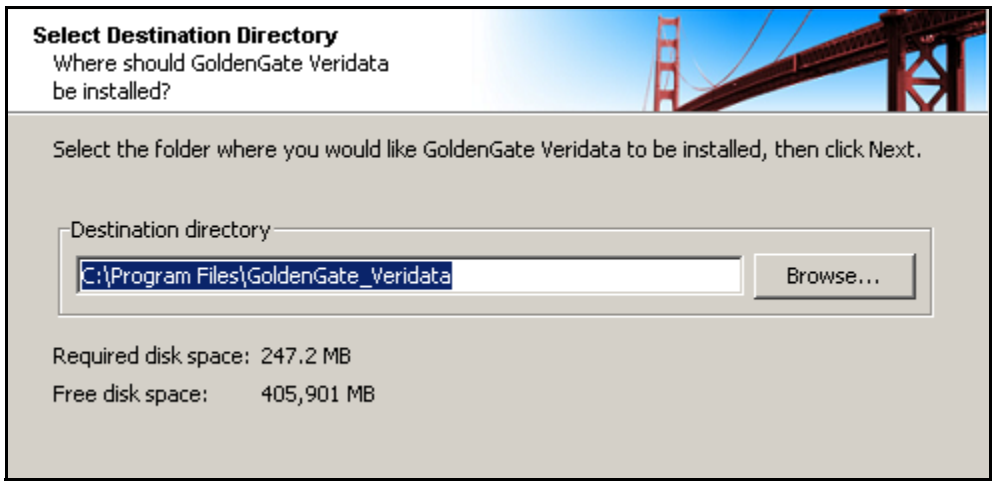
- UNIX and Linux: Run the GoldenGate_Veridata_<platform><version>.sh file from X-windows or an X-windows emulation program.
- Windows: Run the GoldenGate_Veridata_<platform><version>.exe file.

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.

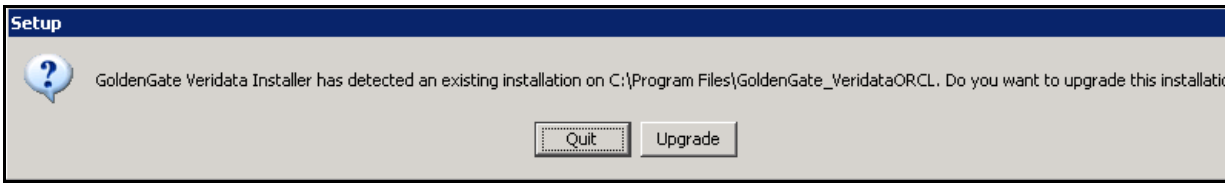
6. Click **Next** to start the installation.



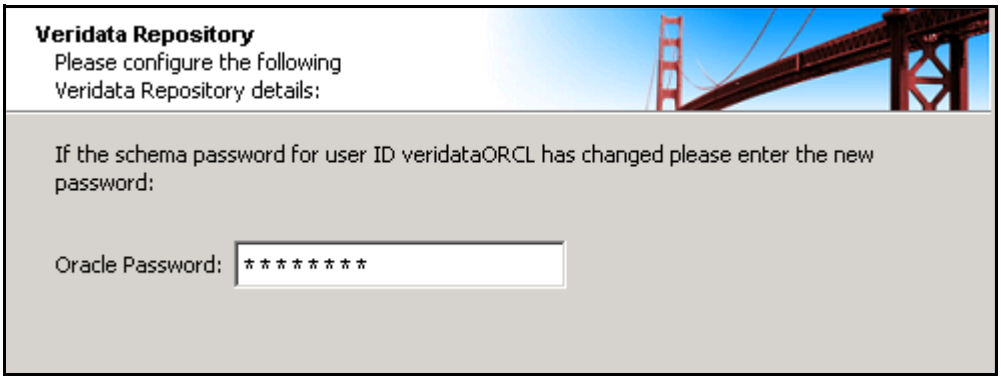
7. Select the GoldenGate Veridata installation that you want to upgrade.



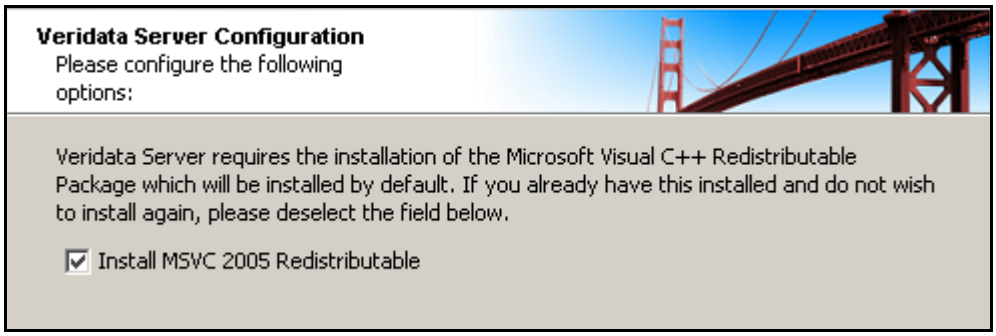
8. Click **Upgrade** to upgrade the existing installation (and preserve the repository). If you do not want to upgrade this installation or preserve the repository, click **Quit**.



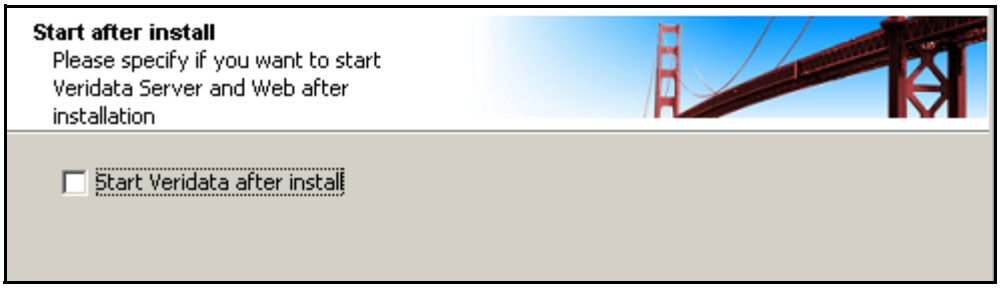
9. The installer displays the encrypted version of the last known password for the repository owner. Either accept the default or type the correct one.



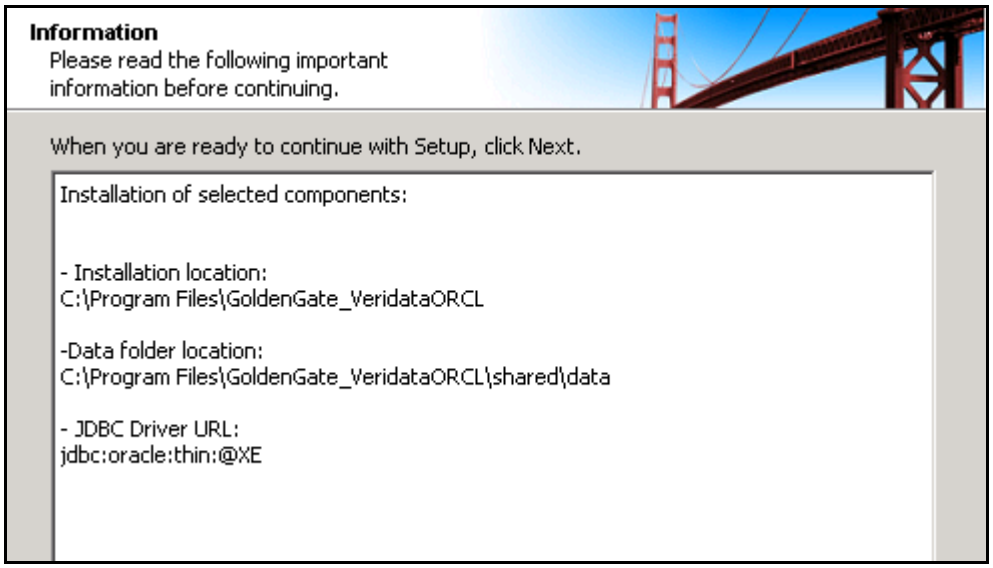
10. Accept the default to install the Microsoft Visual C++ Redistributable Package unless the system contains a version of this package that is at least as current as the one shown by the installer.



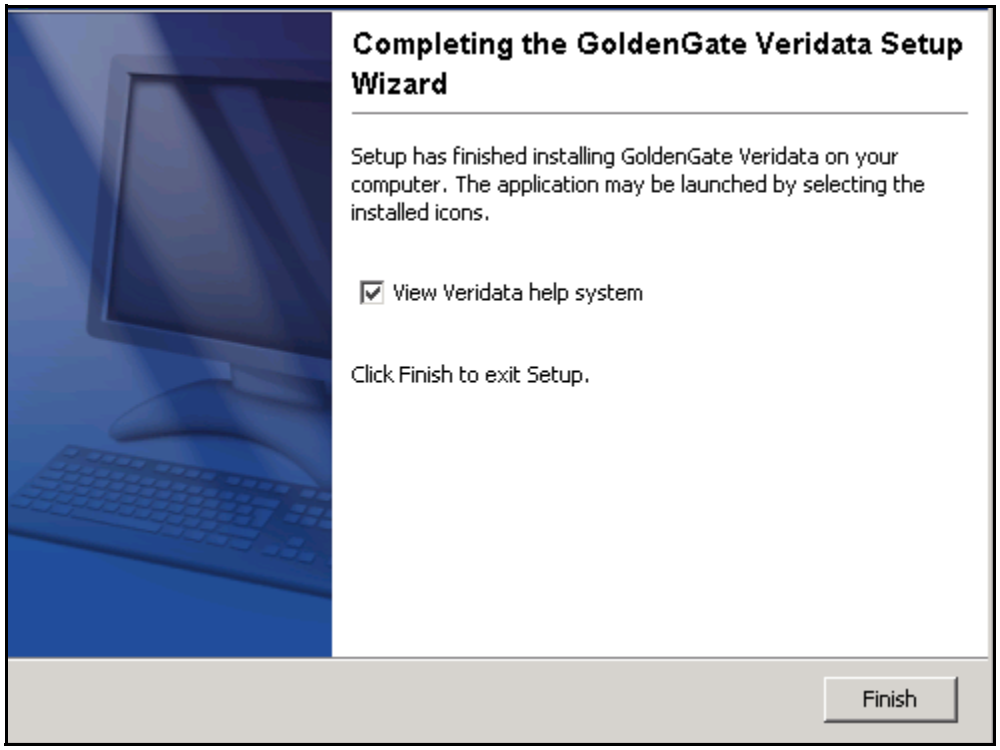
11. Specify whether or not to start the software after the installation is finished. By default, it is installed to start manually. If you are installing it as a service, a system administrator can change the properties so that it starts automatically when the system starts.



12. Review your selections and either click **Cancel** to start over or **Next** to start the upgrade.



13. To view the GoldenGate Veridata help system after the installation, accept the default and click **Finish**.



Upgrading to a new version of the agent

- As of GoldenGate Veridata version 3.0, you can use a Java agent for Oracle, SQL Server, and Teradata.
- To upgrade the agent on NonStop SQL/MP, you must use the version 3.0 C-agent.
- A version 3.0 C-agent is also available for the Oracle database as an upgrade.

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 "Installing GoldenGate Veridata Java Agent" on page 24.
- 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 GoldenGate Veridate Web, so that they can update their connection configurations.

Upgrading a C-agent to a newer version

The instructions provided are for upgrading a C-agent from Veridata 2.x to a newer version. 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.

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. Download the GoldenGate Veridata Agent build file for your platform from the GoldenGate support site.
5. FTP the file to the host system of the agent.
6. Extract the new 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.

   To extract the files, do as follows:

   - Windows: Double-click the file to run your default Windows archiving/compression utility, or use any such utility of your choosing.
   - UNIX and Linux: Unzip and extract the files as follows:

     ```
     gunzip <filename>.tar
     tar -xf <filename>.tar
     ```

7. Run GGSCI from the agent directory.
8. In GGSCI, issue the following command to start the Manager process.

   ```
   START MANAGER
   ```

9. In GGSCI, issue the following command to verify that the installation succeeded. The command should return “Manager is running.”

   ```
   INFO MANAGER
   ```

10. (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. Download the GoldenGate Veridata Agent build file for your platform from the GoldenGate support site.
5. Unzip the files to a temporary directory on your workstation.
6. Transfer the files in binary mode to the volume and subvolume on the NonStop Server where you want to upgrade the agent.
7. Alter the VERUNPAK macro to be an edit file by issuing the following TACL command.

   ```
   FUP ALTER VERUNPAK, CODE 101
   ```

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

   ```
   RUN VERUNPAK
   ```

9. 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.
   ```

10. 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.
    ```

11. Run GGSCI from the agent directory.
12. In GGSCI, issue the following command to start the Manager process.

    ```
    START MANAGER
    ```

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

    ```
    INFO MANAGER
    ```

14. If tables have remote partitions, copy VSNSERV to the remote nodes.

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 GoldenGate Veridata agent subvolume. Otherwise, you might need to download the correct agent build for that hardware type from GoldenGate. It will include the correct VSNSERV.

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

- Install the entire 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 "Creating a GLOBALS file" on page 38.

CHAPTER 13

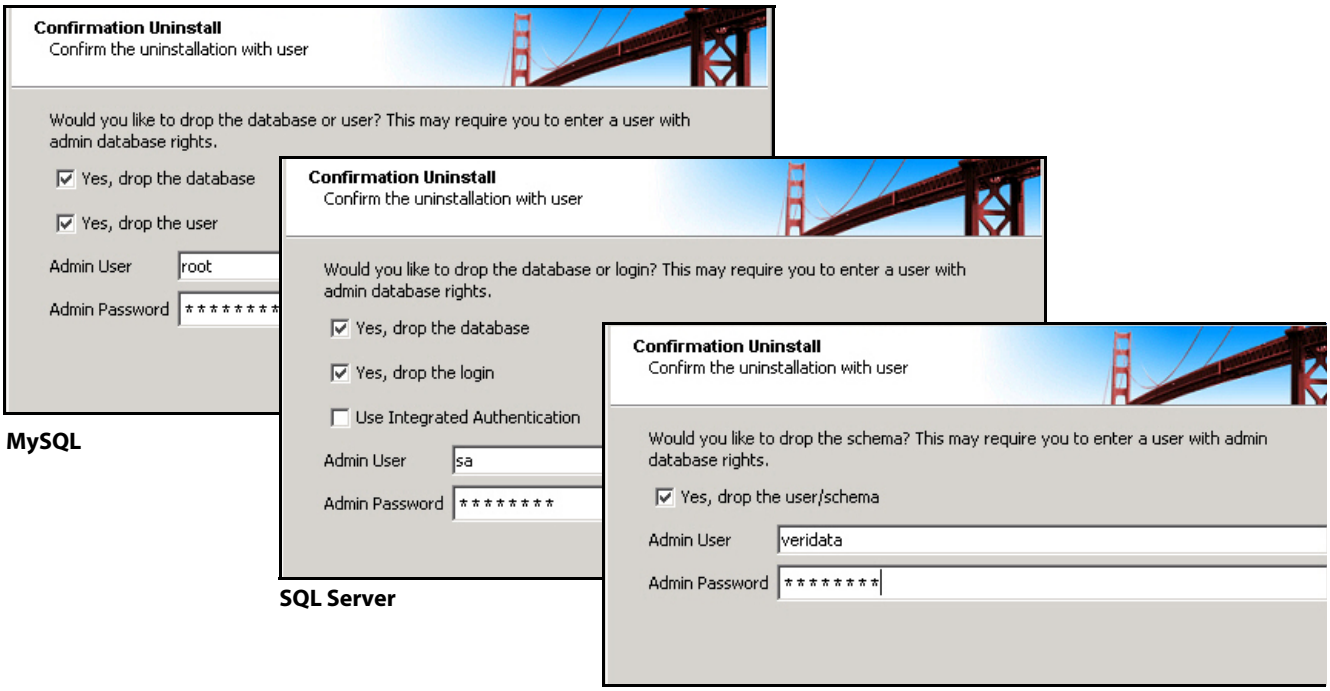
Uninstalling GoldenGate Veridata

Uninstalling the server and web components (all platforms)

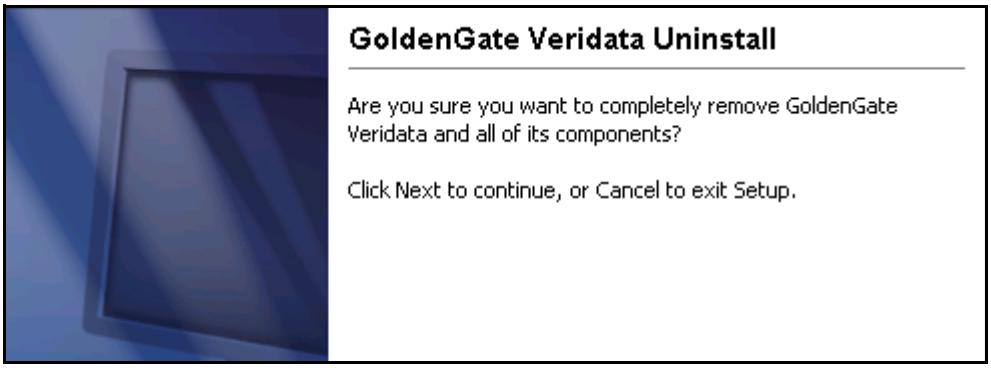
1. Log on to the system as a user who has privileges to uninstall files.
2. Stop all running comparisons, or wait until they are finished.
3. Stop all GoldenGate Veridata Web client sessions.
4. Stop GoldenGate Veridata Server and GoldenGate Veridata Web. See page 74.
5. If you intend to drop the GoldenGate Veridata database repository, start the repository database.
6. Run the uninstall program from the GoldenGate directory:
   - *UNIX and Linux*: Using an X-windows or an X-windows emulation program, run uninstall.sh from the GoldenGate installation directory.
   - *Windows*: From the **Start** menu, go to the GoldenGate Veridata program folder, and select **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.

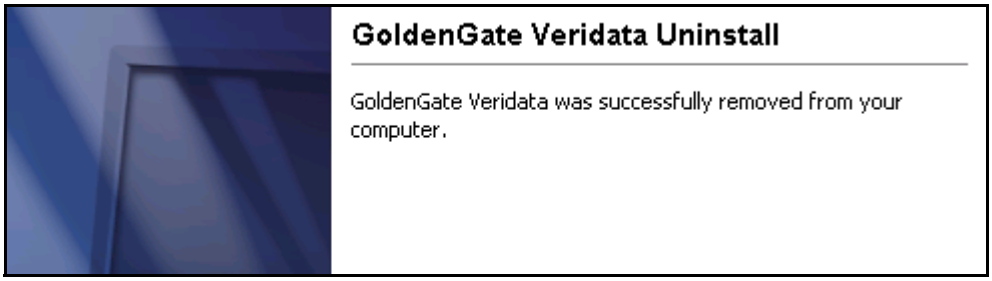
7. By default, the GoldenGate repository is not removed, so that user data is preserved. To drop the GoldenGate repository, select:
   - (Oracle) **Yes, drop the user/schema**. Supply the login of the database user or schema that owns the repository.
   - (Other databases) **Yes, drop the database** 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 datasource configuration accepts it.



8. Click **Next** to confirm and start the uninstallation.



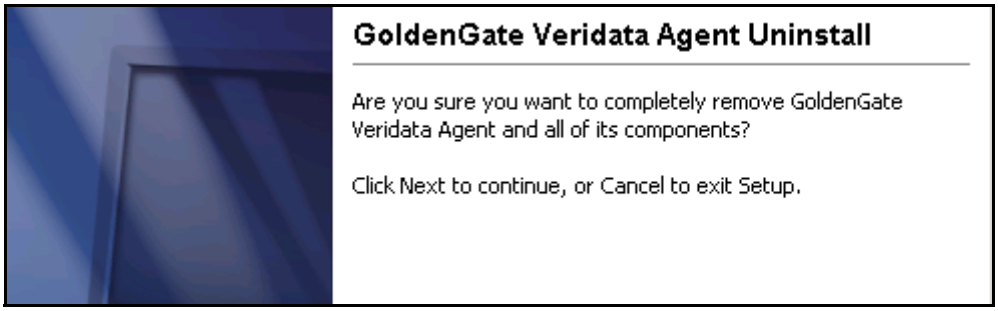
9. Click **Finish** to exit the uninstaller program.



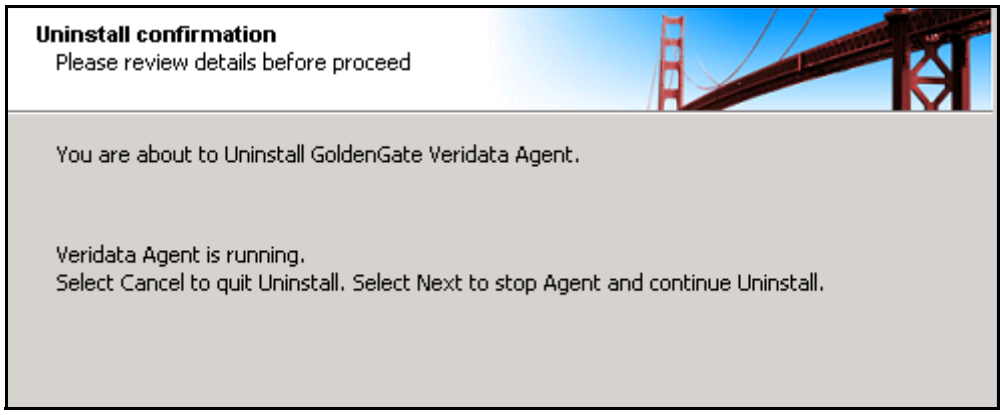
10. (Optional) Remove the GoldenGate Veridata installation directory. The uninstall program does not remove files in this directory that were created by other programs. Logs are an example of such files.

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 page 74).
4. From the **Start** menu, go to the GoldenGate Veridata Agent program folder, and then select **GoldenGate Veridata Agent Uninstaller**. Alternatively, you can run the uninstall_veridata_agent.exe executable file from the installation directory.
5. Click **Next** on the welcome screen to start the removal process.



6. One or both of the messages shown below are displayed, depending on whether the agent is still running. A running agent could indicate that server processes still are using this agent. Click **Next** to continue with the uninstallation process or click **Quit** to cancel it. If you continue with the uninstallation, the status is displayed.



7. Click **Finish** to exit the uninstaller program.



8. (Optional) Remove the GoldenGate VeridataAgent installation directory. The uninstall program does not remove files in this directory that were created by other programs. Logs are an example of such files.

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 page 74).
4. Remove the files in the agent installation directory.

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 GoldenGate_Veridata/agent.
4. 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.

Optional:

5. Remove the files that are in the installation location.
6. (NonStop) Purge any copies of the file named $*.*.VSNSERV.

APPENDIX 1

Changing GoldenGate Veridata Web port numbers

After the GoldenGate Veridata server and web components are installed, you can change certain port numbers if they conflict with other programs.

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

<GoldenGate_Veridata_install_directory>/GoldenGate Veridata/server/web/conf/server.xml

- The first line in the file is `<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:

  ```
  <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 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.

APPENDIX 2

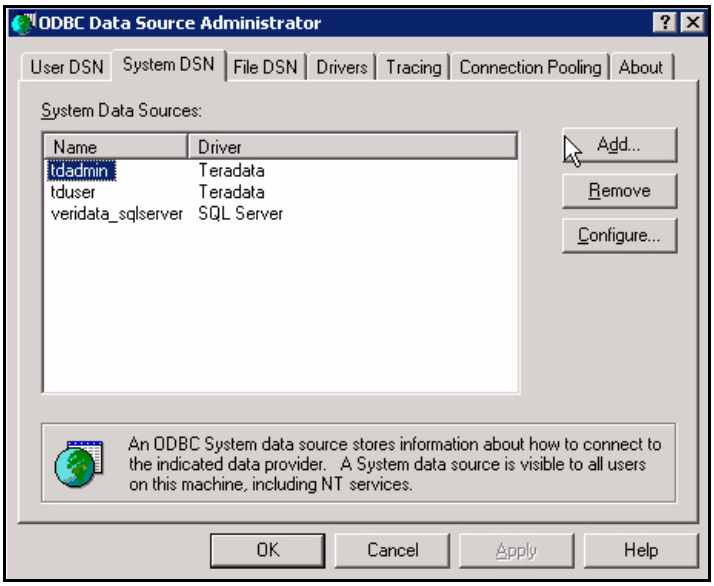
Configuring an ODBC connection for a SQL Server repository

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.

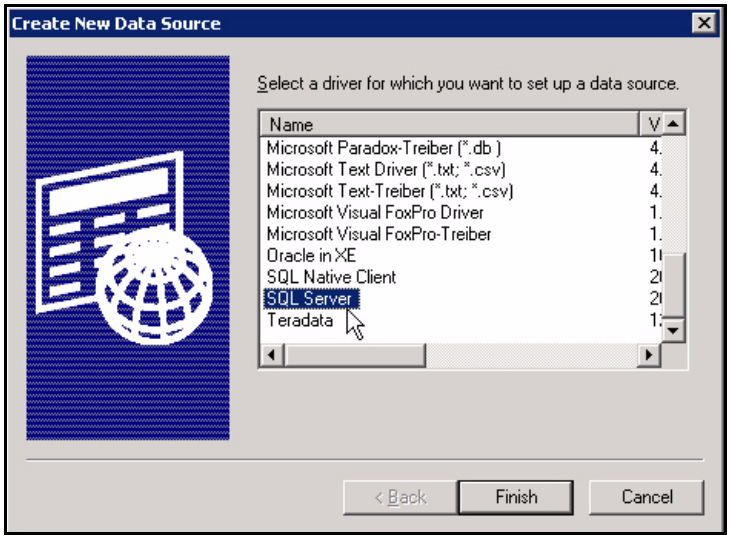
NOTE You can use these same steps to edit or verify an existing DSN.

1. Run one of the following ODBC clients:
   - If using a 32-bit version of 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 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 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**.

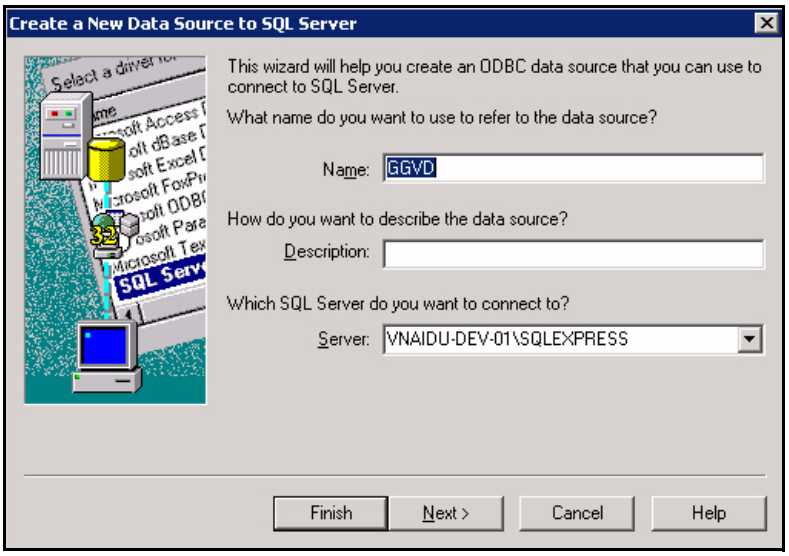


4. (New data source only) Under **Create New Data Source**, select either the **SQL Server** or **SQL Native Client** driver.

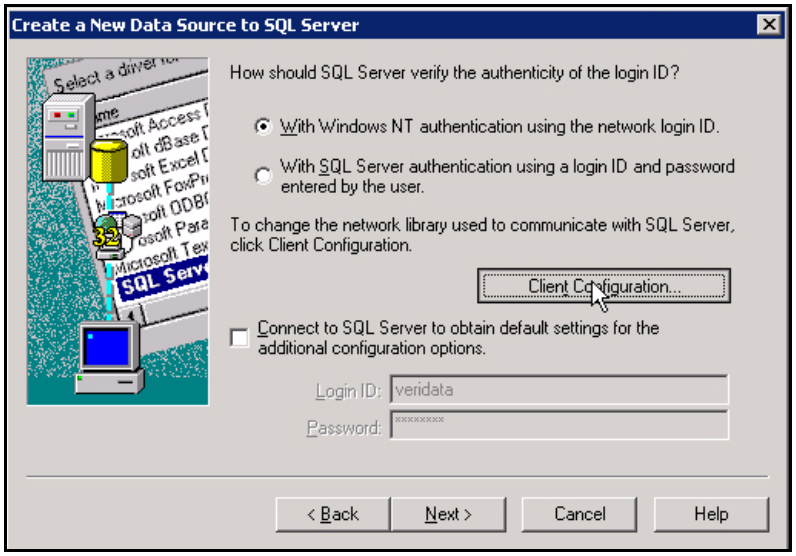


5. Click **Finish**. A new wizard is displayed.

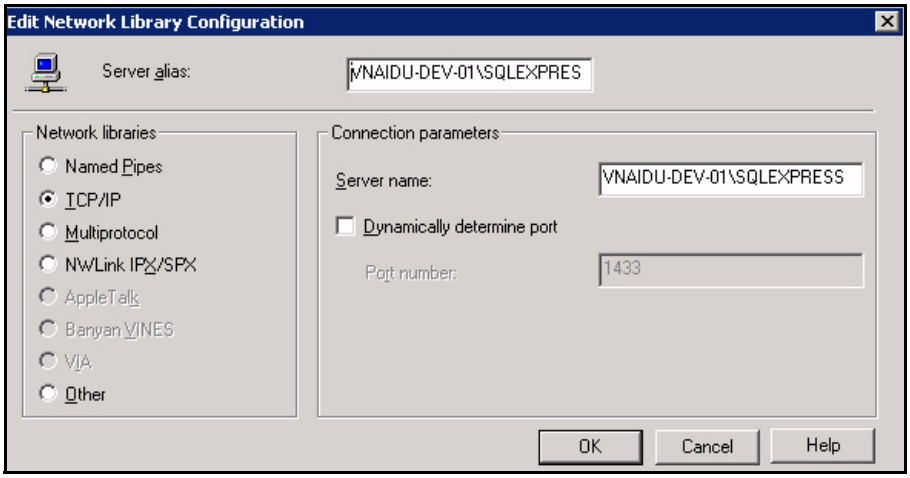
6. Supply or verify the following:
   - **Name**: Can be of your choosing.
   - **Description**: (Optional) Type a description of this data source.
   - **Which SQL Server do you want to connect to**: Select (or type) the name of the SQL Server instance. To create the DSN for a named instance, specify 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.

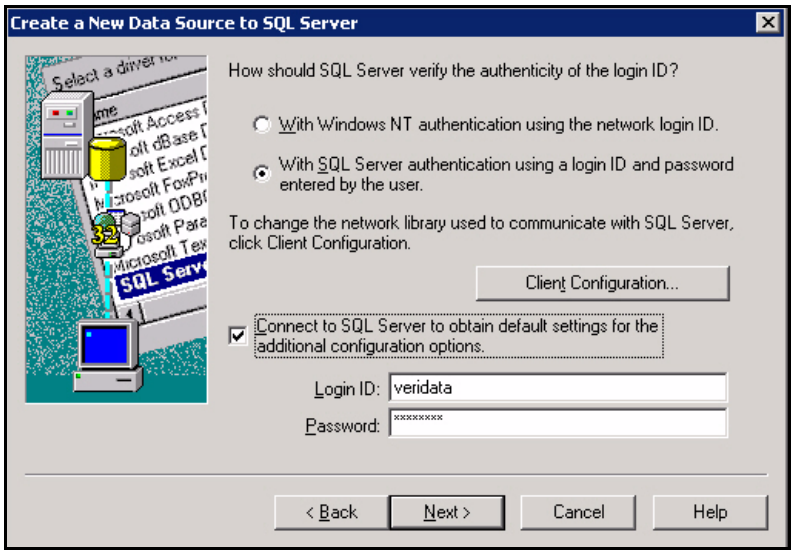


9. In the **Edit Network Library Configuration** popup:
   - Enter a server alias name (optional).
   - Deselect the **Dynamically determine port** box, and then enter the name of a port number to use (the default of 1433 is acceptable). GoldenGate Veridata requires the use of a static port.
   - Under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



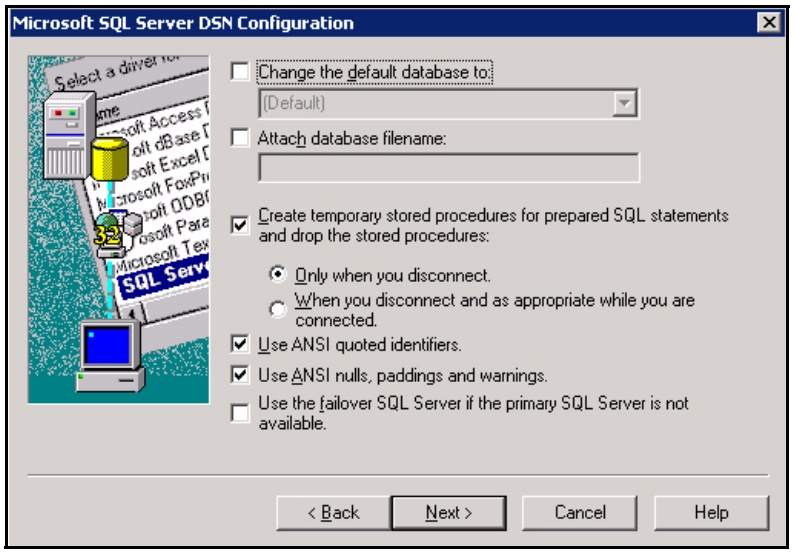
10. Click **OK** .

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 GoldenGate Veridata.



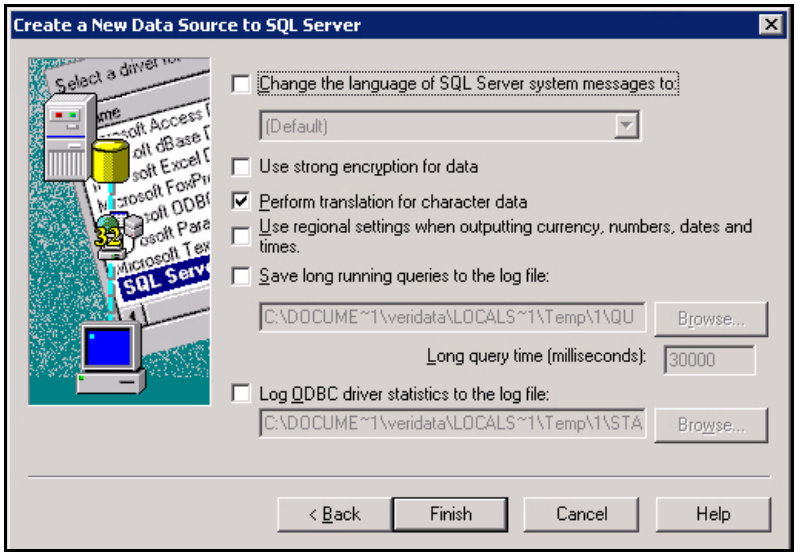
12. Click **Next**.

13. You can leave all of these settings to their default values.



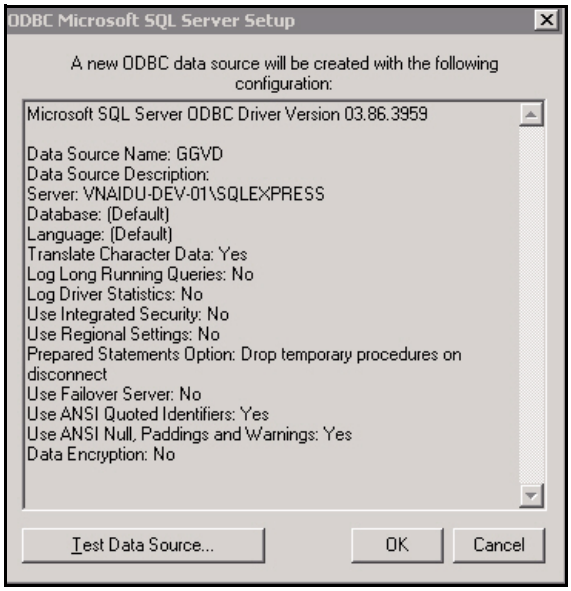
14. Click **Next**.

15. You can leave all of these 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.

APPENDIX 3

Glossary of GoldenGate Veridata Terms

The following explains terminology contained in this manual.

Term	Definition
agent	See GoldenGate Veridata Agent.
client	An interface to GoldenGate Veridata Server, from which you configure or execute comparison work. GoldenGate Veridata provides a GUI client named 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	A GoldenGate Veridata Web configuration object that establishes the correct structural relationship between the source and target objects in a compare pair.
compare pair	A configuration object in 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 GoldenGate Veridata transforms the native data type of a column. The use of standardized formats enables GoldenGate to support heteogeneous 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.

Term	Definition
connection	A configuration object in GoldenGate Veridata Web that stores information about how the 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 datasource type.
datasource	A set of stored data, such as a relational or hierarchical database.
delta processing	A performance feature that GoldenGate Veridata supports on the NonStop platform by which 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 GoldenGate replication software, which also is used as the command interface to the Manager process of the GoldenGate Veridata Agent.
GLOBALS file	A parameter file that is required on some platforms where GoldenGate Veridata Agent is installed. It stores parameters that may be required in some installation conditions.
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 GoldenGate Veridata Server and is controlled by the GoldenGate Veridata Manager process.
GoldenGate Veridata Server	The comparison engine that coordinates 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 was purchased.
group, or compare group	A configuration object in GoldenGate Veridata Web that is a logical container for one or more compare pairs (sets of related source and target tables) and their associated datasource 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.

Term	Definition
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 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 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 GoldenGate Veridata Agent component. It coordinates requests sent to the agent process by the 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.
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 GoldenGate Veridata Web or the vericom interface, which enables comparisons to be performed against a subset of the rows in a table or file.

Term	Definition
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 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 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 GoldenGate Veridata Server.
server-side sorting	The sorting of source and target rows for comparison when performed by 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 GoldenGate Veridata to test source and target table structures for compatibility.
vericom	The commandline interface to GoldenGate Veridata.

Index

A

ADDEVENTS Windows service option 35
ADDSERVICE Windows service option 35
administration tool, using 71
Administrator role 71
administrator, specifying user as 44, 53, 63
agent component
- about 10
- installing 24, 32
- starting 73
- stopping 73
- system requirements 15
- uninstalling 103
- upgrading 96

AUTOSTART Windows service option 35

B

browser
- connecting to server from 75
- supported kinds 22

C

category.dll 35
client
- about 10
- connecting to 75
- system requirements 22

clustered keys 23
command line, running comparisons from 77
compare pairs, about 11
comparison
- configuring in web interface 11
- how performed 11
- results of, viewing 13
- running from command line 77

comparison report 13
confirmation step
- about 12
- delaying 80
- skipping 80

connections, as configuration object 11
COOS step
- about 12
- delaying 80
- skipping 80

D

data
- how compared 11
- see also *rows*

data source name (DSN), for SQL Server repository 106
database
- as repository 10
- specifying for repository 45, 54, 63
- verifying 41, 50, 60

delta processing
- about 12
- enabling 82

DetailReportViewer role 71
directory
- installation
  - *agent* 24, 27, 32, 33, 37
  - *server* 27, 43, 52, 61
- user data location 43, 52, 61

disk space
agent component 17
server component 19
DYNAMICPORTLIST parameter 33, 36, 39

E

Enscribe
agent permissions 18
agent supported 17
agent, installing 36

F

Firefox 23

G

GGSMGR default Manager name 34
ggsmsg.dll 35
GGSPREFIX parameter 38
GLOBALS file 34, 38
GoldenGate Veridata
about 9
installing 24, 41, 50
new features 7
system requirements 15
uninstalling 100
upgrading 91
GoldenGate Veridata Agent
about 10
installing 24, 32, 33, 36
starting 73
stopping 73
uninstalling 103
upgrading 96
GoldenGate Veridata repository
about 10
system requirements 20
GoldenGate Veridata Server
about 10
configuration parameters 86
installing 41, 50, 60
starting 73
stopping 79
uninstalling 100
upgrading 91
GoldenGate Veridata Web
about 10
connecting to 75
installing 60
starting 73
uninstalling 100
upgrading 91
groups, about 11
Guardian, installing agent on 17

H

hash, rows 11
HOST parameter 38
HTTP port 44, 53, 62

I

in-flight, definition 12
initial comparison step 11
installing
agent component 24, 32, 33, 36
server component 41, 50, 60
web component 41, 50, 60
in-sync, definition 12
Internet Explorer 22
introduction to Veridata 9
IPC buffer memory 87

J

Java Runtime Environment 22
JDBC driver, Teradata 29
jobs
about 11
running 77

K

keys, clustered 23

L

latency, replication 13
Linux
installing agent 24, 32
installing server and web components 41, 50, 60

M

Manager
about 10
as Windows service 35
configuring on NonStop 39
multiple on same system 34, 38
name, customizing 34, 38
port number 33
starting 73
stopping 73
upgrading 96
MANUALSTART Windows service option 35
maybe out-of-sync queue (MOOSQ) 12
memory
managing for server component 86
server requirements 19
MGRSERVNAME parameter 34
Microsoft Visual C++ Redistributable Package 47, 57, 67, 94
MOOS queue
about 12
memory use 86
multi-tier server installation 20
MySQL, using as repository 41

N

new features in this release 7
NonStop SQL/MP
installing agent 36
using delta processing on 12
Nsort
about 19
memory configuration 86

O

ODBC connection, for SQL Server repository 106
ODBCAD32.EXE client 106
OOSXML file
about 13
based on OOS from previous run 79
controlling size of 80
Oracle
agent, installing 26, 32, 33
ORACLE_HOME variable 54
Oracle, using as repository 50
out-of-sync (OOS) report
about 13
format of 80

P

parameters, server 86
partition, specifying for rows 81, 82
partitions on remote NonStop nodes 37
PASSWORD Windows service option 35
persistently out of sync, definition 12
port
default 23
HTTP 44, 53, 62
listening, server component 46, 56, 65
Manager 33
MySQL database 45
numbers, changing 105
Oracle database 55
shutdown 44, 53, 62
SQL Server database 64
static, SQL Server 22, 109
web component 76
PowerUser role 71
profile
about 11
overriding 79

R

RAM requirements
agent component 17
server component 19

removing GoldenGate Veridata 100
replication latency 13
report, comparison 13
ReportViewer role 71
repository
about 10
system requirements 20
where to install 20
roles, security 70
row hash queue memory 86
row hash step, see *initial comparison step*
rows
comparing 77
comparing only previously out-of-sync 79
how compared 11
number in OOSXML chunk 80
specifying partition 81, 82

S

security, configuring 70
server component
about 10
configuration parameters 86
installing 41, 50, 60
starting 73
stopping 79
system requirements 18
uninstalling 100
upgrading 91
server.max_comparison_nsort_memory 89
server.max_nsort_memory 88
server.nsort_compare_function_pool_size 89
server.nsort_waiting_threshold 89
service, installing as on Windows 30, 33, 46, 56, 65
shutdown port 44, 53, 62
SOAP messaging memory 86
Sorting Retry Interval setting 90
Sorting Retry Limit setting 90
source key range, specifying as partition 81
SQL predicate, specifying 81, 82
SQL Server
agent, installing 26
using as repository 60
SQL/MP clustered keys 23
starting
GoldenGate Veridata Agent 73
GoldenGate Veridata Server 73
GoldenGate Veridata Web 73
Manager 36, 40
statistics for comparisons 13
stopping
GoldenGate Veridata Agent 73
GoldenGate Veridata Server 73
GoldenGate Veridata Web 73
Manager 73
subset, of rows 81, 82
system requirements 15

T

tables, comparing 77
tablespaces, specifying 56
Teradata agent, installing 26
threads, number to use 79
Tomcat Web Server
Administration tool 71
configuration file 105
tracing, setting 80

U

uninstalling GoldenGate Veridata 100
UNIX
installing agent 24, 32
installing server and web components 41, 50, 60
upgrading GoldenGate Veridata 91
USER Windows service option 35
users and groups 70

V

vericom tool 77
Visual C++ 2005 Redistributable package 18
VSNSERV module 37

W

web component
connecting to 75
installing 41, 50, 60
system requirements 22
uninstalling 100
upgrading 91

Windows
installing agent 33
installing server and web components 41, 50, 60