# Oracle® GoldenGate Veridata 23c (23.4.0.0.0)





Oracle GoldenGate Veridata 23c (23.4.0.0.0),

F85008-04

Copyright © 2015, 2025, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

### Preface

Document	ation Accessibility	
Related In	formation	
Convention	ns	
Overvie	W	
1.1 Wha	t is Oracle GoldenGate Veridata?	<u>:</u>
1.2 Con	cepts	<u>-</u>
1.3 Orac	cle GoldenGate Veridata Architecture	2
1.4 How	Oracle GoldenGate Veridata Works?	;
1.4.1	Initial Step or Row Hash Step in Maybe Out of Sync	;
1.4.2	Confirm Out of Sync	į
1.5 Orac	cle GoldenGate Veridata Supported Use Cases	-
Prepare	<del></del>	
2.1 Sup	ported Databases for Compare	<u> </u>
2.1 Supp 2.2 Supp	ported Databases for Compare ported Databases for Repair	;
2.2 Supp 2.3 Supp	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL	
2.1 Supple 2.2 Supple 2.3 Supple 2.4 Supple	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes	
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes Supported Datatypes - Oracle	
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server	
2.1 Supple 2.2 Supple 2.3 Supple 2.4.1 2.4.2 2.4.3	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL	
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB	
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5	corted Databases for Compare corted Databases for Repair corted Databases for Repair SQL corted Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB Supported Datatypes - DB2 LUW	
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5 2.4.6	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB Supported Datatypes - DB2 LUW Supported Datatypes - DB2 for i	; ; ;
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5	corted Databases for Compare corted Databases for Repair corted Databases for Repair SQL corted Datatypes  Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB Supported Datatypes - DB2 LUW Supported Datatypes - DB2 for i Supported Datatypes - DB2 z/OS	
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5 2.4.6 2.4.7	ported Databases for Compare ported Databases for Repair ported Databases for Repair SQL ported Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB Supported Datatypes - DB2 LUW Supported Datatypes - DB2 for i	1:
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5 2.4.6 2.4.7 2.4.8	corted Databases for Repair corted Databases for Repair SQL corted Databases for Repair SQL corted Datatypes  Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB Supported Datatypes - DB2 LUW Supported Datatypes - DB2 for i Supported Datatypes - DB2 z/OS Supported Datatypes - Sybase Adaptive Server Enterprise	10 12 13 14 15
2.1 Supp 2.2 Supp 2.3 Supp 2.4 Supp 2.4.1 2.4.2 2.4.3 2.4.4 2.4.5 2.4.6 2.4.7 2.4.8 2.4.9	corted Databases for Repair corted Databases for Repair SQL corted Datatypes Supported Datatypes - Oracle Supported Datatypes - Microsoft SQL Server Supported Datatypes - PostgreSQL Supported Datatypes - MySQL, MariaDB Supported Datatypes - DB2 LUW Supported Datatypes - DB2 for i Supported Datatypes - DB2 z/OS Supported Datatypes - Sybase Adaptive Server Enterprise Supported Datatypes - Teradata Vantage	1: 1:

	2.5.1 Disk and Memory Requirements for the Agent Component	15
	2.5.2 Database Privileges for the Agent Component	15
	2.6 Oracle GoldenGate Veridata Server System Requirements	17
	2.6.1 Location for the Server Component	17
	2.6.2 Disk and Memory Requirements for the Server Component	17
	2.7 Oracle GoldenGate Veridata Distribution	19
	2.7.1 Downloading Oracle GoldenGate Veridata	19
	2.7.2 Downloading Oracle GoldenGate Veridata C-Agent and Java Agent	19
3	Install	
	3.1 Installing and Running Oracle GoldenGate Veridata	1
	3.2 Logging	7
	3.2.1 Logs and Setting Log Levels	7
	3.2.2 Agent	8
	3.2.2.1 Agent Logs Location	8
	3.2.2.2 Updating Agent Log Settings	9
	3.2.2.3 Enabling Performance Logs	9
	3.2.2.4 Enabling Trace Logs	9
	3.2.2.5 Enabling Query Logs	10
	3.2.2.6 Modifying Log Size	10
	3.2.3 Server	11
	3.2.3.1 Server Log Location	11
	3.2.3.2 Updating Server Log Levels	11
	3.2.3.3 Changing Log Size	12
	3.3 Running the Configuration Assistant	12
	3.4 Installing and Configuring Source and Target Agents	21
	3.4.1 About Oracle GoldenGate Veridata Agent Deployment Scripts	21
	3.4.2 Configuring Oracle GoldenGate Veridata Agent	22
	3.4.3 Starting the Oracle GoldenGate Veridata Agent	23
	3.4.4 Using the Oracle GoldenGate Veridata Agent Deployment Script in Debug Mode	23
	3.4.5 Reloading the Logging Properties of the Veridata Agent	23
	3.5 Silent Installation	24
	3.6 Uninstalling Oracle GoldenGate Veridata	26
	3.7 Installing Oracle GoldenGate Veridata C-Agent	26
	3.7.1 Installation Overview	26
	3.7.2 Installing the C-Agent on a NonStop System	27
	3.7.2.1 Installing the Oracle GoldenGate Veridata Agent Files	27
	3.7.2.2 Copying VSNSERV to Remote Nodes	28
	3.7.2.3 Creating a GLOBALS File	28
	3.7.2.4 Configuring Manager	29
	3.8 Best Practices	30

3.8.1.1	ridata Components Proximity	30
	How to Check Connection between 2 Nodes	31
3.8.2 Vei	ridata Server and Agent Memory	32
3.8.2.1	Server Memory	32
3.8.2.2	Agent Memory	33
3.8.2.3	Memory for Sorting and Sort Directory	33
3.8.3 SS	L Certificate Validation	33
3.8.4 Da	tabase Permissions	34
3.8.5 Da	ta for Comparison	35
3.8.5.1	Specifying Primary Key (PK) Columns	35
3.8.5.2	Partitioning	35
3.8.5.3	Exclude Columns	36
3.8.5.4	Delta Comparison	36
3.8.6 Ver	ridata Configurations	38
3.8.6.1	Connection Configurations	38
3.8.6.2	Profile Configurations	38
3.8.6.3	Server Configurations	40
3.8.6.4	Agent Configurations	41
3.8.7 Fre	equently Asked Questions (FAQ)	42
	g Oracle GoldenGate Veridata Web User Interface	1
4.2 Home Pa	ige	1
4.2 Home Pa 4.2.1 Abo		1 2
	out	
4.2.1 Abo 4.2.2 He	out	2
4.2.1 Abo 4.2.2 He	out Ip	2
4.2.1 Abo 4.2.2 He	out Ip	2
4.2.1 Abo 4.2.2 He 4.2.3 Uso Manage 5.1 Connecti	out Ip er Menu ons	2 2 2
4.2.1 Above 4.2.2 He 4.2.3 Use Manage 5.1 Connection 5.1.1 Cree	out  Ip er Menu  ons eating a Connection	2 2 2 1 1
4.2.1 Above 4.2.2 He 4.2.3 Use Manage 5.1 Connecti 5.1.1 Cre 5.1.2 Edit	out  Ip er Menu  ons eating a Connection  iting a Connection	2 2 2 1 1 2
4.2.1 Above 4.2.2 He 4.2.3 Use Manage 5.1 Connecti 5.1.1 Cres 5.1.2 Edi 5.1.3 De	out  Ip er Menu  ons eating a Connection iting a Connection leting a Connection	2 2 2 1 1
4.2.1 Above 4.2.2 He 4.2.3 Use Manage 5.1 Connecti 5.1.1 Cre 5.1.2 Edi 5.1.3 De 5.2 Groups a	out lp er Menu  ons eating a Connection iting a Connection leting a Connection and Compare Pairs	2 2 2 1 1 2
4.2.1 Above 4.2.2 He 4.2.3 Use Manage 5.1 Connecti 5.1.1 Cre 5.1.2 Edi 5.1.3 De 5.2 Groups a	out  Ip er Menu  ons eating a Connection iting a Connection leting a Connection	2 2 2 1 1 1 2 3
4.2.1 Above 4.2.2 He 4.2.3 Use Manage 5.1 Connecti 5.1.1 Cre 5.1.2 Edi 5.1.3 De 5.2 Groups a	out  Ip er Menu  ons eating a Connection iting a Connection leting a Connection and Compare Pairs oup Details	2 2 2 1 1 2 3 4
4.2.1 Above 4.2.2 He 4.2.3 Uso 5.1.1 Cres 5.1.2 Edit 5.1.2 Edit 5.1.3 De 5.2.1 Groups at 5.2.1 Groups at 5.2.1 Groups at 5.2.1.1 Groups at 5.2.1.2	out lp er Menu  ons eating a Connection iting a Connection leting a Connection and Compare Pairs oup Details Creating Groups and Compare Pairs Cloning a Group	2 2 2 2 1 1 1 2 3 4 4 4 4 8
4.2.1 Above 4.2.2 He 4.2.3 Use Manage  5.1 Connecti 5.1.1 Cres 5.1.2 Edi 5.1.3 De 5.2 Groups a 5.2.1 Groups a 5	out lp er Menu  ons eating a Connection iting a Connection leting a Connection and Compare Pairs oup Details Creating Groups and Compare Pairs Cloning a Group	2 2 2 2 1 1 2 3 4 4 4
4.2.1 Above 4.2.2 He 4.2.3 Use Manage  5.1 Connection 5.1.1 Crest 5.1.2 Edit 5.1.3 De 5.2 Groups at 5.2.1 Groups at 5.2.1.1 5.2.1.2 5.2.1.3	out lp er Menu  ons eating a Connection iting a Connection leting a Connection and Compare Pairs oup Details Creating Groups and Compare Pairs Cloning a Group	2 2 2 2 1 1 1 2 3 4 4 4 4 8
4.2.1 Above 4.2.2 He 4.2.3 Use Manage  5.1 Connection 5.1.1 Crest 5.1.2 Edit 5.1.3 De 5.2 Groups at 5.2.1 Groups at 5.2.1.1 5.2.1.2 5.2.1.3	out lp er Menu  ons eating a Connection iting a Connection leting a Connection and Compare Pairs oup Details Creating Groups and Compare Pairs Cloning a Group Deleting a Group mpare Pairs	2 2 2 2 1 1 2 3 4 4 4 4 8 8

	5.3.1 Editing Profile Settings	17
	5.3.2 Using the Default Profile	23
	5.4 Jobs	23
	5.4.1 Overview	24
	5.4.2 Creating a Job	24
	5.4.3 Running Jobs	25
	5.4.3.1 Overriding Manual Row Partitions	26
	5.4.4 Editing a Job	26
	5.4.5 Monitoring Jobs	26
	5.4.5.1 Viewing Completed Jobs	26
	5.4.5.2 Viewing Jobs that are Running	29
	5.4.5.3 Viewing Details of the Repair Jobs	30
	5.4.6 Estimating Comparison Time	32
	5.4.7 Using the Comparison Report	32
	5.4.8 Repairing Out-Of-Sync Jobs	33
	5.4.8.1 Downloading SQL Statements for Out-of-Sync Records	34
	5.5 Users	35
	5.5.1 Configuring User Groups	36
	5.5.2 Configuring Users	37
	5.6 Utilities	38
	5.6.1 Export	38
	5.6.2 Import	38
6	Administer	
	6.1 Starting and Stopping the Veridata Server and the Repository	1
	6.2 Managing Veridata Agent	1
	6.2.1 Starting and Stopping the C-Agent	2
	6.2.2 Managing Java-Based Components	2
	6.2.2.1 Starting and Stopping the Java-Based Components	2
	6.2.2.2 Managing Bequeath Agent	3
	6.2.3 Reloading Logging Information	3
	6.2.4 Connecting to the Oracle GoldenGate Veridata Web Interface	3
	6.3 Vericom	4
	6.3.1 Overview of the Vericom Tool	4
	6.3.2 Supported Commands in Vericom	4
	••	
	6.3.3 Vericom Output Examples	6
	6.3.3 Vericom Output Examples 6.4 Veridata Export and Import Utilities	6 7
	6.3.3 Vericom Output Examples 6.4 Veridata Export and Import Utilities 6.4.1 Introduction to the Export and Import Utilities	6 7 7
	6.3.3 Vericom Output Examples 6.4 Veridata Export and Import Utilities 6.4.1 Introduction to the Export and Import Utilities 6.4.1.1 Supported Configurations	6 7 7 7
	6.3.3 Vericom Output Examples 6.4 Veridata Export and Import Utilities 6.4.1 Introduction to the Export and Import Utilities	6 7 7

	6.4	.2.2	Using the Import Utility	8
	6.4	.2.3	Processing the Configuration	9
	6.4	.2.4	SSL Configuration for Export and Import Utilities	11
	6.4.3	Confi	guration File Element Reference	11
	6.4	.3.1	configuration	12
	6.4	.3.2	column	14
	6.4	.3.3	colfilter	14
	6.4	.3.4	colfiltercol	15
	6.4	.3.5	compare-pair	15
	6.4	.3.6	connection	19
	6.4	.3.7	conn-properties	20
	6.4	.3.8	delta-config	20
	6.4	.3.9	description	21
	6.4	.3.10	enscribe-info	21
	6.4	.3.11	enscribe-key	22
	6.4	.3.12	excluded-column	23
	6.4	.3.13	expandddl	23
	6.4	.3.14	filter	24
	6.4	.3.15	group	25
	6.4	.3.16	job	26
	6.4	.3.17	profile	27
	6.4	.3.18	key-column	28
	6.4	.3.19	profile-general	29
	6.4	.3.20	sorting-method	29
	6.4	.3.21	initial-compare	29
	6.4	.3.22	confirm-out-of-sync	29
	6.4	.3.23	param	30
	6.4	.3.24	repair	30
	6.4	.3.25	sql-partition	30
	6.4	.3.26	table partition	31
6.5	Serve	er Para	ameters	31
	6.5.1	Over	view of the Server Memory	32
	6.5.2	Estim	ating Memory Usage	32
	6.5.3	How	to Set a Parameter	33
	6.5.4	Parar	meter Descriptions	33
6.6	Agen	t Para	meters - General	36
	6.6.1	comp	are.xmldatatype.format	37
	6.6.2	coos.	batch.fetch	37
	6.6.3	datab	pase.characterSet	37
	6.6.4	datab	ase.transaction.isolation	37
	6.6.5	pool.r	maxIdleTime	38
	6.6.6	pool.	checkInterval	38

	6.6.7	pool.maxIdle	38
	6.6.8	pool.maxSize	38
	6.6.9	pool.maxStatements	38
	6.6.10	server.port	38
	6.6.11	rowscn	39
	6.6.12	zlib.buffer.flush.size	39
	6.7 Ager	nt Parameters - Connections	39
	6.7.1	IBM Db2 for i	40
	6.7.2	IBM Db2 for LUW	40
	6.7.3	IBM Db2 for z/OS	40
	6.7.4	Apache Hive	40
	6.7.5	MariaDB	41
	6.7.6	Microsoft SQL Server	41
	6.7.7	MySQL	41
	6.7.8	Oracle Database	41
	6.7.9	PostgreSQL	42
	6.7.10	Sybase Adaptive Server Enterprise	42
	6.7.11	Teradata Vantage	42
7	Configu	re	
	7.1 High	Availability	1
	7.1.1	Setting Up the High Availability for Oracle GoldenGate Veridata Server 23c	2
	7.1.2	Configuring a Shared File Path	2
	7.1	L.2.1 Reports	2
	7.1	L.2.2 Wallet	2
	7.1.3	Setting up High Availability for Oracle GoldenGate Veridata Agent	3
8	Secure		
	8.1 Conf	iguring Workflow for Two-Way SSL in Oracle GoldenGate Veridata 23c	1
	8.1.1	Enabling SSL in the Agent Properties File	1
	8.1.2	Generating Agent Keystore and Certificate	2
	8.1.3	Generating Server Keystore and Certificate	2
	8.1.4	Importing Agent Certificate to Server Truststore	2
	8.1.5	Saving Server Keystore/Truststore Passwords to Server Wallet	3
	8.1.6	Importing Server Certificate to Agent Truststore	3
	8.1.7	Saving Agent Keystore/Truststore Passwords to Agent Wallet	3
	8.1.8	Creating an Agent Connection in UI	4
	8.2 Conf	iguring Two-Way SSL for the NSK C-Agent on the Veridata Server	5
		figuring Oracle GoldenGate Veridata Agent Using Kerberos to Connect to Oracle	
	Data	base	7

8.4	Configuring Oracle Golden-Gate Veridata Agent Using Kerberos to Connect to Hive	8
8.5	Connecting Oracle GoldenGate Veridata to SSL-Enabled Oracle Database	9
8.6	Connecting Oracle GoldenGate Veridata to SSL-Enabled MySQL Database	9
8.7	Connecting Oracle GoldenGate Veridata to SSL-Enabled SQL Server Database	11
8.8	Connecting Oracle GoldenGate Veridata to SSL-Enabled PostgreSQL Database	11
Mię	grate	
9.1	Migrating Veridata Configuration from Veridata 12.2.1.4.0 to Oracle GoldenGate Veridata 23c	1
Pe	rformance	
10.1	Improving the Performance of Oracle GoldenGate Veridata	1
Ac	cessibility	
11.1	About this Accessibility Article	1
	11.1.1 Accessibility Overview	1
	11.1.2 What is Web Accessibility?	1
	11.1.3 Why Accessibility is Important?	2
	11.1.4 About Building for Accessibility	2
11.2	Using Oracle GoldenGate Veridata with Assistive Technologies and Keyboard	3
	11.2.1 Component-Specific Keyboard Controls	5
Tro	oubleshooting	
12.1	Configuration Assistant	1
12.2	2 MySQL Operation	1
12.3	Comparison and Repair Performance	2
	12.3.1 Slowed Comparison	2
	12.3.1.1 Sorting Phase	2
	12.3.1.2 Initial Compare Phase	4
	12.3.1.3 COOS Phase	4
	12.3.1.4 Check Stalled Queries in Database	5
	12.3.2 Slowed Repair	6



# **Preface**

This article explains the installation and configuration of Oracle GoldenGate Veridata, the Veridata Web User Interface and all its command line utilities.

- Audience
- Documentation Accessibility
- Related Information
- Conventions

### **Audience**

This article is intended for system administrators and database users to learn about Oracle GoldenGate Veridata. It is assumed that readers are familiar with web technologies and have a general understanding of Windows and UNIX platforms.

# **Documentation Accessibility**

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

#### **Accessible Access to Oracle Support**

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

### **Related Information**

- Oracle GoldenGate Documentation
- Oracle GoldenGate Veridata

### Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.



### Overview

- What is Oracle GoldenGate Veridata?
- Concepts
- Oracle GoldenGate Veridata Architecture
- How Oracle GoldenGate Veridata Works?
- Oracle GoldenGate Veridata Supported Use Cases

### 1.1 What is Oracle GoldenGate Veridata?

Oracle GoldenGate Veridata is a high-speed, data-comparison and repair solution that identifies, reports on, and fixes data discrepancies between heterogeneous databases without interrupting ongoing business processes automatically. It provides an easy-to-use yet powerful solution for identifying out-of-sync data before it negatively impacts the business.

With its advanced data comparison and validation capabilities, Oracle GoldenGate Veridata empowers businesses to confidently make critical decisions based on reliable, up-to-date information. Whether it's mission-critical systems, data migrations, or disaster recovery, Veridata ensures data integrity, reduces downtime, and enhances overall productivity.

# 1.2 Concepts

Get familiar with the following concepts and other commonly used terms before you get started with Oracle GoldenGate Veridata.

To begin using Oracle GoldenGate Veridata, you need to create some objects that identify the data that you want to compare and which help you to manage your work. Create these objects in the following order:

- Connections: Define a Connection to the source and target databases that contain the data you want to compare. The Oracle GoldenGate Veridata server uses the Connection information to communicate with the Oracle GoldenGate Veridata agent.
- **2. Groups**: Groups are logical containers for one or more Compare Pairs. A Group is associated with a set of connections to the source and target data.
- 3. **Compare Pairs**: A Compare Pair is the logical relationship between a source and target table for the purpose of comparing their data. Compare Pairs are linked to Groups.
- 4. Jobs and Repairs: To run Compare Pairs, you must create a Job. The Job configuration determines which compare groups are processed. In the UI, you can repair jobs and also can generate SQL files for a few supported databases for out-of-sync jobs.
- 5. Users, User Groups, and Roles: You can manage users and user groups from the User Management page. You can create, delete, edit, users and user group. Users and user groups are assigned with various roles, which are assigned with various privileges. The following are the various roles assigned in Oracle GoldenGate Veridata: Administrator, Super User, Monitoring Operator, Detail Monitoring Operator, Repair Operator, Job Operator, and Command Line Operator.

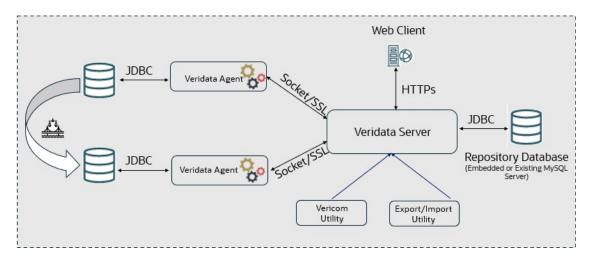


- 6. Delta Comparison: In Oracle GoldenGate Veridata, the source and target tables are configured using compare pairs, which are grouped and added to a job to run the comparison. During the subsequent runs of a comparison job, the comparison of the tables can be performed based on what has changed in the tables from the previous job run; these jobs are Delta Processing Jobs.
- 7. Profile: A profile is a set of global processing parameters, each containing unique settings for a specific purpose. Parameters or properties in a profile define how a comparison job or repair job is processed. Oracle GoldenGate Veridata provides a default profile, but you can create your own profiles. You can create as many profiles as needed and associate them with any job or compare pair (to override the job profile). You can override profile assignments at run time.

### 1.3 Oracle GoldenGate Veridata Architecture

The Oracle GoldenGate Veridata architecture comprises the following:

Figure 1-1 Oracle GoldenGate Veridata Architecture



#### Veridata Agent

It is a light-weight Java program that runs either on DB systems or remotely. The Veridata agent interacts with the source or target DBs to fetch and return blocks of data rows, the column-level detail, and run queries for repairs.

#### Veridata Server

Compares data, confirms out-of-sync data, repairs data and produces reports. It also coordinates the jobs across all other components.

#### Repository Database

Oracle GoldenGate Veridata 23c supports only MySQL Server database to store the metadata.

#### The architecture also consists of three main utilities.

#### Vericom Utility

It is a command line interface. Veridata Command Line Interface. All the administration tasks performed in GUI can be done via this utility.

#### Export and Import Utilities

Used to Export Configuration files of existing Oracle GoldenGate Veridata and import it to the Veridata 23c. The utilities are mainly used in the migration process of Oracele



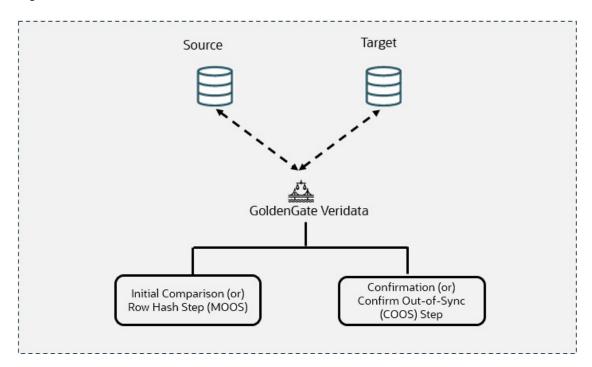
GoldenGate Veridata Servers. The Import Utility also helps in creating Compare Pairs, Groups, and Jobs using the GoldenGate Replicat Process parameter file.

### 1.4 How Oracle GoldenGate Veridata Works?

To compare data accurately while transactional and replication operations are taking place, Oracle GoldenGate Veridata uses a two-step process to maintain data accuracy:

- 1. Initial step or Row hash step MOOS Maybe Out-of-Sync
- 2. COOS Confirm Out-of-Sync

Figure 1-2 How Veridata Works?



- Initial Step or Row Hash Step in Maybe Out of Sync
- Confirm Out of Sync

### 1.4.1 Initial Step or Row Hash Step in Maybe Out of Sync

The steps involved in Maybe Out of Sync (MOOS) are:

- Rows retrieved from source/target databases
- Data converted into standard format
- · Hashes data and sent to Server
- Out-of-sync rows stored in MOOS queue

This marks the initial stage of the comparison process.

After the comparison is initiated, the Oracle GoldenGate Veridata agent retrieves rows from both source and target tables using a specified query.



Oracle GoldenGate Veridata, being a heterogeneous data comparison and repair tool, ensures compatibility by standardizing data types if the source and target databases differ.

Data retrieval involves fetching primary key (PK) column values directly (actual values are fetched), while non-key columns are hashed to minimize network transfer for comparison. This unique hashing process streamlines the comparison process by providing a dependable and efficient means of determining similarities or differences between rows in the source and target databases.

Additionally, it is possible to alter the default hashing option. However, this adjustment may lead to decreased performance and increased network usage, particularly with a higher number of columns. After completing the initial comparison, if Oracle GoldenGate Veridata detects discrepancies in some rows, then it refrains from displaying the comparison results to the user. Instead, it stores these rows in the Maybe Out of Sync (MOOS) queue in memory. This precaution is necessary due to concurrent real-time replication, which may have some replication latency, potentially leading to the out-of-sync rows being in transit. Subsequently, replication will synchronize them again.

Example: Comparing data of the Employee table in both the source and target databases.

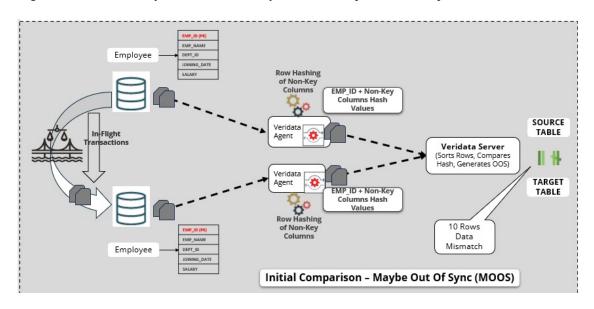


Figure 1-3 Initial step or Row hash step MOOS - Maybe Out-Of-Sync

The Employee table contains a primary key column EMP\_ID.

- Oracle GoldenGate Veridata retrieves the actual values of EMP\_ID and hashes the values
  of other non-key columns.
- The replication process is in progress. If there are any disparities, for example, if there are 10 rows being out of sync, then Oracle GoldenGate Veridata will not immediately confirm them as out of sync.
- 3. Given the real-time replication setup and concurrent data replication, it is possible that these 10 out-of-sync rows have yet to be replicated to the target. This might be due to latency.
- 4. The out-of-sync rows are stored in the MOOS queue in memory.



### 1.4.2 Confirm Out of Sync

The steps involved in Confirm Out of Sync (COOS) are:

- Ensures accurate results by comparing each row
- It happens in parallel with Row hash with configured latency threshold
- Confirm out of Sync requests can be batched
- Good performance for remote databases
- Controlled by a profile setting
- Executing the look ups in parallel

The verification process, known as confirmation or Confirm Out of Sync (COOS), guarantees precise outcomes by validating the status of rows in a dynamic setting. This process entails making conditional inquiries on either the source or target using the rows obtained from the MOOS queue, with the status being assessed as one of the subsequent possibilities:

- In Flight: The row was initially not synchronized during the comparison process, but it has now been corrected. In this scenario, it is presumed that replication or a similar mechanism implemented the modification; however, Oracle GoldenGate Veridata could not verify the synchronization of the rows.
- **In Sync**: The values from the source row were transferred to the target row either through replication or a similar process. Even if the status shows as in-sync, it does not ensure that the rows are synchronized at any given moment, especially if the underlying tables are constantly changing. However, it does signify that replication is functioning properly.
- Persistently out of sync: After the initial comparison step occurred, the row remains unchanged, indicating it is likely out of sync.
  By default, confirmation processing takes place concurrently with the initial comparison step, but the confirmation of each row is delayed until after a specified replication latency threshold has passed. For example, if the latency is set to 60 seconds and an out-of-sync row is identified during the initial comparison step at 10:00, the confirmation step for that row is postponed until 10:01 to ensure replication can apply any pending changes. After the latency is considered, rows can be definitively marked as Out-of-Sync and cataloged in one or more out-of-sync reports.

# Example: Comparing data of the Employee table in both the source and target databases.

After surpassing the latency threshold, the process moves to the second stage known as the COOS step. Here, only the rows that are out of sync are examined, without utilizing hashing. All values from both primary key and non-key columns are retrieved in their original form (actual values). The out-of-sync rows are compared value by value. If the same data inconsistencies persist after this stage, these rows are confirmed as out of sync. You are then presented with details of the out-of-sync rows for further action.



Employee

DEFT\_JD

JOHNNG\_DATE

SALAN

No Hashing is
Done

All Columns Literals

Veridata Server
(Sorts Rows, Compares Hash, Generates OOS)

Agent

Veridata Server
(Sorts Rows, Compares Hash, Generates OOS)

Agent

Agent

Agent

Agent

Confirm Out of Sync (COOS)

Figure 1-4 Confirm Out of Sync (COOS)

#### (i) Note

The latency threshold is crucial in this context. In the previously mentioned scenario, it was established at 60 seconds, but it is adjustable manually. The **Delay Confirm-Out-Of-Sync** parameter within the Profile Configuration governs this threshold. Users are better acquainted with their replication environment and the daily latency they encounter. Hence, they can customize this parameter according to their requirements.

#### Delay Confirm-Out-Of-Sync By (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.

#### (i) Note

You have the option to bypass the two-step-comparison process altogether, specifically skipping the second step known as COOS.

The parameter **Perform Confirm Out-Of-Sync Step** in the Profile Configuration allows users to decide whether to execute or bypass the second step when comparing data in Oracle GoldenGate Veridata.

#### Perform Confirm-Out-of-Sync Step

Controls whether or not the confirmation step is performed. By default, it is performed. Clear the check box under Value to skip the confirmation step and only perform the initial (row hash) comparison. You might skip the confirmation step if, for example, activity on the source tables is quiesced or if replication is not continuously updating the target table(s).



# 1.5 Oracle GoldenGate Veridata Supported Use Cases

In today's complex, hybrid (Cloud and On-premises) IT environment with the same data stored in multiple locations or while migrating the data from one system or application to another, some data discrepancies are almost inevitable. If not discovered and addressed, bad data can lead to poor decision-making; failed service level agreements; and, ultimately, operational, financial, and legal risk.

Furthermore, for a comprehensive explanation, the following are the diverse use cases supported in Oracle GoldenGate Veridata:

Data Migration
Data Migration
Big Data Lake

Data Migration
Cloud

Data Migration
Cloud

Data Migration
Cloud

Auditing & Compliance
Data compliance standards
(HIPAA, PCI-DSS, GDPR)

Data Consistency
Across systems, data centers, onpremises to cloud, relational to data lake

Data Migration
Cloud

Application Upgrades
DB & Application upgrades

Figure 1-5 Oracle GoldenGate Veridata Use Cases

- Data Migration: It is the process of transferring data from one location or one system to another. Data migration is a critical task in various scenarios, including:
  - Database Upgrades: When organizations upgrade their software or hardware systems, they often need to migrate data from the old system to the new one. For example, when transitioning to a new database management system (DBMS) or a newer version of an application.
  - Cloud migration: Many businesses are moving their data and applications to cloud platforms. Data migration in this context involves moving data from on-premises servers to cloud-based storage or from one cloud provider to another.
  - Data center relocation: When a company moves its data center to a new physical location, data migration is essential to ensure the continuity of operations.
     There are various factors involved in the Data Migration out of which Data Validation post migration is an important task. After the migration is complete, it is essential to verify that the data in the new system functions correctly and retains its integrity.
- Auditing & Compliance: As compliance standards evolve, corporations and their leaders
  are increasingly required to uphold exceptionally high levels of responsibility. Organizations
  must ensure their alignment with auditing and compliance standards and furnish data to
  regulatory authorities, such as Health Insurance Portability and Accountability Act (HIPAA),
  Payment Card Industry Data Security Standard (PCI-DSS), General Data Protection
  Regulation (GDPR), and others.



- Data Consistency: In any replication scenario, it is essential to ensure data consistency
  across all systems, whether it involves replicating data between data centers, migrating
  data from on-premises to the cloud, or transferring data from a relational database to a
  data lake. It is imperative that no data discrepancies occur.
- Application Upgrades: When conducting application upgrades, numerous Data Definition Language (DDL) and Data Manipulation Language (DML) operations are executed on the database tables. If this database has replication configured, it is imperative to ensure that all changes made in the source systems are seamlessly and accurately replicated to the target systems.

# Prepare

- Supported Databases for Compare
- Supported Databases for Repair
- Supported Databases for Repair SQL
- Supported Datatypes
- Oracle GoldenGate Veridata Agent System Requirements
- Oracle GoldenGate Veridata Server System Requirements
- Oracle GoldenGate Veridata Distribution

# 2.1 Supported Databases for Compare

Oracle GoldenGate Veridata supports the following databases for comparisons:

- Apache Hive
- HPE NonStop
- IBM Db2
- MariaDB Server
- Microsoft SQL Server
- MySQL
- Oracle Database
- PostgreSQL
- Snowflake
- Sybase Adaptive Server Enterprise
- Teradata Vantage

For the latest information about Oracle GoldenGate Veridata release, including the list of certified database versions and operating systems, go to My Oracle Support at <a href="http://support.oracle.com">http://support.oracle.com</a>.

# 2.2 Supported Databases for Repair

Oracle GoldenGate Veridata supports the following databases for repair functionality:

- HPE NonStop
- IBM Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle Database



- PostgreSQL
- Snowflake
- Sybase Adaptive Server Enterprise
- Teradata Vantage

For the latest information about Oracle GoldenGate Veridata release, including the list of certified database versions and operating systems, go to My Oracle Support at http://support.oracle.com.

# 2.3 Supported Databases for Repair SQL

Oracle GoldenGate Veridata supports the following target databases for Repair SQL functionality:

- Oracle Database
- Microsoft SQL Server
- Snowflake

# 2.4 Supported Datatypes

- Supported Datatypes Oracle
- Supported Datatypes Microsoft SQL Server
- Supported Datatypes PostgreSQL
- Supported Datatypes MySQL, MariaDB
- Supported Datatypes DB2 LUW
- Supported Datatypes DB2 for i
- Supported Datatypes DB2 z/OS
- Supported Datatypes Sybase Adaptive Server Enterprise
- Supported Datatypes Teradata Vantage
- Supported Datatypes Snowflake
- Generic Limitations and Clarifications

### 2.4.1 Supported Datatypes - Oracle

The Oracle GoldenGate Veridata supports the listed datatypes for the Oracle database:

- BOOLEAN
- CHAR
- NCHAR
- VARCHAR2
- VARCHAR
- NVARCHAR2
- NUMBER
- BLOB



- CLOB
- NCLOB
- LONG
- RAW
- LONG RAW
- ROWID
- BINARY\_FLOAT
- BINARY DOUBLE
- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- TIMESTAMP WITH LOCAL TIME ZONE
- UROWID
- XMLTYPE
- INTERVAL YEAR [(year precision)] TO MONTH
- INTERVAL DAY [(day\_precision)] TO SECOND [(fractional\_seconds\_precision)]
- FLOAT
- UDTs
- VECTOR

#### For Oracle Boolean Data Type

- Supported mapping MySQL Data Type using veridata number compare format: Boolean and Number.
- Supported mapping MySQL Data Type using veridata string compare format with limitation: Char and Varchar.
- Limitation: Only uppercase values TRUE and FALSE are treated as in-sync.
- Not supported MySQL Data Type: Binary and Varbinary

#### **Limitations and Clarifications:**

Both Object types and Collection types UDTs are supported

### 2.4.2 Supported Datatypes - Microsoft SQL Server

The Oracle GoldenGate Veridata supports the listed datatypes for the SQL Server database:

- BIGINT
- BIT
- INT
- SMALLINT
- TINYINT
- DECIMAL
- MONEY



- SMALLMONEY
- NUMERIC
- FLOAT
- REAL
- CHAR
- NCHAR
- VARCHAR
- NVARCHAR
- TEXT
- NTEXT
- BINARY
- VARBINARY
- IMAGE
- DATE
- DATETIME
- DATETIME2
- SMALLDATETIME
- DATTIMEOFFSET
- TIME
- GEOGRAPHY
- GEOMETRY
- UNIQUEIDENTIFIER
- XML

For more information about conversion of SQL Server data types to Oracle data types, see <a href="Data Type Conversion">Data Database Gateway for SQL Server User's Guide</a>.

### 2.4.3 Supported Datatypes - PostgreSQL

The Oracle GoldenGate Veridata supports the listed datatypes for PostgreSQL:

- Bit(n)
- Bit Varying(n)
- Boolean
- Char
- citext
- Varchar(n)
- Time with/without timezone
- Date
- Interval
- Bigint



- Serial
- Smallserial
- Bigserial
- Numeric
- Decimal
- Money
- Real
- Double precision
- cidr
- inet
- macaddr
- macaddr8
- uuid
- text
- bytea (binary)
- xml
- smallint
- integer
- json
- jsonb

#### Non-Supported PostgreSQL Data Types

- arrays
- box
- circle
- composite types
- line
- Iseq
- object identifiers
- OID
- pg\_lsn
- pseduo types
- Point
- path
- polygon
- range types
- tsvector
- tsquery



- enum
- domain

#### For Postgres BIT Varying Data Type

- Supported mapping Oracle/MySQL Data Type using veridata string compare format: Char and Varchar
- Not supported Oracle/MySQL Data Type: Binary, Varbinary, and Number

#### **Limitations of Support**

- Bit(n)/Bit Varying(n):
  - Heterogeneous: As the source side, Bit(n)/Bit Varying(n) can only be mapped with a character type of the non-PostgreSQL target database. This is because, the leading "0" of the source data gets truncated in the target db during repair, for example, the target is a number type.
  - Homogeneous: None.
- Network datatype (inet,cidr,uuid):
  - Heterogeneous: All string datatypes like char/nchar/varchar in source side and network datatypes like inet/cidr/uuid in PostgreSQL target side can only considered as hash column.
  - Homogeneous: None.
- MAC/MAC8 datatype:
  - Heterogeneous: All string datatypes like char/nchar/varchar in source side and datatypes like mac/mac8 in PostgreSQL target side, while repair operation, it is successful but it always save in the one formatxx:xx:xx:xx:xx:xx even though the allowed format is multiple, that is xxx:xxx:xxx:xxx or xx-xx-xx-xx. Therefore, it always shows the OOS after repair whenever there is a different format from source side.
  - Homogeneous: None.
- JSONB datatype:
  - Heterogeneous: All string datatypes like char/nchar/varchar in source side and jsonb in PostgreSQL target side, after repair the order of json key/values are not stored in original order. Therefore, it will be OOS for next compare pair run.
  - Homogeneous: None.
- Timestamp with timezone:
  - Heterogeneous: PostgreSQL is not storing the timezone value in the database. Whenever a timestamp with timezone value is inserted, PostgreSQL converts the timestamp into UTC and inserts into it. While retrieving, the actual timezone is not known. Comparing it with the DBs like oracle, sqlserver which stores complete timestamp along with timezone result in OOS all the time.
- Time/Time with Timezone:
  - Homogeneous: PostgreSQL- PostgreSQL Time/Time with Timezone datatype comparison and repair has inconsistencies when the time column is primary key. Some of the rows may not be picked for compare and repair.
- Real:
  - Heterogeneous: Oracle GoldenGate for Veridata cannot support the compare pair of FLOAT (Oracle) to REAL (PostgreSQL). Some data is always OOS after repair due to



the internal representation of the data type in db. Substitute pairs are binary float (Oracle) to real (PostgreSQL) and float (Oracle) to numeric (PostgreSQL).

 Real does not store the value as exact number in database. For example, 0.8 is stored as 0.80000011920929. On compare it always shows extra values, but it does not impact the repair functionality.

#### Interval:

- Heterogeneous: During repair Veridata may insert/update 00 to column.
   Oracle GoldenGate Veridata currently uses CAST (? AS INTERVAL) as part of its insert and update queries. This cast defaults to INTERVAL SECOND. Therefore, when values like '05', '57', or '98' are used as cast, they result in values, such as:
  - \* 5 --> 00:00:05
  - \* 57 --> 00:00:57
  - \* 98 --> 00:01:38

Therefore, if these values are inserted/updated to a column with INTERVAL HOUR, the result is 00. The same is for other type of INTERVAL.

#### citext

Repair function is not fully supported when citext is mapped to a string-like data type on either source or target side. When citext is on source side, following change is needed for repair to work: go to **Profile Configuration > Edit Existing Profile > Repair**, and unselect the **Check Changed Values** setting.

### 2.4.4 Supported Datatypes - MySQL, MariaDB

The Oracle GoldenGate Veridata supports the listed datatypes for the MySQL, MariaDB databases:

- BIGINT
- BINARY
- BIT
- BLOB
- CHAR
- DATE
- DATETIME
- DECIMAL
- DOUBLE
- ENUM
- FLOAT
- INT
- INTEGER
- LONGBLOB
- LONGTEXT
- MEDIUMBLOB
- MEDIUMINT



- NUMERIC
- REAL
- SET
- SMALLINT
- TEXT
- TIME
- TIMESTAMP
- TINYBLOB
- TINYINT
- TINYTEXT
- VARCHAR
- VARBINARY
- YEAR

#### For MySQL BIT Data Type

- Supported mapping Oracle/Postgres Data Type using veridata number compare format: Number
- Supported mapping Oracle Data Type using veridata string compare format with limitation: Char and Varchar.
- **Limitation**: not applicable to compare value with leading 0, such as 0001101
- Supported mapping Postgres Type using veridata string compare format with limitation: Char, Varchar and Bit varying
- **Limitation**: not applicable to compare value with leading 0, like 0001101
- Not supported Oracle Data Type: Binary and Varbinary

#### **Limitations and Clarifications**

- For Oracle GoldenGate Veridata repair to properly throw errors when invalid or missing values are detected, user should set SQL Mode to "strict". This can be set on global level when MySQL starts, or session level via connection string in agent.properties, for example: database.url=jdbc:mysql://host:3306?sessionVariables=sql\_mode=(select concat(@@SESSION.sql\_mode,',STRICT\_TRANS\_TABLES'))
- When using REAL as primary key, MySQL client and Veridata may not able to retrieve certain rows (for example, WHERE `PKCOL` = '-99.9999'), Oracle GoldenGate Veridata skips these rows during compare and repair.
- When using nonstandard FLOAT(M,D) and DOUBLE(M,D), the (M,D) in DDL need to match the actual precision and scale stored, and both Source and Target need to be MySQL.

# 2.4.5 Supported Datatypes - DB2 LUW

The Oracle GoldenGate Veridata supports the listed datatypes for the DB2 LUW database:

- BIGINT
- BLOB
- CHAR
- CHAR FOR BIT DATA



- CLOB
- DATE
- DBCLOB
- DECFLOAT
- DECIMAL
- DOUBLE
- FLOAT
- GRAPHIC
- INTEGER
- LONG VARCHAR
- LONG VARCHAR FOR BIT DATA
- LONG VARGRAPHIC
- NUMERIC
- REAL
- SMALLINT
- TIME
- TIMESTAMP
- VARCHAR
- VARCHAR FOR BIT DATA
- VARGRAPHIC
- XML

# 2.4.6 Supported Datatypes - DB2 for i

The Oracle GoldenGate Veridata supports the listed datatypes for DB2 for i:

- BIGINT
- BLOB
- CHAR
- CHAR FOR BIT DATA
- CLOB
- DATE
- DBCLOB
- DECFLOAT
- DECIMAL
- DOUBLE
- FLOAT
- GRAPHIC
- INTEGER



- LONG VARCHAR
- LONG VARCHAR FOR BIT DATA
- LONG VARGRAPHIC
- NUMERIC
- REAL
- ROWID
- SMALLINT
- TIME
- TIMESTAMP
- VARCHAR
- VARCHAR FOR BIT DATA
- VARGRAPHIC
- XML

#### **Limitations and Clarifications**

• The decimal part of REAL value from the wldb2 jdbc driver always store in the 16 digits precision, for example, 0.8 converts 0.800000011920929 and compare will always fail if the decimal values are present.

# 2.4.7 Supported Datatypes - DB2 z/OS

The Oracle GoldenGate Veridata supports the listed datatypes for the DB2 z/OS database:

- BIGINT
- BINARY
- BLOB
- CHAR
- CHAR FOR BIT DATA
- CLOB
- DATE
- DBCLOB
- DECFLOAT
- DECIMAL
- DOUBLE
- FLOAT
- GRAPHIC
- INTEGER
- LONG VARCHAR
- LONG VARCHAR FOR BIT DATA
- LONG VARGRAPHIC
- NUMERIC



- REAL
- ROWID
- SMALLINT
- TIME
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- VARBINARY
- VARCHAR
- VARCHAR FOR BIT DATA
- VARGRAPHIC
- XML

# 2.4.8 Supported Datatypes - Sybase Adaptive Server Enterprise

The Oracle GoldenGate Veridata supports the listed datatypes for the Sybase database:

- bigdatetime
- bigint
- bigtime
- binary
- bit
- char
- date
- datetime
- decimal
- float
- image
- int
- money
- numeric
- real
- smalldatetime
- smallint
- smallmoney
- sysname
- text
- time
- timestamp
- tinyint



- unichar
- unitext
- univarchar
- unsigned bigint
- unsigned int
- unsigned smallint
- varbinary
- varchar

# 2.4.9 Supported Datatypes - Teradata Vantage

The Oracle GoldenGate Veridata supports the listed datatypes for the Teradata database:

- ARRAY
- BIGINT
- BLOB
- BYTE
- BYTEINT
- CHAR
- CLOB
- DATE
- DECIMAL
- DISTINCT
- FLOAT
- GRAPHIC
- INTEGER
- INTERVAL DAY
- INTERVAL DAY TO HOUR
- INTERVAL DAY TO MINUTE
- INTERVAL DAY TO SECOND
- INTERVAL HOUR
- INTERVAL HOUR TO MINUTE
- INTERVAL HOUR TO SECOND
- INTERVAL MINUTE
- INTERVAL MINUTE TO SECOND
- INTERVAL MONTH
- INTERVAL SECOND
- INTERVAL YEAR
- INTERVAL YEAR TO MONTH



- LONG VARCHAR
- LONG VARGRAPHIC
- NUMBER
- SMALLINT
- STRUCT
- TIME
- TIME WITH TIME ZONE
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- VARBYTE
- VARCHAR
- VARGRAPHIC

# 2.4.10 Supported Datatypes - Snowflake

The Oracle GoldenGate Veridata supports the listed datatypes for the Snowflake database:

- BIGINT
- BINARY
- BOOLEAN
- BYTEINT
- CHAR
- CHAR VARYING
- CHARACTER
- DATE
- DATETIME
- DECIMAL
- DOUBLE
- DOUBLE PRECISION
- FLOAT
- FLOAT4
- FLOAT8
- INT,
- INTEGER
- NCHAR
- NCHAR VARYING
- NUMBER
- NUMERIC
- NVARCHAR



- NVARCHAR2
- REAL
- SMALLINT
- STRING
- TEXT
- TIME
- TIMESTAMP
- TIMESTAMP LTZ
- TIMESTAMP NTZ
- TIMESTAMP TZ
- TINYINT
- VARBINARY
- VARCHAR
- VARIANT
- VECTOR

#### Non-Supported Snowflake Data Types

- ARRAY
- OBJECT
- GEOGRAPHY
- GEOMETRY

### 2.4.11 Generic Limitations and Clarifications

This topic lists a few generic limitations and clarifications:

- As the floating-point data types are approximate values by definition, Oracle GoldenGate Veridata UI may display slightly different values than expected (for example, extra decimal values)
- For database with TIME data type that supports value outside range of 00:00:00 to 23:59.59, the compare will only work when Source and Target are the same database.
- Whenever there is a precision for any datatype, such as REAL/FLOAT or Double on the database side, Oracle GoldenGate Veridata rounds off the lower precision present either on the source or the target database. Veridata considers the lower precision of data during comparison phase.
- Oracle GoldenGate Veridata internally converts decimal numbers with a large scale by truncating tail digits. The limitations applies to all databases except Oracle.

This is because Oracle GoldenGate Veridata utilizes a template to mask decimal numbers, which, in cases of a large scale, leaves few digits for representing the scale part. For example, a number like 6.710123456789e-26, with a scale of 38 digits, would be truncated by Oracle GoldenGate Veridata to retain the first 15 digits, converting it to 0. For scenarios where preserving all digits of the scale is crucial, Oracle recommends using the string or other compare formats instead of the number compare format when configuring the compare pair.



# 2.5 Oracle GoldenGate Veridata Agent System Requirements

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

#### **Comparing Multi-Byte Data**

The following considerations apply when you are comparing tables with multibyte data:

- A Java agent should be used for all platforms except NonStop, which has only a C-agent.
- The Java agent uses the UTF-8 character for comparing character data. Out-of-sync data is written to the report file using the UTF-8 character set.

This topic contains the following:

- Disk and Memory Requirements for the Agent Component
- Database Privileges for the Agent Component

### 2.5.1 Disk and Memory Requirements for the Agent Component

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

### 2.5.2 Database Privileges for the Agent Component

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

#### Required database privileges for Oracle GoldenGate Veridata Agent

#### IBM Db2

SELECT privileges on the tables that will be compared.



#### **Oracle Database**

- GRANT CONNECT
- GRANT SELECT on the tables to be compared. It is recommended, but not necessary, to GRANT SELECT ANY TABLE.
- SELECT\_CATALOG\_ROLE
- EXECUTE\_CATALOG\_ROLE (for GET\_TAG and SET\_TAG procedures)
- CREATE TABLE (for COOS Join in case of tables with no primary/unique constraint)
- CREATE TABLESPACE (for COOS Join in case of tables with no primary/unique constraint)
- GRANT UPDATE, DELETE, INSERT (for COOS Join in case of tables with no primary/unique constraint)

#### **HPE NonStop**

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

#### Microsoft 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.

#### **Sybase Adaptive Server Enterprise**

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

#### Teradata Vantage

SELECT privileges on the tables to be compared.

#### **Apache Hive**

SELECT privileges on the tables to be compared.

#### Required Database Privileges for Using the Repair Feature

For all databases, the database user must have the UPDATE, INSERT, and DELETE privileges on the tables to be repaired.

For Sybase database, if the table has triggers and suppression of triggers enabled, the database user must have the replication\_role privilege.



For SQL Server database, If the table being repaired has 'identity columns', the Repair User specified must be either the table owner, or should have ALTER permission on the table that is being repaired.

# 2.6 Oracle GoldenGate Veridata Server System Requirements

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

Oracle GoldenGate Veridata server is designed to run on a dedicated Linux host only. Ensure to stop all other applications and remove any MySQL instances before server install to avoid port conflicts.

Oracle recommends that you use the certification matrix and system requirements documents with each other to verify that your environment meets the requirements for installation. See GoldenGate Certifications.

- Location for the Server Component
- Disk and Memory Requirements for the Server Component

### 2.6.1 Location for the Server Component

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

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

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

### 2.6.2 Disk and Memory Requirements for the Server Component

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

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

((number of cols in key + 1) \* 4) + 16 + (comparison width of a key col)

#### Where:

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

Comparison format data sizes:



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

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

((1+1)\*4)+16+9=33 bytes of memory for this row



#### (i) Note

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

Oracle GoldenGate Veridata uses an external merge sort to sort the data. As data is received from the agent, the rows are sorted in memory. When a memory buffer is full, the sorted rows are written to disk.

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

- In-Memory Sort: This sorts the data entirely in memory and is the fastest method, but the memory requirements may exceed what is available. The sort memory must be approximately 2.5 times larger than the size of the data set.
- One Disk Pass: This sorts data and writes to the disk only once. It requires sort disk space equal to the size of the data set. This process is almost as fast as the in-memory sort and the memory requirements are lower. In general the Oracle GoldenGate Veridata server can write the rows to disk faster than the agent can read them from the database.
- Two Disk Passes: This sorts and writes to the disk twice, requiring sort disk space twice the size of the data set. Although the disk requirement is greater, very large data sets can be sorted with a reasonable amount of memory.
- Three or More Disk Passes: After all of the rows have been received from the agent, additional sorting may be required before the rows are ready for the final write to disk. If it is necessary to access the disk three or more times, the required sort space will be three or more times the data set size. This is slow and should be avoided.

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

On 64-bit systems, more memory can be addressed, so more data can be stored in main memory instead of on slower disk devices. The memory that is used in the initial comparison step is not necessarily all released at once to be available for the confirmation step.



Consequently, some memory will be shared between processes. When the sort cannot hold all of the rows in memory, it uses disk storage.

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

- The temporary space should be located on a reasonably fast file system. A network file system located on a remote server may slow the comparison processing.
- You can increase disk I/O performance by specifying multiple temporary directories with profile settings. For maximum benefit, put the directories on different physical disks.
- You can use a profile setting to terminate the confirmation step after a given number of outof-sync rows, to work around resource limitations.
- Additional memory properties can be controlled with server parameters. See <u>Server</u> Parameters.

### 2.7 Oracle GoldenGate Veridata Distribution

This topic details how to download the files of Oracle GoldenGate Veridata files and C-Agent.

- Downloading Oracle GoldenGate Veridata
- Downloading Oracle GoldenGate Veridata C-Agent and Java Agent

### 2.7.1 Downloading Oracle GoldenGate Veridata

A distribution is an archive file containing an installer; when you run the installer, the set of Oracle GoldenGate Veridata components and feature that are included with the distribution are installed. You will need a certified JDK on your system in order to be able to run the installer. For more information, see **Prerequistes** in <u>Installing and Running Oracle GoldenGate Veridata</u>

You can download Oracle GoldenGate Veridata from the Oracle GoldenGate Downloads page at <a href="https://www.oracle.com/middleware/technologies/goldengate-downloads.html">https://www.oracle.com/middleware/technologies/goldengate-downloads.html</a> and from the Oracle Software Delivery Cloud site, at <a href="https://edelivery.oracle.com/osdc/faces/SoftwareDelivery">https://edelivery.oracle.com/osdc/faces/SoftwareDelivery</a>.

The following table describes the products and feature sets in Oracle GoldenGate Veridata.

Table 2-1 Oracle GoldenGate Veridata Product and Feature Sets

Product	Feature Set	Description
Oracle GoldenGate Veridata	Veridata Server	The Veridata Server component includes the web server, the web application, and command-line utilities such as vericom.
	Veridata Agent	This component installs only the Veridata Agent.
Internal Features	OPatch	The OPatch utility is a tool that allows the application and rollback of interim patches to Oracle products.

### 2.7.2 Downloading Oracle GoldenGate Veridata C-Agent and Java Agent

Oracle GoldenGate Veridata C-Agent and the Java Agent platforms are available for download on Oracle Technology Network Oracle GoldenGate Downloads page at:

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

### Install

- Installing and Running Oracle GoldenGate Veridata
- Logging
- Running the Configuration Assistant
- Installing and Configuring Source and Target Agents
- Silent Installation
- Uninstalling Oracle GoldenGate Veridata
- Installing Oracle GoldenGate Veridata C-Agent
- Best Practices

# 3.1 Installing and Running Oracle GoldenGate Veridata

This article describes all the steps to install, configure, and uninstall Oracle GoldenGate Veridata.

### **Prerequisites**

- If your system does not have graphical desktop, then ensure that you setup VNC and connect to it using a VNC client for graphical desktop: vncserver
- Ensure to install the JDK software.
  - 1. Go to <a href="http://www.oracle.com/technetwork/java/javase/downloads/index.html">http://www.oracle.com/technetwork/java/javase/downloads/index.html</a>.
  - Select JDK version 17 and click **Download**. You can change the default installation location.
  - 3. Set JAVA HOME.

#### Korn and bash shells:

export JAVA\_HOME=jdk-install-dir export PATH=\$JAVA\_HOME/bin:\$PATH

#### Bourne shell:

JAVA\_HOME=jdk-install-dir export JAVA\_HOME PATH=\$JAVA\_HOME/bin:\$PATH export PATH

#### C shell:

setenv JAVA\_HOME jdk-install-dir setenv PATH \$JAVA\_HOME/bin:\$PATH

Ensure not to run additional application on the same host as it may have a strong impact on the performance of Oracle GoldenGate Veridata. Note that Veridata is a Java application designed to run on dedicated host with optimal heap and JVM settings.

#### Installing Oracle GoldenGate Veridata

To install Oracle GoldenGate Veridata

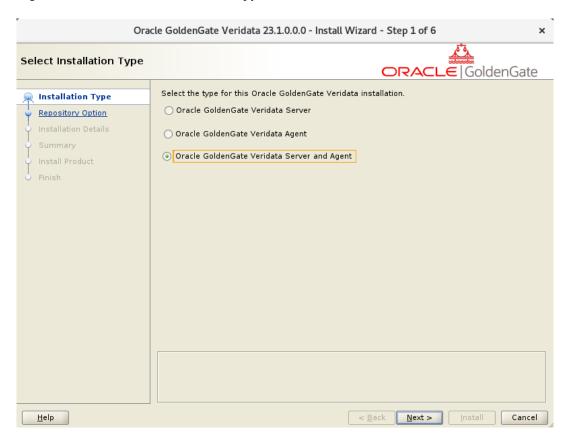


- 1. Download the installer. See <u>Downloading Oracle GoldenGate Veridata</u>.
- 2. Run the following command to launch the installer UI:

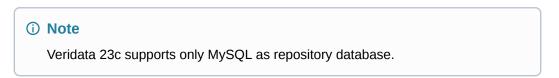
unzip fbo\_oggvdt\_linux\_services\_shiphome.zip cd fbo\_oggvdt\_linux\_services\_shiphome/Disk1/./runInstaller

- In the Select Installation Type page, select one of the following 3 options to set the type for the Oracle GoldenGate Veridata installation and click Next:
  - Oracle GoldenGate Veridata Server
  - Oracle GoldenGate Veridata Agent
  - · Oracle GoldenGate Veridata Server and Agent

Figure 3-1 Select the Installation Type



4. If the Installation Type is chosen as Oracle GoldenGate Veridata Server or Oracle GoldenGate Veridata Server and Agent in step 3, then the Repository Option page is displayed.



Select one of the following 2 options to set the Repository type for the Oracle GoldenGate Veridata Server and click **Next**:

Install a new MySQL database on this host as the repository

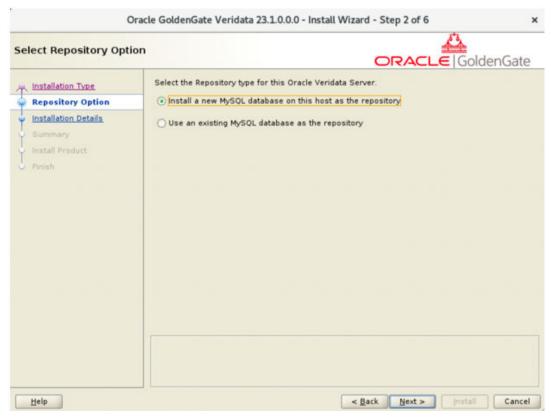


Using an existing MySQL database as the repository

(i) Note

If the **Using an existing MySQL database as the repository** option is selected, then ensure that the existing MySQL database is a commercial version 8.0.34 and above. It can be a user installed or system preinstalled MySQL database. However, ensure that it is not an embedded MySQL database from a prior Oracle Veridata installation.

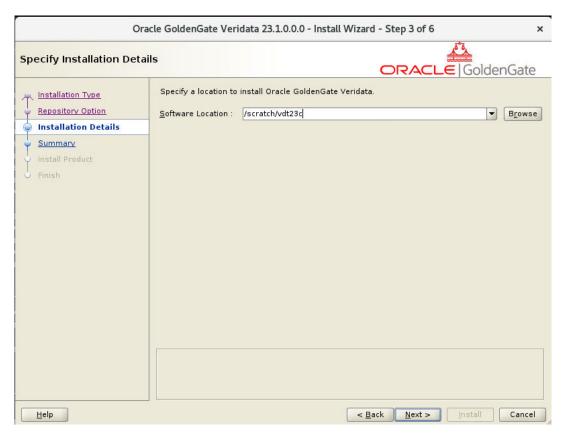
Figure 3-2 Select Repository Option



 In the Specify Installation Details page, specify the Software Location to install Oracle GoldenGate Veridata. For example, /scratch/vdt23c



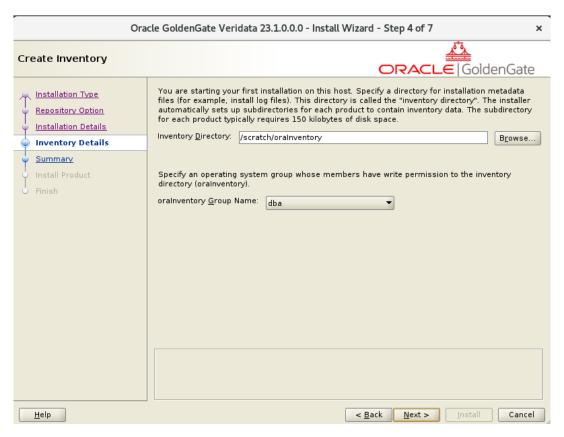
Figure 3-3 Specify Installation Details



6. In the **Create Inventory** page, you can either retain the defaults or provide or enter a new directory for the inventory and click **Next**.



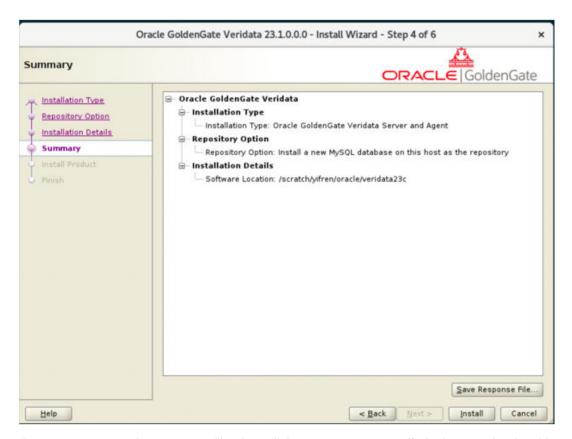
Figure 3-4 Inventory Details



7. Review the content in the Summary page, click Install.



Figure 3-5 Installation Summary

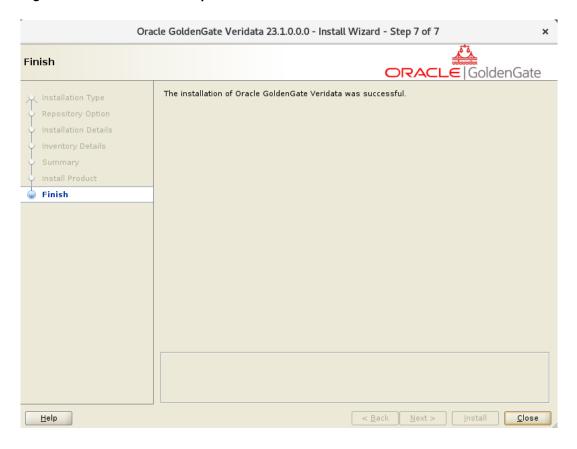


8. If you want to save the response file, then click **Save Response File** before you begin with installation.

At the end of the installation process, if the Oracle GoldenGate Veridata installation is complete, then the Installation Complete page is shown as follows:



Figure 3-6 Installation complete



# 3.2 Logging

- Logs and Setting Log Levels
- Agent
- Server

## 3.2.1 Logs and Setting Log Levels

Oracle GoldenGate Veridata maintains logs for both the agent and the server, helping with monitoring, troubleshooting, and performance analysis.

- The agent generates:
  - Main logs
  - Query logs for executed SQL queries
  - Performance logs for tracking execution efficiency

Veridata agent logs are categorized into different log levels to control the verbosity of logging:

- TRACE Detailed debugging information
- NOTIFICATION General operational messages
- WARNING Potential issues that may require attention
- ERROR Errors that impact functionality but do not stop execution



- INCIDENT\_ERROR Critical errors requiring immediate action
- The server generates:
  - Server logs
  - Api logs
  - Performance logs for monitoring system performance

Veridata server logs are categorized into different log levels to control the verbosity of logging:

SEVERE

Indicates serious failures that require immediate attention.

WARNING

Indicates potential problems that might cause issues in the future.

INFO

Used for general information messages about the application's normal execution.

FINE

Provides general tracing information for debugging.

FINER

More detailed tracing messages than FINE.

FINEST (Lowest priority)

Most detailed tracing information, usually at a very granular level.

ALL

Captures all messages at all levels (from FINEST to SEVERE).



Production systems typically log at INFO or WARNING levels. Debugging scenarios may use FINE, FINER, or FINEST.

### 3.2.2 Agent

- Agent Logs Location
- Updating Agent Log Settings
- Enabling Performance Logs
- Enabling Trace Logs
- Enabling Query Logs
- Modifying Log Size

### 3.2.2.1 Agent Logs Location

Agent logs are stored in the following directory: \$AGENT\_DEPLOYED\_LOCATION/logs/



The main agent logs are recorded in: veridata-agent.log.

The performance logs are stored in: vdtperf-agent.log.

### 3.2.2.2 Updating Agent Log Settings

Agent log settings can be configured in the odl.xml file.

File Location: \$AGENT\_DEPLOYED\_LOCATION/config/odl.xml

### 3.2.2.3 Enabling Performance Logs

**1.** Modify the odl.xml file and change level from NOTIFICATION to TRACE for logger with name oracle.veridata.agent.performance.

For example:

```
<logger name='oracle.veridata.agent.performance' level='TRACE:1'
     useParentHandlers='false'>
     <handler name='perf-handler' />
     <handler name='my-console-handler' />
     </logger>
```

2. Restart the agent.

If this logger is not present in odl.xml then:

a. Add below log handler under <log\_handlers>

```
<log_handler name='perf-handler'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='path' value='${agentHome}/logs/vdtperf-${agent.id}.log'/>
<property name='maxFileSize' value='10485760'/>
<property name='maxLogSize' value='104857600'/>
<property name='useSourceClassAndMethod' value='TRACE:1'/>
</log_handler>
```

**b.** Under the <loggers> tag add the following:

**3.** Restart the agent.

### 3.2.2.4 Enabling Trace Logs

Modify the odl.xml file and change level from NOTIFICATION to TRACE for loggers having name oracle.veridata. Restart the agent.

### For example:

```
<logger name='oracle.veridata' level='TRACE:1'
    useParentHandlers='false'>
        <handler name='odl-handler' />
        <handler name='my-console-handler' />
</logger>
<logger name='oracle.veridata.XML' level='TRACE:1'
    useParentHandlers='false'>
        <handler name='odl-handler' />
```



```
<handler name='my-console-handler'/> </logger>
```

### 3.2.2.5 Enabling Query Logs

 Modify the odl.xml file and change level from NOTIFICATION to TRACE for loggers having name oracle.veridata.agent.query.

```
<logger name='oracle.veridata.agent.query' level='TRACE:1'
useParentHandlers='false'>
<handler name='odl-handler' />
<handler name='my-console-handler' />
</logger>
```

2. Restart the agent.

### 3.2.2.6 Modifying Log Size

The log rotation is enabled by default, and therefore older logs get deleted when log size reached a certain size. However, this can be controlled as described below.

Log size is specified per log handler in the odl.xml file. This is controlled by the following 2 properties:

- maxFileSize Defines the maximum size (in bytes) a log file can reach before it gets rotated and a new log file is created.
- maxLogSize It controls the max log size (in bytes) before rolling over.

### For all the logs:

Find the log handler with name odl-handler and change values of maxFileSize and maxLogSize, then restart the agent.

#### For Performance Logs:

Find the log handler with name perf-handler and change values of maxFileSize and maxLogSize, then restart the agent.

```
<log_handler name='perf-handler'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='path' value='${agentHome}/logs/vdtperf-${agent.id}.log'/>
<property name='maxFileSize' value='10485760'/>
<property name='maxLogSize' value='104857600'/>
<property name='useSourceClassAndMethod' value='TRACE:1'/>
</log_handler>
```



Values of maxFileSize and maxLogSize are in bytes, therefore 10485760 is 10 MB.



### 3.2.3 Server

- Server Log Location
- Updating Server Log Levels
- Changing Log Size

### 3.2.3.1 Server Log Location

All the Server logs are stored in the following directory: <\textit{VERIDATA\_HOME} > \textit{veridata/logs}.

Server API logs are recorded in vdtapi.log. The format of the Server API log files is as follows: vdtapi.log.x.

For example, vdtapi.log.0. Older logs are named as vdtapi.log.1, vdtapi.log.2, or vdtapi.log.n.

The Server logs are recorded in vdtserver.log. The format of the Server log files is as follows: vdtserver.log.x.

For example, vdtserver.log.0. Older logs are named as vdtserver.log.1, vdtserver.log.2, or vdtserver.log.n. vdtapi.log.0 file will have the latest logs.

Server performance logs are recorded in vdtperf.log.

#### (i) Note

- Log configurations can also be updated from the config file located at: <*VERIDATA\_HOME*>/config/logging.properties.
- If you update the config file, you will need to restart the server.
- You can avoid a restart by using APIs. See <u>Updating Server Log Levels</u>.

### 3.2.3.2 Updating Server Log Levels

Server log levels can be configured using update log config API. For more information, see <u>REST API for GoldenGate Veridata</u>. This API will be used to set log levels for:

#### 1. Setting Log Levels for All Logs:

To apply a log level to all logs at once, send the following request body:

```
{
"level":<global_log_level>
}
```

Valid values for <global\_log\_level>: ALL, SEVERE, WARNING, INFO, FINE, FINEST

### 2. Configuring Performance Logs:

To enable performance logging, use the following request body:

```
{
    "level": <global_log_level>,
    "logComponents":
    "oracle.veridata.server.performance=FINE"
}
```



Here value of logComponents can be any of the following values:

```
"oracle.veridata.server.performance=FINE"
"oracle.veridata.server.performance=FINER"
"oracle.veridata.server.performance=FINEST"
```

#### 3. Configuring Server Logs

By default, INFO level server logs are written to the vdtserver.log file. To modify the server log level, use the following request body:

```
{
    "level": <global_log_level>,
    "logComponents": "oracle.veridata.server=<desired_log_level>"
}
```

Valid values for <desired\_log\_level>: ALL, SEVERE, WARNING, INFO, FINE, FINER, FINEST

Configuring Log Handlers (File & Console): You can choose where the logs should be stored:

FileHandler: Stores logs in a file.

**ConsoleHandler**: Displays logs in the console.

You can enable both handlers simultaneously.

```
{
    "level": <global_log_level>,
    "logHandlers": "FileHandler,ConsoleHandler"
}
```

### 3.2.3.3 Changing Log Size

Log rotation is enabled by default, to ensure older logs are deleted when they reach a specified

Log size is controlled by <VERIDATA\_HOME>/config/logging.properties file, where each log handler has two key properties:

- FileHandler.limit Defines the maximum size (in bytes) a log file can reach before it is rotated (a new file is created). This is basically the size of 1 log file.
- FileHandler.count Defines the number of log files to keep in the rotation. When the number of log files exceeds count, the oldest file is deleted.

For example: To change the file size to 10MB and log file count to 10 for perf logs, modify as follows:

```
com.oracle.goldengate.veridata.logging.PerfFileHandler.limit=10485760 com.oracle.goldengate.veridata.logging.PerfFileHandler.count=10
```

Note that you will need to restart the server after this update.

# 3.3 Running the Configuration Assistant

- 1. After the Oracle GoldenGate Veridata installation is complete, navigate to the bin directory, and run vdtca.sh to display the Configuration Assistant.
- 2. If you choose to install a new MySQL instance, then ensure that the system does not have any other MySQL database installed on the same node.



### ① Note

The Schema Prefix field supports only the following specific set of characters, as described in <u>Permitted characters in unquoted identifiers</u> of the MySQL Reference Manual:

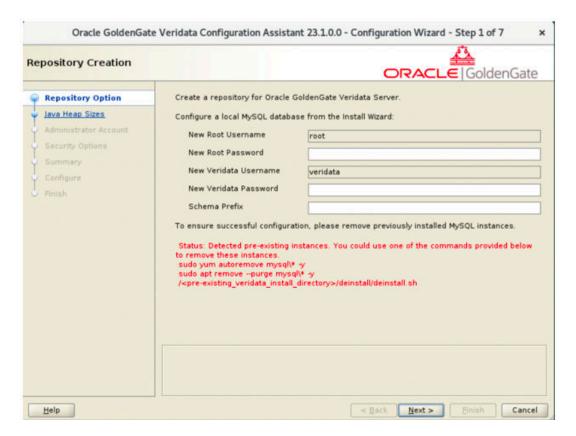
- ASCII: [0-9,a-z,A-Z\$ ] (basic Latin letters, digits 0-9, dollar, underscore)
- Extended: U+0080 .. U+FFFF

Certain special characters, such as hyphens or dashes (-) , are not allowed in the schema prefix.

### (i) Note

If the Configuration Assistant detects a pre-existing MySQL instance in the system, then it displays the information with red text shown as follows:

Figure 3-7 Repository Creation Page for a New MySQL Install







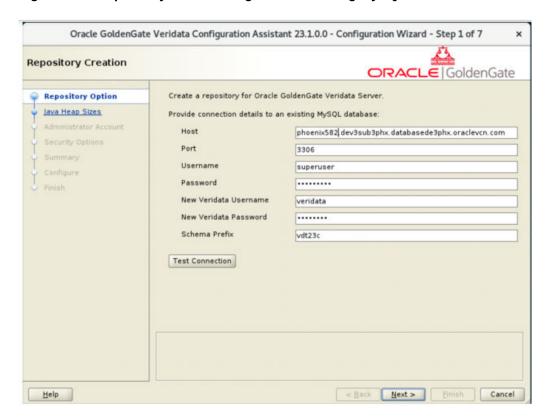
If the Configuration Assistant shows a warning message saying it has detected a pre-existing MySQL for the new MySQL installation option, then run the following command: yum list installed | grep mysql and verify if any MySQL database or package is installed in the system. If you are sure that the MySQL database does not exist in the system, then you can safely ignore the warning and proceed with the configuration. For example, pcp-pmda-mysql.x86\_64 package can cause a false warning.

If MySQL is an existing database, then in the **Repository Creation** page, enter the database details for all the text fields.

- The New Veridata Username and the Username cannot be same. The new veridata
  user which gets created as part of this process will be allowed to access only the
  Veridata repository being created.
- Ensure that the user given in Username has full privilege on the Mysql instance as this
  user would create a veridata repository in database. If the user does not have the
  sufficient privileges then the same can be provided by running below queries in
  MySQL:

CREATE USER 'superuser'@'%' IDENTIFIED BY 'superpassword'; GRANT ALL PRIVILEGES ON \*.\* TO 'superuser'@'%' WITH GRANT OPTION; FLUSH PRIVILEGES; SHOW GRANTS FOR 'superuser'@'%';

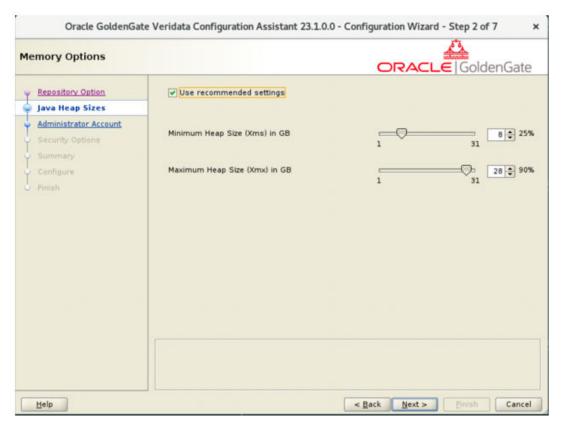
Figure 3-8 Repository Creation Page for an Existing MySQL





3. In the Memory Options page, for the Java Heap Sizes, select Use recommended settings (or modify according to your requirements) and click Next.

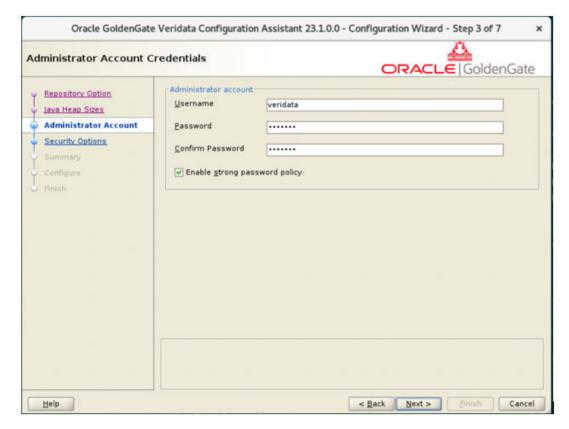
Figure 3-9 Memory Options



 In the Administrator Account Credentials page, enter the Veridata Administrator Username, Password, and Confirm Password and click Next.



Figure 3-10 Administrator Account Credentials



5. The Security Options step of the Oracle GoldenGate Veridata Configuration Assistant allows users to configure the security settings for the Web User Interface (UI) by selecting encryption options, certificates, and communication ports. The primary function of this screen is to secure the connection between the client and the server using SSL/TLS protocols. This point outlines each of the available options and describes how to configure them.

### a. SSL / TLS Security for Web UI

- Purpose: Enabling this option ensures that communication between the Web UI and users is encrypted using SSL (Secure Sockets Layer) or TLS (Transport Layer Security).
- Default Setting: This checkbox is enabled by default, meaning SSL/TLS is turned on.
- **Implications**: If disabled, the communication will not be encrypted, making it vulnerable to eavesdropping and other security threats. You can opt in to disable SSL/TLS for faster and simpler deployment in a test environment.



### Note

The SSL / TLS Security for Web requires 8830 port to be open apart from the selected port. If 8830 the port is not open, then Oracle GoldenGate Veridata server does not start after the installation. In such a case, do the following after installation:

- i. Update the following property in <*VERIDATA\_HOME*>/config/oggvdt\_cainput.properties server.port=8830 to an available port.
- ii. Start the Oracle GoldenGate Veridata server from <*VERIDATA\_HOME*>/bin: /run.sh

### **b.** Certificate Configuration

To secure the Web UI, Oracle GoldenGate Veridata uses X.509 certificates. The configuration wizard provides two ways to specify certificates outlined as follows:

### i. Use Veridata Self-Signed Certificate:

 Purpose: Use this option if you want Veridata to automatically generate a selfsigned SSL certificate.

#### Pros:

- Quick and easy setup with no need for external certificate files
- Useful in non-production environments or internal use cases where a trusted Certificate Authority (CA) is not necessary.

#### Cons:

- Self-signed certificates are not trusted by web browsers without manual intervention.
- Not suitable for production environments where certificates must be validated by a third-party CA.

#### ii. Upload Custom PEM Files

- **Purpose**: This option allows users to specify their own SSL certificate, private key, and CA certificate files (in PEM format).
- **Required Files**: This is the public certificate (usually with a .pem or .crt extension) that will be used to identify the server to clients.
  - File Input Field: Certificate
  - Example Path: /scratch/vdt23c/web/certificate.pem
- CA Certificate File: This file contains the certificate(s) of the Certificate
   Authority (CA) that signed the server certificate. The CA file is used to verify
   that the server certificate is trusted.
  - File Input Field: CA Certificate File
  - Example Path: /scratch/vdt23c/web/ca.pem
- When to Use: This option is best suited for production environments where an
  official SSL certificate issued by a trusted CA (such as Let's Encrypt, DigiCert,
  or GoDaddy) is required.

#### c. Port Configuration

Oracle GoldenGate Veridata operates over specific ports, and users can define whether to use the default port or a custom one.



#### Use Default Port (8831)

 Purpose: The default port for SSL/TLS connections to the Veridata Web UI is 8831.

#### When to Use

- Suitable for environments where port 8831 is open and available.
- Easier to configure, especially in less restrictive network environments.

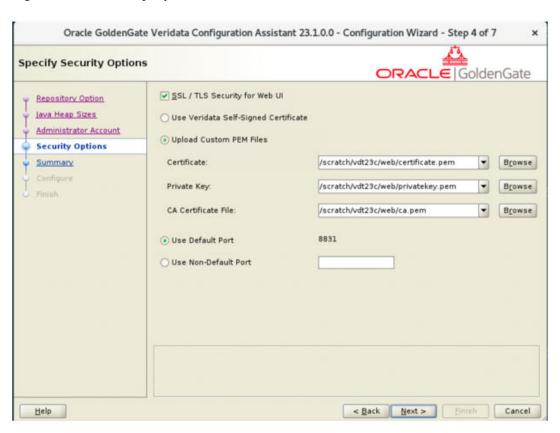
#### ii. Use Non-Default Port

 Purpose: Allows users to specify a custom port for Veridata to use for SSL/TLS traffic instead of the default 8831.

#### When to Use

- In cases where the default port (8831) is blocked or already in use.
- When a specific port policy is enforced by your IT or security team (for example, using port 443 for HTTPS traffic).
- Configuration: Enter the custom port number into the input field provided.

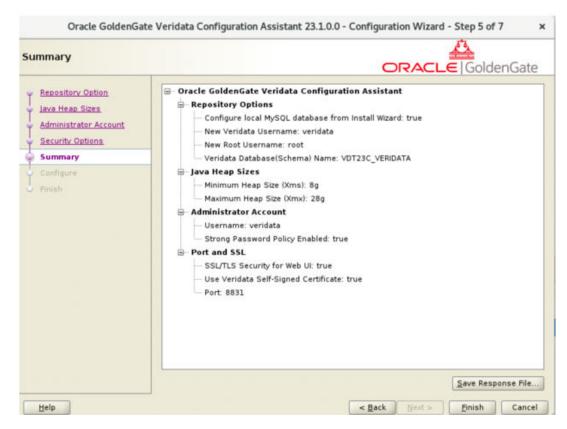
Figure 3-11 Security Options



6. Review the configuration summary on the **Summary** page. Click **Save Response File** to save the respose file if needed.



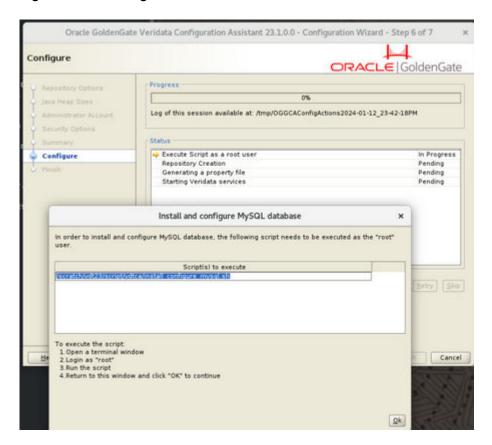
Figure 3-12 Conifuguration Assistant Summary



7. Open a terminal and run sudo su to change to root. After you see the following screen, run the script as root.



Figure 3-13 Configure root



### Note

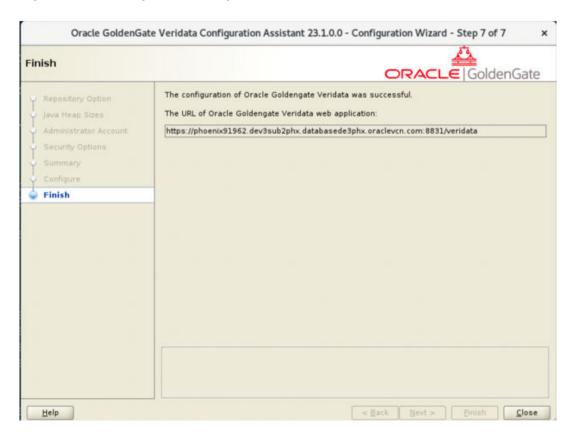
The root script should typically take 10-15 seconds.

If the <code>install\_configure\_mysql.sh</code> script was executed successfully, the console will display the following information:

- The Configuration Assistant will set the root user password as what you entered in the next stage of veridata repository creation.
- MySQL has been installed and started.
- Please return to the Configuration Assistant and click OK to continue.
   Then follow the console instruction, return to the Configuration Assistant and click OK to continue.
- 8. After a successful configuration, the **Finish** page displays the URL of Oracle Goldengate Veridata web application. Input the URL to a web browser to verify that the Oracle GoldenGate Veridata server is running.



Figure 3-14 Configuration Complete



# 3.4 Installing and Configuring Source and Target Agents

From the agent directory, install and configure source and target agents.

- About Oracle GoldenGate Veridata Agent Deployment Scripts
- Configuring Oracle GoldenGate Veridata Agent
- Starting the Oracle GoldenGate Veridata Agent
- Using the Oracle GoldenGate Veridata Agent Deployment Script in Debug Mode
- Reloading the Logging Properties of the Veridata Agent

# 3.4.1 About Oracle GoldenGate Veridata Agent Deployment Scripts

After installing Oracle GoldenGate Veridata Agent using Oracle Universal Installer, you must deploy the agent to a non Oracle Home location and configure the agent before running comparison jobs using a deployment script provided in the installation.

The following table describes the directories and the variables that are used when referring to those directories in this section.



Directories in an Oracle GoldenGate Veridata Installation Table 3-1

Directory Variable	Directory Path
VERIDATA_HOME	This is the home directory where Oracle GoldenGate Veridata is installed. For example, <i>VERIDATA_HOME</i> in this documentation is / scratch/vdt23c.
VERIDATA_AGENT_HOME	VERIDATA_HOME/agent. This is the location where Oracle GoldenGate Veridata agent is installed.
AGENT_DEPLOY_LOCATION	This is the location where the Oracle GoldenGate Veridata Agent is deployed.
	Note that this location should be outside the VERIDATA_HOME.

The agent config.sh script located in the VERIDATA AGENT HOME directory is used for deploying the Oracle GoldenGate Veridata Agent.

#### Syntax:

export JAVA\_HOME=/usr/java/jdk17 cd/scratch/vdt23c/agent ./agent\_config.sh /scratch/agent1 ./agent\_config.sh /scratch/agent2 cd /scratch/agent1 cp sample\_properties/agent.properties.oracle agent.properties (EDIT agent.properties per documentation) ./agent.sh start agent.properties

The AGENT\_DEPLOY\_LOCATION can be an absolute path or a path relative to the location from where the script is running.



#### (i) Note

You must deploy the Oracle GoldenGate Veridata Agent to a directory outside VERIDATA HOME.

### 3.4.2 Configuring Oracle GoldenGate Veridata Agent

You must configure the Oracle GoldenGate Veridata Agent to use your database.

- Go to the agent deployed location *AGENT\_DEPLOY\_LOCATION*. There is an agent, properties, sample file in this directory that contains database related properties like JDBC URL and driver.
- Copy the agent.properties.sample file and rename to agent.properties.
- Most of the properties defined in the file have default values. However, you must update the following properties:
  - The server property is the port where the Oracle GoldenGate Veridata Agent listens for connection requests.

server.port= < Port Number>

For example, <*Port Number>* = 7862



b. The database.url specifies the JDBC connection URL for the database. Samples for all supported databases are provided in the file.

database.url=<Database URL>

For example, *<Database URL>*=jdbc:oracle:thin:@localhost:1521:orcl.

c. The server.jdbcDriver property specifies the list of JDBC driver JAR files. Sample lists for the supported databases are provided in the file.
Use the driver corresponding to the database.url in the preceding step.

server.jdbcDriver=<*JDBC Driver*>

For example, *<JDBC Driver>*= ojdbc11-23.2.0.0.jar.

4. Oracle, MySQL, PostgreSQL, SQL Server, DB2 and Sybase drivers are available in VERIDATA\_HOME/agent/drivers. For any other drivers you want to use, copy the driver to VERIDATA\_HOME/agent/drivers. For more information, see Agent Parameters - Connections.

After the Oracle GoldenGate Veridata agent is installed, you need to configure 2-way SSL. See Configuring Workflow for Two-Way SSL in Oracle GoldenGate Veridata 23c.

### 3.4.3 Starting the Oracle GoldenGate Veridata Agent

Go to the agent deployment location AGENT\_DEPLOY\_LOCATION and run following command.

\$ ./agent.sh start agent.properties

Where *agent.properties* is the properties file that contains your database properties.

# 3.4.4 Using the Oracle GoldenGate Veridata Agent Deployment Script in Debug Mode

For debugging issues with the Veridata Agent deployment, run the agent\_config script with an additional command line argument as follows:

#### Syntax:

<VERIDATA\_AGENT\_HOME>/agent\_config.sh AGENT\_DEPLOY\_LOCATION true

When this option is true, the debug logs are printed on the screen.

### 3.4.5 Reloading the Logging Properties of the Veridata Agent

You can reload logging information from the *VERIDATA\_AGENT\_HOME*/config/odl.xml configuration file to a running agent by using the reloadLog option. The changes in the odl.xml file are put into effect on the agent. The agent must be running for this command to work.

- 1. Open the command prompt and navigate to the directory where the agent is installed.
- 2. Go to <AGENT\_DEPLOY\_LOCATION> and run the following command:

./agent.sh reloadLog

If agent properties are stored in a custom file other than agent.properties, then run the following command:

./agent.sh reloadLog < AGENT\_PROPERTIES\_FILENAME>



### 3.5 Silent Installation

In the Summary page of the Installer, click Save Response File to save the response file.
The default name of the Installer's response file is vdt.rsp. The content of a sample vdt.rsp is
as follows:

oracle.install.responseFileVersion=/oracle/install/rspfmt\_ogginstall\_response\_schema\_v23\_1\_0
INSTALL\_OPTION=SERVERAGENT/SERVER/AGENT
IS\_NEWREPO=true/false
SOFTWARE\_LOCATION=<>
INVENTORY\_LOCATION=<>



Set INSTALL\_OPTION as:

- SERVERAGENT: To silent-install both the server and agent components.
- SERVER: To silent-install only the server component.
- AGENT: To silent-install only the agent component.

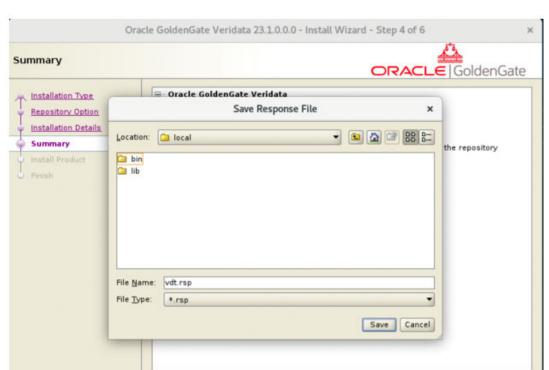


Figure 3-15 Silent Installation - Save Response

 In the Summary page of the Configuration Assistant, click Save Response File to save the response file. The default name of the Configuration Assistant's response file is vdtca.rsp.

< Back

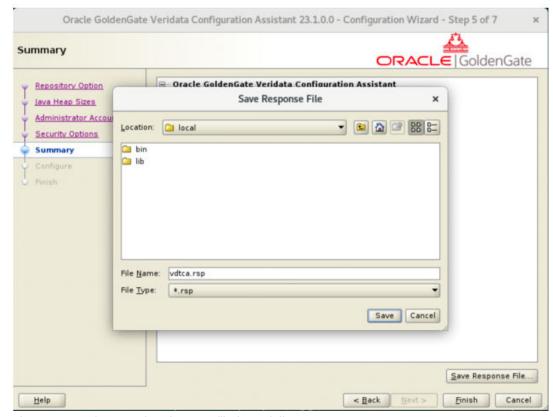
Help

Save Response File.

Install



Figure 3-16 Silent Install - Summary Page



The content of a sample vdtca.rsp file is as follows:

oracle.install.responseFileVersion=/oracle/install/rspfmt oggca response schema v23 1 0

IS\_NEW\_MYSQL=true

EXISTING\_MYSQL\_HOST=

EXISTING\_MYSQL\_PORT=

EXISTING\_MYSQL\_USERNAME=

EXISTING\_MYSQL\_PASSWORD=

NEW\_MYSQL\_ROOT\_USERNAME=<root-username>

NEW\_MYSQL\_ROOT\_PASSWORD=<password>

NEW MYSQL VERIDATA USERNAME=<username>

NEW MYSQL VERIDATA PASSWORD=<password>

SCHEMA\_PREFIX=VDT

MIN HEAP SIZE=4g

MAX\_HEAP\_SIZE=14g

ADMINISTRATOR\_USER=<admin\_username>

ADMINISTRATOR\_PASSWORD=<password>

STRONG\_PWD\_POLICY\_ENABLED=true

 $IS\_SSL\_ENABLED = false$ 

IS\_SELF\_SIGNED\_CERTIFICATE=

IS\_PEM\_FILES=

CERTIFICATE LOCATION=

PRIVATE KEY LOCATION=

CA\_CERTIFICATE\_LOCATION=

3. The following is the command to make a silent installation of the Installer: / fbo\_oggvdt\_linux\_services\_shiphome/Disk1/runInstaller -silent -responseFile /scratch/vdt.rsp. If the silent installation was successful, the console will display the following information.



The installation of Oracle Veridata Services was successful.

Please check '/scratch/oracle/app/oracle/product/oraInventory/logs/silentInstall2024-10-09\_08-18-05PM.log' for more details. Successfully Setup Software.

**4.** To install and configure the built-in MySQL, run the install\_configure\_mysql.sh with the following sudo command:

sudo <VERIDATA\_HOME>/script/vdtca/install\_configure\_mysql.sh

5. The following is the command to make a silent installation of the Configuration Assistant: / <*VERIDATA\_HOME*>/bin/vdtca.sh -silent -responseFile /scratch/vdtca.rsp. If the silent installation was successful, then the console will display the following information: Successfully Setup Software.

# 3.6 Uninstalling Oracle GoldenGate Veridata

 Run the following script to uininstall: ./<VERIDATA HOME>/deinstall/deinstall.sh

The uninstall script first shuts down the Web Server, then shuts down the MySQL server. However, the uninstall script does not shut down the MySQL server and displays the following warning message if it cannot detect whether the MySQL server is up and running: Unable to detect whether the MySQL server is up and running. Please manually check and shut it down if running.

- If the uninstall script does not stop the MySQL server, please manually stop it by using the MySQL command listed below. The MySQL command will prompt a password input. Use the root password you entered in the Configuration Assistant Repository Options step. < VERIDATA\_HOME>/mysql-commercial-8.0.34-linux-glibc2.17-x86\_64-minimal/bin/mysqladmin -u root -p shutdown.
  - Another way to manually stop the MySQL instance process is using the Linux OS command: kill -9 <mysql-instance-pid>.
- 3. If the folders are located within mysql-commercial-8.0.34-linux-glibc2.17-x86\_64-minimal, then take a backup of the folders data and the mysql-files from mysql-commercial-8.0.34-linux-glibc2.17-x86\_64-minimaldirectory.
- **4.** Remove the remaining MySQL-related directory using OS root user privileges: sudo rm -rf <*VERIDATA\_HOME*>/mysql-commercial-8.0.34-linux-glibc2.17-x86\_64-minimal

# 3.7 Installing Oracle GoldenGate Veridata C-Agent

Learn how to install the Oracle GoldenGate Veridata C-Agent on a NonStop platform. This chapter includes the following sections:

- Installation Overview
- Installing the C-Agent on a NonStop System

### 3.7.1 Installation Overview

These instructions are for installing a new, clean copy of the C-based Oracle GoldenGate Veridata Agent software.

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



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

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

- 1. Install the Oracle GoldenGate Veridata Agent files.
- 2. 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)
- 4. Configure the Manager process.
- (Optional) Add the following parameter that will rollover report files after the specified increment

VERIDATAREPORTAGE <nnn> [time units]

Time units are, one of:

DAY(s), HOUR(S), MINUTE(s), or SECOND(s)

For example:

**VERIDATAREPORTAGE 1 DAY** 

- Installing the Oracle GoldenGate Veridata Agent Files
- Copying VSNSERV to Remote Nodes
- Creating a GLOBALS File
- Configuring Manager

### 3.7.2.1 Installing the Oracle GoldenGate Veridata Agent Files

- Follow the steps in <u>Oracle GoldenGate Veridata Distribution</u> to download the Oracle GoldenGate Veridata Agent build file to a Windows workstation.
- 2. Using 7-zip, unzip the files to a temporary directory on your workstation.
- 3. Transfer the files in binary mode to the volume and subvolume on the NonStop Server where you want to install and run the agent. The agent software must be installed in a dedicated subvolume, including one that is separate from other Oracle GoldenGate software.
- 4. Execute the following TACL command:

unpak VERJ06PK (or VERH06PK), \$\*.\*.\*, <userid>, vol <volume>.<subvolume>, KEEP

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

Run veriinst

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

LUDAY File decomposion program T1255C06

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.

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 <u>Copying VSNSERV to Remote Nodes</u> and <u>Creating a GLOBALS File</u> as necessary for your environment.

After the Oracle GoldenGate Veridata agent is installed, you need to configure 2-way SSL. See Configuring Two-Way SSL for the NSK C-Agent on the Veridata Server.

### 3.7.2.2 Copying VSNSERV to Remote Nodes

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

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

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

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

### To copy VSNSERV to remote nodes:

- 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
FUP license <volume>.<subvolume>.VSNSERV

- 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 Oracle GoldenGate Veridata Agent installation directory. See <u>Creating a GLOBALS File</u>.

### 3.7.2.3 Creating a GLOBALS File

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

 Other Manager processes exist on this system, such as the one used by Oracle GoldenGate data synchronization software. A unique name for each Manager process



must be specified in this file with the GGSPREFIX parameter, including the one that will be used by the Veridata agent.

 Partitions for tables that will be compared with 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.
- 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.

### (i) 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.

### 3.7.2.4 Configuring Manager

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

RUN GGSCI

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

EDIT PARAMS 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 Oracle GoldenGate software.

PORT number



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

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

#### Where:

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

### 3.8 Best Practices

- Veridata Components Proximity
- Veridata Server and Agent Memory
- SSL Certificate Validation
- Database Permissions
- Data for Comparison
- Veridata Configurations
- Frequently Asked Questions (FAQ)

### 3.8.1 Veridata Components Proximity

For an optimal working of Oracle GoldenGate Veridata, Oracle recommends the following:

 Ensure that the Oracle GoldenGate Veridata Server and Veridata Repository are on the same node:

The Veridata Repository stores essential data needed for Veridata to function smoothly. To ensure a fast and reliable connection between the server and the repository, place them on the same node. This helps preventing delays. However, if that is not possible for some reason then ensure that both are on nearby nodes and the network connection between them is fast. Check the time taken in connecting from server node to repository node. Time taken should be in milli seconds or lesser than that. For more information, see <a href="How to check connection between 2 nodes">How to check connection between 2 nodes</a>.

Veridata Server and Veridata Agents:

Veridata Server and Veridata Agents can be on different nodes/machines. If the server and agents are on different nodes/machines, then ensure that the connection between them is fast as there are bound to be several calls between server and agents. Check the connection between the Veridata server and source agent. We also need to check



connection between Veridata server and target agent. For more information, see <u>How to</u> check connection between 2 nodes.

### Veridata Agent and the corresponding database on the same node:

A large amount of data is frequently transferred between the database and the agent, so a fast connection is essential. It is recommended to keep the source agent and source database on the same node. Similarly, place the target agent and target database on the same node. However, if that is not possible for some reason then ensure that both are on nearby nodes and network connection between them is fast. Check the time taken in connecting from agent node to DB node. Time taken should be in milli seconds or lesser than that. For more information, see <a href="How to check connection between 2 nodes">How to check connection between 2 nodes</a>.

Veridata Server and Sort Directory on the same node:

The server processes data in small chunks that fit into main memory, sorting them and storing them in temporary files within a designated directory called the sort directory. For optimal performance, the sort directory should be located on the same node as the server.

If not, connection time between server and sort directory should be in milli seconds or smaller than that. For more information, see How to check connection between 2 nodes.

• How to Check Connection between 2 Nodes

### 3.8.1.1 How to Check Connection between 2 Nodes

You can test connectivity between any two nodes or machines using these methods. For example: If you want to check the connection between the Veridata server and the source agent, then:

Call the Veridata server → node\_1

Call the source agent → node\_2

To Test Connectivity using Ping:

- 1. On the node\_1 open a terminal and run: ping [node\_2].
- Replace [node\_2] with the hostname or IP address of node\_2.
- 3. To measure latency and to check the speed of the connection, run ping -c 4 [node\_2]. The following is a sample output:

Success Case:

PING 192.168.1.2 (192.168.1.2): 56 data bytes

64 bytes from 192.168.1.2: icmp\_seq=0 ttl=64 time=0.123 ms

--- 192.168.1.2 ping statistics ---

4 packets transmitted, 4 packets received, 0.0% packet loss

 $round-trip\ min/avg/max/stddev = 0.123/0.124/0.125/0.001\ ms$ 

Note: If the time for round-trip is in range of milli seconds or less than then connection between source host target host is considered fast.

Failure Case:

PING 192.168.1.2 (192.168.1.2): 56 data bytes

Request timeout for icmp\_seq 0



```
--- 192.168.1.2 ping statistics ---
```

4 packets transmitted, 0 packets received, 100.0% packet loss

To check connectivity when Ping is not available:

- Use Telnet to check specific ports. On the node\_1, run: telnet [node\_2] [port]
- 2. Replace [node\_2] with the hostname or IP address of node\_2, and [port] with the specific port number. For example, 8089.

#### **Sample Output**

Success Case:

Trying 192.168.1.2...

Connected to 192.168.1.2.

Escape character is '^]'.

Failure Case:

Trying 192.168.1.2...

telnet: Unable to connect to remote host: Connection refused

To check connectivity when Ping is not available using Netcat (NC), On the node\_1, run: nc -zv [node\_2] [port]

#### Sample Output

Success Case:

Connection to 192.168.1.2 8089 port [tcp/\*] succeeded!

Failure Case:

nc: connect to 192.168.1.2 port 8089 (tcp) failed: Connection refused

### 3.8.2 Veridata Server and Agent Memory

For a proper working of Oracle GoldenGate Veridata, ensure to set correct memory for Veridata Server, Veridata agents and sort directory.

- Server Memory
- Agent Memory
- Memory for Sorting and Sort Directory

### 3.8.2.1 Server Memory

#### How to check current memory of server?

Memory settings are defined in file: <VERIDATA\_HOME > /config/oggvdt\_cainput.properties

Here jvm.memory.xms is the minimum heap and jvm.memory.xmx is the maximum heap.

### How to modify the memory for server?

Memory for the server can be modified by changing values of properties jvm.memory.xms and jvm.memory.xmx in file <*VERIDATA\_HOME*>/config/oggvdt\_cainput.properties.



This change will require Veridata server restart.

### What should be the optimum memory for server?

See Disk and Memory Requirements for the Server Component, to know size for 1 row.

**Total size for 1 table**: (size for 1 row) \* number of rows in table.

It is recommended to allocate between 12 and 16 CPU cores to ensure optimal system performance.

### 3.8.2.2 Agent Memory

### How to check current memory of agent?

Locate the agent.sh file in the AGENT HOME location, open file and check the value for USER\_MEM\_ARGS. Here, Xmx is maximum agent memory and Xms is minimum agent memory.

#### What should be the optimum memory for agent?

Optimum Memory for Agent = MAX ((compareFetchSize \* size of 1 row), (coosBatchSize \* size of 1 row)).

The resulting size is for 1 compare pair.

### How to modify the memory for agent?

Open agent.sh file and check the value for USER\_MEM\_ARGS. By default the value will be USER\_MEM\_ARGS="-Xmx1024M -Xms1024M".

To increase the memory, -Xmx value should be increased.



#### (i) Note

Any configuration change needs the Agent to be restarted.

### 3.8.2.3 Memory for Sorting and Sort Directory

#### How to calculate disk space and memory requirements for Sort Directory?

See Disk and Memory Requirements for the Server Component.

The maximum memory available to Oracle GoldenGate Veridata is set using jvm.memory.xmx in <VERIDATA\_HOME>/config/oggvdt\_cainput.properties.

When server-side sorting is enabled, a significant portion of this memory is used for sorting during comparisons.

The server.max\_sort\_memory parameter in the veridata.cfg file determines the maximum sorting memory. To avoid issues, ensure that the jvm.memory.xmx value is higher than the sorting memory requirements.

See #unique 97.

### 3.8.3 SSL Certificate Validation

An SSL (Secure Sockets Layer) connection is a secure and encrypted communication link between a client (such as a web browser or application) and a server. It ensures that data



transmitted between the two parties remains private and protected from eavesdropping, tampering, and forgery.

If SSL is enabled, then verify the validity of the certificate.

#### How to check if SSL is enabled?

For SSL between agent and server:

Check agent.properties file and see the value of the following properties: server.useSsl. If this is set to true, then SSL is enabled.

For SSL between agent and database:

Check db\_url in the agent.properties file.

If db\_url has SSL embedded in it then SSL is enabled.

#### How to check if SSL certificate is Valid?

Using browser

- 1. Visit the website in question.
- Click the padlock icon in the address bar.
- Select Certificate or Connection is secure (wording varies by browser).
- 4. View the certificate details, including:
  - a. Issuer
  - b. Validity period (start and expiration dates)
  - c. Subject name (who the certificate is issued to)

#### **Command line tools**

#### 1. Using openssl

Run the following command and examine the output for Certificate validity period and Issuer details::

openssl s client -connect <hostname>:443 -showcerts

2. Using curl:

curl -v https://<hostname> --insecure 2>&1 | grep 'start date\|expire date'

This shows the start and expiration dates of certificate.

### 3.8.4 Database Permissions

In the Oracle databases, during the COOS phase, if COOS join is enabled, then a temporary table is created and deleted when the session ends. Grant the user (specified during connection setup) the privilege to create private temporary tables (PTT), as they are useful in this phase.

How to check the user?

This can be checked from the **Connection Configuration** page.

- Click Connections and select a connection.
- 2. Click Connection Details, select Data Source, and then check the value of the User field.



### (i) Note

Ensure that the user has -> CREATE TABLE, CREATE TABLESPACE, PRIVATE TEMP TABLE, and INSERT/UPDATE/DELETE privileges.

# 3.8.5 Data for Comparison

Knowing what data to compare and what not to is one of the most crucial aspect of using Oracle GoldenGate Veridata the right way. This topic describes the following best practices to identify the data that should be compared.

- Specifying Primary Key (PK) Columns
- Partitioning
- Exclude Columns
- Delta Comparison

## 3.8.5.1 Specifying Primary Key (PK) Columns

For key columns, Oracle GoldenGate Veridata compares the data by value and non-key columns values are compared by hash. When a table lacks key columns, Oracle GoldenGate Veridata automatically treats all columns as key columns. This approach can significantly degrade performance because the comparison is done by value for every column in the table. Solution - Optimizing with Key Columns

Add at least one key column or a combination of multiple key columns such that the values remain unique.

By ensuring proper key column definitions in the table, Oracle GoldenGate Veridata can perform more efficient comparisons.

To add key columns:

- 1. In the Oracle GoldenGate Veridata UI, go to Group and Compare Pairs.
- Select Group and then click Compare Pair.
- Click the Column Mapping tab.
- 4. Set Key Mapping Method to User Defined.
- 5. Set one or more columns which can be set as keys during the operation.
- 6. Click Save.

# 3.8.5.2 Partitioning

Working with large tables might impact Veridata performance so you can divide large source and target tables into smaller dataset.

It can be done using the following: Auto Partitions, Manual Row Partitions, or Table Partitioning.

#### **Automatic Row Partitions**

This option is available only for Oracle DB which can split the compare pair (tables) into the multiple parts based on the values you provide. Automatic Row Partitions can be configured while creating a compare pair. This speeds up the comparison as it will run on smaller dataset.



To configure Automatic Row Partitions:

- 1. In the Oracle GoldenGate Veridata UI, go to Group and Compare Pairs.
- 2. Click Create, click Mapping Rules, and then Mapping.
- 3. Click **Automatic Row Partition** and toggle the switch to change partition numbers.

#### **Manual Row Partitioning**

Manual Row Partitioning can be useful for manually distributing the workload across multiple processes or restricting the comparison to a specific subset of data, such as the last n days.

This option is available for both Oracle as well as non-Oracle databases. This can be achieved by applying a SQL Predicate and filtering the records.

To configure Manual Row Partitions:

- 1. In the Compare Pair page, click Row Partitioning and enable Source/Target Partition.
- 2. Click + sign and enter the SQL predicate to partition the table.

#### **Table Partitions**

This option is available only for Oracle DB. If a table is created with partitions then all the partitions are listed on the page. You can select only required partitions during creation of Compare Pair.

To configure Table Partitions:

- 1. In the Oracle GoldenGate Veridata UI, go to Group and Compare Pairs.
- Click Create, click Mapping Rules, and then select the Include Table Partitions check box.

## 3.8.5.3 Exclude Columns

You can exclude columns from comparison to redice the processing load in either of the following conditions:

- If a table contains columns that will never change.
   Or
- Irrespective of the columns being in sync or not.

The columns can be excluded when you create or edit the compare pairs.

To exclude columns:

- 1. In the Oracle GoldenGate Veridata UI, go to **Group and Compare Pairs**.
- 2. Select Group and then click Compare Pair.
- 3. Click the Column Mapping tab.
- 4. Click Remove mapping for the columns you want to exclude.
- Click Save.

## 3.8.5.4 Delta Comparison

Compare pairs can be configured to use delta processing, a performance feature that compares only the data blocks that have changed since the last run, instead of scanning the entire table.



Delta Comparison prevents comparing all the historical data again and again and compares only changed (delta) data.

To enable Delta Comparison:

- 1. In the Oracle GoldenGate Veridata UI, go to Group and Compare Pairs.
- Select Group and then select a compare pair from Existing compare pair list.
- 3. Click Delta Processing.
- 4. In the **Delta Processing** tab, toggle **Delta Processing**.
- Select Delta Columns.
- 6. Click Save.
- Delta Column Selection
- ROWDEPENDENCIES

### 3.8.5.4.1 Delta Column Selection

You can use a column with the following properties as a delta column:

- A column that increments on every DML operation (DML Data Manipulation Language.
   These operations allow you to insert, update, delete, or retrieve records from tables).
- A column with either of the following data types: number or timestamp.
- This column should not be primary key as primary key does not change on every DML operation. (It remains same for update operations).

If you cannot find such columns, then for an Oracle DB, the default delta column is ORA\_ROWSCN. This works best when ROWDEPENDENCIES are enabled for table. For DBs other than Oracle, there is no default delta column. Therefore, select a column with the properties listed in this topic.

### 3.8.5.4.2 ROWDEPENDENCIES

ROWDEPENDENCIES are enabled when creating database tables.

How to check if ROWDEPENDENCIES are enabled for Oracle DB?

This should be checked at DB level with the following query:

 $SELECT\ owner,\ table\_name,\ dependencies\ FROM\ dba\_tables;$ 

This will return ENABLED or DISABLED for each table. If you don't have access to dba\_tables, query all\_tables instead.

You cannot change this after the table has been created, therefore re create the table to set it on.

What is SCN and how to change its value?

See ROWSCN.

### Why ORA\_ROWSCN works best when ROWDEPENDENCIES are enabled for the table?

The SCN (System Change Number) plays a critical role in Veridata's comparison process. Here's how it impacts the comparison workflow and why enabling row dependencies is important:



SCN Behaviour with Row Dependencies Enabled:

When row dependencies are enabled, an SCN is associated with each individual row in the database.

This ensures that only the rows that have undergone changes between the source and target are flagged for comparison.

2. SCN Behaviour with Row Dependencies Disabled:

If row dependencies are not enabled, then the SCN is associated with data blocks instead of individual rows.

3. Impact on Initial Comparison:

Divergence in data blocks can result in numerous false positives during the initial compare phase. The compare-out-of-sync (COOS) step will unnecessarily include additional rows for comparison, even if the data within those rows has not changed. So more data are oos.

4. Why Enable Row Dependencies?

Enabling row dependencies reduces unnecessary comparisons, improving the accuracy and efficiency of the process.

Without row dependencies, the system allows rows in a diverged block to be compared, potentially increasing the workload and causing delays.

# What happens when ORA\_ROWSCN is used as delta column but ROWDEPENDENCIES are not enabled?

Block level comparison are used. Block-level comparison in databases refers to the process of comparing data stored in blocks or pages rather than at the individual record or row level. In this case, delta compare will be slower. It will still be faster than comparison without delta.

# 3.8.6 Veridata Configurations

- Connection Configurations
- Profile Configurations
- Server Configurations
- Agent Configurations

# 3.8.6.1 Connection Configurations

#### **Compare Fetch Size**

This configuration is useful in determining the batch size for fetching the data from database. Default value is 1000.

To speed up the compare process, the number of rows fetched from database can be increased in the connection settings by changing Compare Fetch Size. Start with 10,000 and increase up to 100,000.

To update Compare Fetch Size, from the **Connections** tab, open the connection that is used in the compare job and change the value of the **Initial Compare Fetch Batch Size**. This speeds up the fetch process during the initial comparison.

# 3.8.6.2 Profile Configurations

- Sorting Configuration
- Initial Compare Configuration
- Out-of-Sync Configuration



Repair Configuration

## 3.8.6.2.1 Sorting Configuration

Oracle recommends you to set the value of **Sort Data Using** to **Server**, as sorting done at the Oracle GoldenGate Veridata server will be faster than sorting at the Database level.

To set this sorting method:

- 1. In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- Click the Sorting Method tab and select Sort Data Using and select a value from the drop-down list.

## 3.8.6.2.2 Initial Compare Configuration

#### **Max Concurrent Comparison Threads**

This is used to run multiple compares in parallel.

When running multiple compare pairs in parallel, exceeding the machine's capacity can exhaust memory, leading to slower processing. This can be addressed by configuring the Max Concurrent Comparison Threads setting. Default value is 4.

### What should be the value of Max Concurrent Comparison Threads?

Value of Max Concurrent Comparison Threads depends on the value of number of CPU cores in Veridata server node. Start with 30-40% of the CPU core and it can be increased up to the number of cores.

#### To change the value of Max Concurrent Comparison Threads

- 1. In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- 2. Click the **Initial Compare** tab and select **Max Concurrent Comparison Threads**. Maximum value for this field is 100 and minimum value is 1.

#### **Source and Target Optimizer Hint**

In Oracle, the optimizer hint is used to enable parallel execution for a specific table or query operation. An optimizer hint can be used to increase the processing of the queries.

For example, PARALLEL(x,16) tells Oracle to execute operations on the table x using 16 parallel execution threads (or degrees of parallelism - DOP).

To set the Optimizer hint:

- 1. In the Oracle GoldenGate Veridata UI, click Profiles and select a profile.
- Click the Initial Compare tab and select Source Oracle Optimizer Hint and Target Oracle Optimizer Hint.



Parallel running is beneficial for large tables but can increase system resource usage.



## 3.8.6.2.3 Out-of-Sync Configuration

#### Confirm-Out-Of-Sync(COOS) Batch Size

This setting enables fetching data in batches instead of fetching individual rows and making lot of round trips to the database.

To update coos batch size

- In the Oracle GoldenGate Veridata UI, click Profiles and select a profile.
- Click the Confirm-Out-Of-Sync tab and select Confirm-Out-Of-Sync Batch Size and select a value from the drop-down list.



#### Note

The default value is 1000. Maximum value for this field is 100,000 and minimum value is 1. Start with 10,000 and increase up to 100,000.

### **Source and Target Optimizer Hint**

To set the Optimizer hint:

- In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- Click the Confirm-Out-Of-Sync tab and select Source Oracle Optimizer Hint and Target **Oracle Optimizer Hint.**

## 3.8.6.2.4 Repair Configuration

## Repair Batch Size

This setting enables repairing data in batches instead of individual rows and making lot of round trips to the database.

This can be set from profiles. The default value is 1000. Maximum value for this field is 100000 and the minimum value is 1.

To set the Repair Batch Size:

- In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- Click the **Repair** tab and select **Repair Batch Size**.

### **Number of Concurrent Repair Operations**

To run repair concurrently, you can use this setting. The default value for this is 1. However, this can be increased to perform a repair in parallel.

To set the Number of Concurrent Repair Operations:

- In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- Click the Repair tab and select Number of Concurrent Repair Operations.

# 3.8.6.3 Server Configurations

The veridata.cfg file is located in the <VERIDATA\_HOME>/config/veridata folder. It stores many configurations that are needed to run and customise Veridata operations.



#### COOS join:

This is enabled by default for table with no unique or primary key and the behaviour can be overridden from the veridata.cfg file in the <*VERIDATA\_HOME*>/config/veridata folder.

With this configuration, the database queries will use join and will be a lot faster compared to queries without join.

You need to add/edit the following properties:

- coos.join.strategy = Values nokey, always, and never
- nokey This is the default value of the field and it is useful on the tables that do not have primary key/indexes defined.
- always Always use COOS join.
- never Don't use COOS Join.
- server.concurrent.writers: The number of writer threads per sort directory.
- server.concurrent.readers: The number of reader threads for entire server.
- server.number\_sort\_threads: The number of threads used to sort input buffers from the agent. Should not be larger than number of available processes.

## 3.8.6.4 Agent Configurations

- Coos Batch Fetch
- ROWSCN

### 3.8.6.4.1 Coos Batch Fetch

Using COOS batch, Oracle GoldenGate Veridata can be leveraged to perform fetch on the agents as batches instead of running individual queries. This in turn saves a lot of database trips and hence will be faster.

#### Adding coos batch fetch in agent.properties

To enable this, add the following property to the agent.properties file for both the agents and restart the agents:coos.batch.fetch=true



Do not use COOS join and COOS batch at the same time. Use either of them at a time.

### 3.8.6.4.2 ROWSCN

This step is useful when customer wants to skip full table comparison and want to compare data after a particular SCN value.

### What is SCN?

In a database, a System Change Number (SCN) is a logical, internal, and monotonically increasing number used to track changes made to the database. Each committed transaction in the database is assigned a unique SCN that represents the point in time when the transaction was committed.



If the current SCN value is 100, then designate the rowscn value to 100, then the comparison omits those records with that specific SCN value. This property can be added in agent.properties. For more information, see Frequently Asked Questions (FAQ).

rowscn=10000

After adding this for both the agents, restart both agents and run the comparison again.

# 3.8.7 Frequently Asked Questions (FAQ)

#### What is a sort directory?

In the sorting phase, chunks of data small enough to fit in main memory are read, sorted, and written out to a temporary file in a directory.

### How to find a sort directory?

This can be found from profile setting page under **Sorting Method**.

- 1. Go to Profiles, Select Profile, and click Sorting method
- Select Temporary Storage Directory for Source Data and click Temporary Storage Directory for Target Data.

### What is Agent Properties file location?

This can be found in <agent\_home> path with name agent.properties.

#### Where is Veridata server config file present?

<VERIDATA\_HOME>/config/veridata/veridata.cfg

#### Where is oggvdt\_cainput.properties file present?

< VERIDATA\_HOME > /config/oggvdt\_cainput.properties

#### Where to find the Reports?

Reports are present in path: <VERIDATA\_HOME>/veridata/reports

#### Can we have multiple agents for 1 DB?

Yes. But we recommend 1 agent for 1 DB.

### How many Compare pairs I can create in a group?

No restriction on this. Any number is fine.

### How many Compare pairs I can run in parallel?

100 is the max default number. But it can be updated from <code>veridata.cfg</code> file by changing the <code>value</code> of <code>max\_concurrent\_jobs</code>.

# **Get Started**

- Accessing Oracle GoldenGate Veridata Web User Interface
- Home Page

# 4.1 Accessing Oracle GoldenGate Veridata Web User Interface

To protect data and comparison configurations, Oracle GoldenGate Veridata has security roles. Before attempting to use the Oracle GoldenGate Veridata Web User Interface, you should confirm which role has been granted to you by the Oracle GoldenGate Veridata Administrator. For more information, see Configuring User Groups.

To connect to the Veridata Web User Interface, open a web browser and type the following address. For example, http://hostname/ipaddress:port/veridata or https://hostname/ipaddress:port/veridata (https in case SSL / TLS Security for Web is selected while installation).

#### Where:

*hostname* is the name of the system where Oracle GoldenGate Veridata Server is installed and *port* is the port number where it is running.



Oracle GoldenGate Veridata 23c is designed to run optimally at HD resolution 1920x1080 and minimally  $1342 \times 672$ .

# 4.2 Home Page

The Oracle GoldenGate Veridata Home page comprises the following:

• Left Navigation Pane: Menu to all of the following Oracle GoldenGate Veridata objects:

#### (i) Note

Users with less-privileged role may see only a subset of these objects. For more information, see <u>Configuring User Groups</u>.

- Connections
- Groups and Compare Pairs
- Jobs
- Monitor Jobs
- Running Jobs
- Profiles



- Users
- Utilities
- **Getting Started**: Comprises information about all the Oracle GoldenGate Veridata Objects and the tasks you can perform on the respective pages.
- Dashboard: The Dashboard area lists the following:
  - Compare Pair Status of Running Jobs
  - Compare Pair Status of Completed Jobs
  - Recently Completed Jobs
  - Favorites
- Quick Guide: Lists details of Oracle GoldenGate Veridata documentation.

The Home page also comprises the following:

- About
- Help
- User Menu

## 4.2.1 About

Click the **About** icon to get information about the Oracle GoldenGate Veridata version installed.

# 4.2.2 Help

Click the **Help** button to access the Oracle GoldenGate Veridata documentation.

## 4.2.3 User Menu

#### My Account

To access the My Account page, click User Menu, and then select My Account.

On this page you can:

- View the currently-logged-in user information, the Group that user belongs to.
- Edit user account details, such as Description, Email ID, and Reset Password.

#### **Preferences**

To access the Preferences page, click User Menu, and then select Preferences.

On this page you can:

 Auto Refresh Interval: Set the time for refreshing any of the Monitoring pages, such as Completed Jobs, Running Jobs, or Repair Jobs.

For example, if you (logged-in user) set the **Auto Refresh Interval** to 40, these pages will automatically refresh after 40 seconds.

The default value is set to 30 seconds.

The interval range has to be between 10 seconds and 100 seconds.

Default view for Monitor Pages: You can set either Completed Jobs or Running Jobs
as the default page to display when you open the Monitor Jobs page on the navigation
menu. The default is set to the Completed Jobs page.



To sign out off Oracle GoldenGate Veridata, click User Menu, and then select Sign Out.

# Manage

- Connections
- Groups and Compare Pairs
- Profiles
- Jobs
- <u>Users</u>
- Utilities

# 5.1 Connections

To get started with Oracle GoldenGate Veridata, you must define a connection to the source and target databases that contain the data that you want to compare. Oracle GoldenGate Veridata Server uses the connection information to communicate with Oracle GoldenGate Veridata Agent.

A connection is defined by:

- A host where Oracle GoldenGate Veridata Agent is running.
- The port number for Oracle GoldenGate Veridata Agent on that host.
- The datasource that is associated with this agent.

Connections are managed from the **Connections** page in the navigation pane.

All connections that exist within the Oracle GoldenGate Veridata repository are shown in the list on this page.

#### (i) Note

Only users with Adminstrator and Super User roles can perform the following tasks:

- Create Connections
- Edit Connections
- Delete Connections
- Creating a Connection
- Editing a Connection
- Deleting a Connection

# 5.1.1 Creating a Connection

To create a Connection:

1. Click **Connections** in the navigation pane.



On the **Connections** page that opens, enter the following parameters:

- 2. Enter a Name and Description for the new connection and click Next.
- 3. Enter the following parameters on the **Agent Connection** page:
  - Host Name: The DNS (Domain Name Server) host name or IP address of the host on which the Oracle GoldenGate Veridata Agent is installed.
  - Port: The port number that is assigned to the agent (or the Manager process, if a C-agent). To find out the port number of a Java agent, view the server.port parameter in the agent.properties file within the agent's installation directory. To find out the port number for a C-agent Manager, run the GGSCI program from the agent's installation directory, and then use the INFO MANAGER command.
  - Database: This is an optional field. You can click on Verify without selecting the datasource type. The correct datasource is identified and displayed after the verification is done:
    - Apache Hive
    - HPE NonStop
    - IBM Db2
    - MariaDB
    - Microsoft SQL Server
    - MySQL
    - Oracle Database
    - PostgreSQL
    - Snowflake
    - Sybase Adaptive Server Enterprise
    - Teradata Vantage
  - Use SSL for communication: Select the Use SSL for communication checkbox for secure communication between the Veridata agent and the server.
    - This option is is enabled only of you have set the <code>server.useSsl</code> parameter as true, while configuring SSL communication. See <a href="Parameters for Configuring SSL Communication">Parameters for Configuring SSL Communication</a>.
  - A user name and password for connecting to the datasource (if required by the database). Database users need certain DB priviliges. See <a href="Database Privileges for the Agent Component">Database Privileges for the Agent Component</a>, for more information.
  - (Optional) A separate user can be configured for executing repair operations at the target database. This user needs permission to update as well as to query the tables.
- 4. Click Test Connection to confirm that the connection is. If you have selected the Use SSL for communication checkbox in the previous screen, then SSL will be used for verifying the datasource connection.

# 5.1.2 Editing a Connection

- 1. Click **Connections** in the navigation pane.
- In the connections list, click the edit (pencil) icon next to the connection that you want to edit.

Alternatively,



Click the connection name from the connections list, to display the **Connection Details** and **Connection Configuration** sections.

To edit the Connection Details:

- Click Edit in the Connection Details section, to edit the description, Agent or Datasource connection details.
- 2. Click **Save** to save your edits.

To edit the Connection Configuration:

This tab modifies the behavior of the agent.

A check mark under **Use Default** indicates that a parameter is set to the default value.

The current setting is shown under Value.

If the parameter is a toggle, a check mark under **Value** indicates that it is enabled.

To change a setting

- Clear the Use Default check box.
- 2. Edit Value.
- Click Save.

#### **Parameter Details**

- Agent Message Timeout: Specifies a time interval, in seconds, after which the Server abends, if it has not received a message from the Agent.
- Truncate Trailing Spaces When Comparing Values: Space (U+0020) and Ideographic Space (U+3000) are the truncate targets. This parameter works only for String or Binary columns. You can configure the truncated length with the truncate\_spaces\_len entry in the veridata.cfg file of the Oracle GoldenGate Veridata Server. This parameter also truncates the column padding character if the padding character is one of the targe spaces. This parameter does not trim trailing spaces on LOB data.
- **Compare Fetch Size**: Sets the number of rows that are fetched at once for the initial comparison and for the confirm out of sync using batch or join. Increasing the batch size may increase throughput, as compared to standard database access.

Exclusively, for the Oracle and PostgreSQL databases, the default value is 1000 and it fetches a batch size of 1000 rows; the minimum value that you can enter is 100. Any lesser value that you enter automatically changes to 100. The maximum value is 100000. For databases other than Oracle and PostgreSQL (SQL Server, Sybase, or DB2), the default value is 0, minimum is 0, and maximum value is 1000.

# 5.1.3 Deleting a Connection

- 1. To delete a connection, click the delete (trach can) icon.
- 2. Click **Delete** to confirm the delete action.

Alternatively,

- Click on a connection in the connections list, click Delete in the Connection Details section.
- 2. Click **Proceed** to confirm the delete action.





Before deleting a connection, you must unlink it from any groups and jobs to which it is linked, or delete the group or job, if appropriate.

# 5.2 Groups and Compare Pairs

- Group Details
- Compare Pairs

# 5.2.1 Group Details

**Groups** are logical containers for one or more Compare Pairs. They help you to organize and partition large or diverse sets of data into more manageable units. Groups are linked to jobs when jobs are created. Any group can be linked to one or more jobs, allowing you complete control over how and when data is compared.

### (i) Note

A Group is associated with a set of connections to the source and target data. Before creating a group, you must create these connections.

The **Group Details** section of the **Groups and Compare Pairs** page lists the following details of the selected Group. You can also see the list of Compare Pairs associated to that Group.

- Name
- Description
- Source Connection
- Target Connection

You can modify the **Source Connection** and the **Target Connection** of the Group and click **Save** to save to Group details.

- Creating Groups and Compare Pairs
- Cloning a Group
- Deleting a Group

# 5.2.1.1 Creating Groups and Compare Pairs

To create Groups and Compare Pairs:

- Click Groups and Compare Pairs, and then click Create to display the Groups Details page.
- 2. Enter the following details for the Group name and description Name and Description.
- Select Source Connection and Target Connection for the Group. See Connections.
  - Source Catalog: The source metadata catalog or database.
  - Source Schema: The owner of the source database objects that are to be compared.
  - Target Catalog: The target metadata catalog or database.



- Target Schema: The owner of the target database objects that are to be compared.
- 4. Click Next to display the <u>Mapping Rules</u> section. By default, Mapping Rules is toggled on. Mapping Rules help map source and target tables automatically based on rules set. If you do not want to apply Mapping rules, then you can turn off this switch. Let's consider the Mapping Rules switch is toggled off.
- In the Mapping Rules section, the following options are displayed All, Mapped, and Unmapped.
  - All: Lists all tables or files that are contained by the specified datasources. The Source tables are already listed. You can select the Target table from the Target drop-down list for Column Mapping.
  - Mapped: Shows only previously mapped tables or files.
  - UnMapped: Shows only tables or files not mapped in any other mapping.
- Mapping Rules
- Mapping
- Preview and Generate Compare Pairs

## 5.2.1.1.1 Mapping Rules

To set the Mapping Rules for the Groups and Compare Pairs:

- In the Mapping Rules section, select one of the following methods for matching source table names to target table names. By default, Mapping Rules is toggled.
  - Using Exact Names
    - This pattern matches names character-for-character, so ensure that each source and target name are identical. As an example, this mapping rule is useful for comparing production and failover databases. By, default the **Using Exact Names** option is selected. Select **Using Exact Names** if you want to map the tables defined at database. For example, if there are 10 partitions in Source Table matching 10 table partitions in Target Table with same table names, then 10 compare pairs are created. If no table exists then, one compare-pair per table gets generated. After all partitions are mapped, details of compare-pairs generated are displayed in Preview Tab.
  - Using Wildcard Pattern. (Enclose within single % on either side)
    To use this method, supply a wildcard string in Source Pattern and Target Pattern
    Like, and select either Like or Not Like options (for both Source Pattern as well as
    Target Pattern) that include the percent symbol (%) as the wildcard. A % in the target
    matches the text that is matched by the % in the source. By default, the Like option is
    selected for both the source and target tables. You can select Not Like if needed
    during the compare-pair generation.
    - Example:

Assume source tables of:

SOURCE\_TABLE\_1 SOURCE\_TABLE\_2 MY\_SOURCE\_TABLE\_1 MY\_SOURCE\_TABLE\_2 DUMMY\_TABLE

Assume target tables of:

TARGET\_TABLE\_1 TARGET\_TABLE\_2 MY\_TARGET\_TABLE\_1



MY\_TARGET\_TABLE\_2 DUMMY\_TABLE MY\_DUMMY\_TABLE

Some possible **Like** pattern matches are:

Table 5-1 Examples for Mapping Rules Using Wild Card Option % - Like option selected

Source Pattern	Target Pattern	Matches
%SOURCE_%	%TARGET_%	SOURCE_TABLE1=TARGET_ TABLE1, SOURCE_TABLE2=TARGET_ TABLE2
%MY_SOURCE_%	%MY_TARGET_%	MY_SOURCE_TABLE1=MY_T ARGET_TABLE1, MY_SOURCE_TABLE2=MYT ARGET_TABLE2
%SOURCE_%	%MY_TARGET_%	Table1=Table2 TABLE2=TABLE2
%DUMMY_TABLE%	%_TABLE%	DUMMY_TABLE=DUMMY_T ABLE
%DUMMY_TABLE%	TARGET_%_	None

Table 5-2 Examples for Mapping Rules Using Wild Card Option % - Not Like option selected

Source Pattern	Target Pattern	Matches
%SOURCE_%	%TARGET_%	DUMMY_TABLE=DUMMY_T ABLE

- The wildcard resolution is case-sensitive.
- Include Views: The Include Views check box is selected by default. In the Mapping page, you can select the compare pairs based on the Mapping rules. This also has the View Table pair if you include views in the mapping rules page. Mapping rules are ways to ease mapping or to automatically create mapping between source and target database based on selected rules. If you do not want to include Views, then you can deselect the check box.
- Include Table Partitions: The Include Table Partitions check box is selected by default. This option allows you to include all the table partitions (when applicable to the database) while generating compare pairs. If you do not want to include table partitions, then you can deselect the check box.
- 2. After you have selected the Mapping Rules, click **Next** to display the **Mapping** section.

## 5.2.1.1.2 Mapping

By default, the **Mapping** section shows all of the source and target objects that are contained by the specified datasources. You can filter the list at any time in the process of creating compare pairs. For example, after you finish mapping several compare pairs, it might be helpful to reduce the size of the list by using the filter to show only unmapped objects. This saves you navigation time.

To filter the list of mappings, click one of the following options:



- All: Lists all tables or files that are contained by the specified datasources. To view all the Source and Target tales, select the All option. However, ensure to give this Compare Pair a different name.
- Mapped: Lists only previously mapped tables or files.
- **Unmapped**: Shows only tables or files not mapped in any other compare pair.

You can also enter a search string in the Search box to filter the mappings list.

#### To map tables:

- Select the Source table. In case of the All option selected, all the Source and Target tables are selected by default. You may want to deselect the tables that you do not want to be mapped.
- 2. From the **Target** drop-down list, select the target table.
- (Optional) You can also toggle <u>Automatic Row Partition</u> to enable partitioning of multiple rows.
- 4. After the mapping is complete, click **Next** to go to the **Preview** section and review the configurations of the **New Compare Pairs**. The following details are displayed:
  - Compare Pair Name
  - Source Table
  - Target Table
  - Profile
- Automatic Row Partition

If you want to split the large table comparison into multiple partitions, then you can use Automatic Row Partition. The partitioning is possible only when both the source database and target database are Oracle.

### 5.2.1.1.2.1 Automatic Row Partition

If you want to split the large table comparison into multiple partitions, then you can use Automatic Row Partition. The partitioning is possible only when both the source database and target database are Oracle.

The automatic row partition creates configurable partitions and generates compare pairs for each generated partition.

To configure Automatic Row Partition:

- 1. On the Mapping section of the Create Group and Compare Pair (or Edit Group and Compare Pair) page, select the Source and Target tables.
- Toggle Automatic Row Partition against the selected table pair. By default, the toggle option is disabled.
- 3. Select a numeric value between **2 to 100** in the text field before generating compare pair. The minimum value you can select is 2 and the maximum is 100. This value decides the number of compare pairs to be created. Each auto generated compare pair compares the subset of data from the selected source and target table.
- **4.** After the required datasources are mapped, details of compare pairs generated are displayed in **Preview** section.



## 5.2.1.1.3 Preview and Generate Compare Pairs

You can view compare pairs that can be generated by using the mapping in the **Preview** section. You can select the compare pairs in the Preview page. Based on the selection, you can generate the compare pairs.

- 1. If you want to modify any of the mappings, you can click **Back** to go back to the **Mapping Rules** section, modify the details. For example, from the currently selected **Using Exact Names**, if you want to switch to **Using Wildcard Pattern** and enter emp% in the **Source Pattern** and the **Target Pattern**, and select **Like**, and then click **Next** to display the **Retain Compare Pairs** dialog box. Click **Yes** if you want to retain the previously-selected compare pairs. If you click **No** and proceed further, the previously-selected compare pairs are not retained and Compare Pairs of the Employee tables are displayed in the **Preview** section.
- 2. After reviewing the Compare Pair details, click Generate Compare Pairs.

The compare pairs are generated and are listed in a tabular format. See **Compare Pairs**.

## 5.2.1.2 Cloning a Group

You can clone a Group from the **Groups and Compare Pairs** page.

- To clone a Group, select a Group from the list, and click the 3 dots (...) and select Clone
  to display the Enter Group details dialog box.
- 2. Enter the Group Name and Description and click Clone.

A new Group is created with the same Compare Pair details of the Group you just cloned.

# 5.2.1.3 Deleting a Group

You can delete a Group from the **Groups and Compare Pairs** page.

- 1. To delete a Group, select a **Group** from the list, and click the 3 dots (...) and select **Delete**.
- 2. Click **Delete** in the confirmation dialog box to confirm deletion of the selecte **Group**.

A Group deleted successfully message is displayed after the Group is deleted.

# 5.2.2 Compare Pairs

A compare pair is the logical relationship between a source table or file and a target table or file for the purpose of comparing their data. Compare pairs are linked to groups. As a reason, all of the source and target objects that you configure into compare pairs for any given group must be accessible from the datasource connections that are associated with that group.

The **Groups and Compare Pairs** page lists the following details of the Compare Pairs created for that particular Group.

- Enabled: By default, all the compare pairs are enabled. You can toggle this switch to
  disable the selected Compare Pair. You can select the compare pairs to be included in the
  Group by toggling the Enabled switch, and then click Save to save the Group and
  Compare Pairs details.
- Compare Pair Name: The default name format is <source>=<target>. If you hover the mouse cursor over a Compare Pair name, the actual source and target object names are displayed.
- Source Table: The source table in the Compare Pair.



- **Target Table**: The target table in the Compare Pair.
- Delta Processing: See Delta Processing
- Validation Status: By default, the Validation Status is Never Validated. See Validate Column Mapping. After you select the Compare Pair and click Validate Column Mapping, the Validatation Status of the selected Compare Pair gets updated to Validated. You can also select multiple Compare Pairs and click Validate Column Mapping to validate their status.
- Actions: You can delete the selected compare pairs by clicking the Delete icon. See Deleting a Compare Pair

Use the Compare Pair section on the Groups and Compare Pairs page to do the following:

- Add Compare Pairs
- Validate Column Mapping
- **Enable Delta Processing**
- **Delete Compare Pairs**
- **Compare Pair Details**

## 5.2.2.1 Compare Pair Details

The Compare Pair Pair Details section of the Groups and Compare Pairs page lists the following details of the selected Compare Pair. Click a Compare Pair to view the following Compare Pair details:

- Source Table: The source table or database.
- **Source Catalog**: The catalog of the source database objects that are to be compared.
- **Source Schema**: The owner of the source database objects that are to be compared.
- Compare Pair Name: This is either the default name format of <source>=<target> or a userdefined name. In either case, if you hover the mouse cursor over a Compare Pair name, the actual source and target object names are displayed.
- **Profile**: If a run profile exists for a compare pair, it is shown here. Otherwise, this field is blank and the default profile will be used during comparisons.



#### (i) Note

If you select a profile with comparison type set to Row Count, the delta feature is not applicable.

- **Target Table**: The target table or database.
- **Target Catalog:** The catalog of the target database objects that are to be compared.
- **Target Schema**: The owner of the target database objects that are to be compared.
- Comparison Type: Displays the opted comparison type of the profile that you have selected.

Supported Values: Full, Row Count.

Full: stands for regular Veridata (data) comparison of a table.

Row Count: Only row count comparison for a table, without any actual data comparison.



 Validation Status: Shows whether or not the columns of the source and target objects are suitable for being compared, based on the results of any previous validation that was performed.

You can modify the Source Table, Source Catalog, Source Schema, Target Table, Target Catalog, Target Schema and update the Profile of the Compare Pair, and then click Save to save to Compare Pair details.

Click **Full Screen** to view the view details of the Compare Pair on full screen. To go back to the default view, click **Partial Screen**.

Click Back to Compare Pairs List to go back to the Compare Pair section of the Groups and Compare Pairs page.

For a selected Compare Pair, you can also perform the following from the Compare Pair Details section:

- Column Mappings
- Delta Processing
- Row Partitions
- Deleting a Compare Pair

# 5.2.2.1.1 Column Mappings

Use the **Column Mapping** tab to create or change a column mapping.

You must have the Administrator or Super User role to create or edit column mappings.

The Column Mapping tab lists the following Compare Pair details:

- **All**: Lists all tables or files that are contained by the specified datasources. Ensure to give this compare pair a different name.
- Mapped: Lists only previously-mapped columns.
- Unmapped: Lists only columns not mapped with any other column of the compare pair.
- Source: Details of the source table, such as Include, Name, and Data Type.
- Target: Details of the target table, such as Name, Data Type, Format, and Key.
- Validate Column Mapping
- Key Mapping Method

## 5.2.2.1.1.1 Validate Column Mapping

Validation is a preliminary test to determine whether or not the source and target table structures are compatible, and that they both have primary or unique key columns that match. You can perform a manual validation at any time. Oracle GoldenGate Veridata always performs a validation at runtime.

To validate column mapping, on the **Groups and Compare Pairs** page > **Compare Pairs** section, select a Compare Pair, click **More Actions**, and then select **Validate Column Mapping**. By default, the Validation Status is **Never Validated**.

The selected Compare Pair is validated and the following message is displayed: **Compare Pairs Validated Successfully**.



### 5.2.2.1.1.2 Key Mapping Method

There are two methods for mapping key columns and comparison columns:

- System Generated: Column mappings are configured automatically by Oracle GoldenGate Veridata at runtime based on current object metadata.
- User Defined: Column mappings are configured manually by an Oracle GoldenGate Veridata user who has the Administrator or Super User role.

All new compare pairs default to System Generated for key columns and for comparison columns. You can change to a different mapping method at any time.

### How these methods apply to keys

- System Generated: If you know that the objects in a compare pair both contain a primary key or a unique index, you can leave the key mapping method set to the default of System Generated. The key columns will be mapped automatically. To map keys, Oracle GoldenGate Veridata finds all indexes on the source and target objects and tries to find a primary key on each one. If primary keys are not found, Oracle GoldenGate Veridata tries to use the smallest index (least number of columns), and then it maps the columns that have identical names and comparison formats. Any columns that cannot be matched are excluded from the configuration.
- User Defined: If an object has neither a primary key or unique key, you can use the User
  Defined method to map key columns manually, one by one. You can also use the userdefined method to override existing keys or indexes, but the columns that you select to use
  as a key must ensure the uniqueness of rows. Also avoid using source and target indexes
  that have different precision levels or other characteristics that can reduce the accuracy of
  row selection, especially in a heterogeneous environment.

### How these methods apply to comparison columns

- System Generated: If the source and target comparison columns have the same names and comparison formats, you can leave the comparison column mapping method set to the default of System Generated. Oracle GoldenGate Veridata will map those columns automatically at runtime. Non-matching columns are excluded from the configuration. By default, this method includes all of the columns in a comparison. This method defaults to the hash comparison method. You can change the comparison method later by editing the compare pair.
- **User Defined**: Use this method to map source and target columns manually and to control the comparison method.

You can combine these methods to speed up the mapping process. If most of the column names support system-generated mapping, you can use it and then switch to the user-defined method to map the remaining columns, or to exclude columns from the comparison. For example, you can exclude columns if you know that their values never change or if you expect their values to be out-of-sync.

## 5.2.2.1.2 Delta Processing

This topic provides answers to typical questions about the delta processing feature that is available for all Oracle GoldenGate Veridata supported databases.

In Oracle GoldenGate Veridata, the source and target tables are configured using compare pairs, which are grouped and added to a job to run the comparison.



During the subsequent runs of a comparison job, the comparison of the tables can be performed based on what has changed in the tables from the previous job run; these jobs are Delta Processing Jobs.

- For all supported databases, you can use the delta processing performance feature if you are using server-side sorting.
- For the NonStop platform, Oracle GoldenGate Veridata finds a changed block by detecting a change in its Volume Sequence Number (VSN) since the time of the last comparison.
- For all platforms, the delta comparison column must contain a value that is modified every time the row is modified and this value must always keep increasing. For example, TIMESTAMP or NUMBER. For the Oracle database, ROW\_SCN is a default delta column, which keeps on increasing at every delta run.
- The comparison of the tables can be performed based on what has changed during the subsequent runs of a comparison job.

With delta processing enabled, Oracle GoldenGate Veridata compares the difference between the 2 consecutive jobs.

To enable Delta Processing:

- 1. In the Groups and Compare Pairs page, click the Delta Processing tab under the Compare Pairs Details section.
- Ensure that the column names are populated in the Use Source Delta Column and Use Target Delta Column. Note that you cannot edit the source and target queries.
- From the Edit Group Compare Pair page also, you can enable delta processing by selecting the particular Compare Pair and by selecting the dropdown option to enable delta processing.
- What is Delta Processing?

### 5.2.2.1.2.1 What is Delta Processing?

#### How does it work on NonStop Platforms?

Oracle GoldenGate Veridata finds a changed block by detecting a change in its Volume Sequence Number (VSN) since the time of the last comparison. The VSN is a disk-specific change number that increments sequentially with each database operation that is performed on the data. Each time that a row changes, there is a change in the VSN of the disk block where the row resides.

There is no relationship between a VSN in a file on one disk and a VSN on another. Oracle GoldenGate Veridata tracks VSNs on a per-partition basis on the source and target disks and maintains its own correlations to perform accurate delta comparisons. Once you enable delta processing, it is used for all subsequent runs until you disable it again.

#### Note

The first run of a compare pair always compares all of the rows in the source and target objects to establish an initial VSN state from which to evaluate deltas in future runs.



#### How does it work on all other Platforms?

Oracle GoldenGate Veridata compares a source database table to the target database table. The source and target tables are configured using compare pairs, which are grouped and added to a job to run the comparison (see <a href="Compare Pairs">Compare Pairs</a>). When all the rows in the table are compared, it is a Full Comparison Job.

During the subsequent runs of a comparison job, the comparison of the tables can be performed based on what has changed in the tables from previous job run; these jobs are Delta Processing Jobs. Delta processing is usually performed on tables that contain a large number of rows so it is probable that in these tables there will be columns eligible for delta processing. The delta comparison column must contain a value that is modified every time that the row is modified and this value must always be increasing. Any data type that meets this requirement is supported. By default, the columns of the table that are mapped to Numeric or Timestamp comparison formats are supported. For example, TIMESTAMP, TIMESTAMP\_TZ, and NUMBER.

The delta base is the value of the Delta Column on the basis of which the delta comparison was performed. Every time a comparison is run, a delta base value is captured. Depending on the number of delta comparison jobs performed, there can be multiple delta base values so a list of delta base values for the compare pair is generated. For example, the first time a Full Comparison is run and the maximum value of the Delta Column is the delta base, DeltaBase-1. A second Delta Processing Job run based on DeltaBase-1 results in DeltaBase-2 being captured again as the maximum of Delta Column. In the third run, you can use either DeltaBase-1 or DeltaBase-2 for the comparison or run a Full Comparison Job.

#### When Should I use Delta Processing?

Delta processing is suitable for use with very large Enscribe files and SQL tables that, otherwise, would take a long time to process. It does consume additional overhead, so it is probably not practical for use with smaller sets of data. Try running a test comparison without delta processing first. If, in your opinion, the compare pair takes too long to process, try running it again with delta processing enabled. If the delta-enabled run is significantly shorter than the first test, continue to use it. If there is only marginal improvement, it might be better to disable delta processing to prevent the added overhead. The performance gains of delta processing are in the initial comparison step of the run. Delta processing can cause the confirmation step to be longer if the source and target rows end up on different data blocks.

#### What Process Performs the Delta Processing?

The delta processing is performed by the Oracle GoldenGate Veridata Agent.

For NonStop platforms, the VSN information is retrieved by a privileged process named vsnserv. During the installation of Oracle GoldenGate Veridata Agent on a NonStop platform, PROGID was used for the vsnserv program to run as SUPER.SUPER to be able to read the file labels for this purpose.

For all other platforms, the delta processing queries the compare pair to retrieve the delta base values for both the source and target tables, which creates the column mapping.

#### What sorting method can be used with delta processing?

To use delta processing, you must enable server-side sorting by setting the sorting method to Server within the profile that is associated with the compare pair or the one that is associated with the job when you run it.





### (i) Note

If you always will be using delta comparisons, then consider setting the sorting method to Server within the default Oracle GoldenGate Veridata profile. That way, nobody will forget to select the correct profile when the jobs are run.

### What other important things should I know when using delta processing?

The delta processing mechanism can fail to detect an out-of-sync delete, if that delete was the only source row that was modified in a block, and if that delete did not get propagated to the target. In such a case, the block on the target that contains the relevant row does not get modified, so it is skipped by the target Veridata Agent during delta processing.

The delta value field is pre-populated in the column mapping UI, based on the query for retrieving the delta value.

### **Support Considerations**

The following table shows the supported delta column types. The possible list of delta columns for the delta configuration UI is identified by reviewing the compare formats for the corresponding column-pairs in the source and target tables.

Veridata Comparison Format	Can it be Delta Column?
STRING	No
STRING_EN	No
TIMESTAMP	Yes
TIMESTAMP_TZ	Yes*
DATETIME	Yes
SMALLDATETIME	No
DATE	No
TIME	No
NUMBER	Yes
FLOAT	No
BINARY	Yes*
BLOB	No
CLOB	No
INTERVAL	No
DEC_FLOAT	No
BINARY_TIMESTAMP	Yes*
SYBTIME	No
CLOB_NFC	No
STRING_NFC	No



## Note

\* denotes that Hive doesn't support the delta column. Supported data types for Hive are NUMBER, TIMESTAMP and DATETIME.

For Oracle Database, the ORA\_ROWSCN pseudo delta column is supported, and selected by default when ROWDEPENDENCIS are enabled for that table.

For DB2 for i, z/OS, and LUW, columns having the GENERATED FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP clause is selected by default.

For all supported databases, TIMESTAMP columns are shown in order first then followed by NUMERIC columns.

Only one column is supported for delta processing and is similar to:

SELECT column-names from table name where delta\_column delta\_condition supplied\_delta\_value

### For example:

select \* from TableA where startdate >= '12-01-2012 21:24:00'

### How do I enable or disable delta processing?

You can enable or disable delta processing for database compare pair from the Delta Processing tab and the Existing Compare Pairs tab of the Compare Pair Configuration page. In addition, you can use the Delta Processing Enabled check box on the Run Configuration page to toggle this feature.

### How do I configure delta processing?

On the Column Mapping Configuration page, click the Delta Processing tab.By default, the options are automatically populated and the Enable Delta Processing check box is selected. You can disable the feature by clearing the check box, which renders all other options inactive.

You can use the defaults or change any of the following:

- the source or target columns for processing,
- · whether to use a source or target query
- the source or target database query

After you have configured the delta processing for the compare pair, click **Save** to apply your changes.

### How do I know that delta processing is being used?

When a compare pair is configured for delta processing, Delta Enabled text is displayed in the compare pair table of the Edit Group and Compare Pair Configuration pages.

### Can I override delta processing when I run a job?

What if I perform maintenance on the tables or files for which I am using delta processing?

When you perform maintenance on objects in a compare pair that has delta processing enabled, the best practice is to disable delta processing for the next run so that Oracle GoldenGate Veridata compares all of the rows. You can disable delta processing at the compare pair level or as a job override. Starting again with a full comparison allows a new



delta base state to be established and will make subsequent delta comparisons faster.

Otherwise, delta processing could actually take longer than a complete comparison of all rows.

For example, if a 'FUP RELOAD' is performed on the source, but not on the target, it could cause delta processing to return a much larger number of rows from the source than from the target, based on the last delta state. The source rows that are returned would be rows that actually did not change. This happens because FUP moves records around and combines blocks, but does not change the data. However, the VSNs for the affected blocks will change. Oracle GoldenGate Veridata cannot detect that the reload was done since the last delta state. Thus, the next time that the VSN for a block changes, all of the rows in that block will be returned.

Conversely, on the target, no VSNs are changed for the corresponding data (because a reload was not done there), so those rows are not returned for delta processing. This anomaly will be resolved by the confirmation step, but this slows the overall comparison process because that step is much slower than the initial comparison step.

### 5.2.2.1.3 Row Partitions

You can specify which rows to include or not to include in a comparison by specifying a SQL predicate statement or an Enscribe partition.

Oracle GoldenGate Veridata supports selecting a subset of rows for comparison by row partioning (SQL predicate statement) or by Automatic Row Partition.

Using partitions allows you to compare source and target tables or files that have the same structure but a different number of rows. For example, you could compare a production table to a data warehouse table that may contain more rows because of historical data. The usage of partitions also speeds throughput by splitting the load into multiple processing streams.

- Click the plus (+) icon in the Source Partition area. The Source Partition button is toggled on by default.
- 2. In the Name box, type a name for this partition, such as emp. Use one word that can include underscores, hyphens, and other standard keyboard special characters.
- 3. Enter a **SQL Predicate**. For example, where emp\_salary>1000. The SQL Predicate Statement is a condition expression, where the WHERE clause is used to evaluate to a boolean value, either true or false. You can also edit or delete the SQL Predicate by clicking the Pencil or the Delete icons adjacent to it.
- Similarly, enable Target Partition, and click Copy from Source to copy the SQL Predicate statement details from the source.

## 5.2.2.1.4 Deleting a Compare Pair

Use the Compare Pair Configuration page to delete compare pairs.

You must have the Administrator or Super User role to delete compare pairs.

- To delete a Compare Pair, on the Groups and Compare Pairs page > Compare Pairs section, select a Compare Pair, click More Actions, and then select Delete.
- Click Delete in the confirmation dialog box to confirmation deletion.

# 5.3 Profiles

A profile is a set of global processing parameters, each containing unique settings for a specific purpose. Parameters or properties in the profile define how a comparison job or repair job behaves. Oracle GoldenGate Veridata provides a default profile, but you can create your



own profiles. You can create as many profiles as needed and associate them with any job or compare pair (to override the job profile; see **Compare Pairs**). You can override profile assignments at run time.

You must have the Administrator or Super User role to create, edit, or delete a profile.

You can create, edit, clone, delete, and view profile information from the **Profiles** page.

#### To create a Profile:

- 1. Click **Profiles** in the navigation pane.
- On the Create a Profile page, click Create.
- 3. Enter the name and descriptoin for the new profile.
- **4.** From the **Create new profile using copy of existing profile** drop-down, select an existing profile you want copy to create a new profile.
- 5. Click Submit

#### To edit a Profile:

- 1. Click the profile name or the corresponding edit (Pencil) icon.
- 2. On the **Profile Information** page, edit the required settings and click **Save**.

See Editing Profile Settings.

#### To clone a Profile:

- 1. Click the copy icon next to the profile that you want to clone.
- Enter a name and a description for the new profile. Click Clone. You can now see the cloned profile in the profiles list.

#### To delete a Profile:

Click the corresponding delete icon in the **Actions** column. Alternatively, click the profile name, then click **Delete** on the **Profile Information** page.

### Change the default profile

Oracle GoldenGate provides a default profile (see <u>Using the Default Profile</u>) that is used for all of the compare pairs and jobs that are not linked to a user-defined profile. This profile is used automatically unless another profile is selected when creating a compare pair or a job, or when running a job. You can change the settings of the default profile by editing it.

- Editing Profile Settings
- Using the Default Profile

# 5.3.1 Editing Profile Settings

This section describes the parameters that you can edit for a profile.

- 1. General: Controls output options. You can set the following output options:
  - Report In-sync rows to Report file: Prints all the in-sync rows to the veridata report.
  - Report In-sync after In-flight rows to Report file: Print the in-flight rows that becomes in-sync after the Confirm Out-of-Sync phase to Veridata report.
  - Out-of-Sync Output Format: Defines the format in which out-of-sync file gets generated. The default format is Binary. Supported values are Binary, XML, Both and None. If you select None, you will not be able to view the out-of-sync data.



- Maximum size of each Out-of-Sync XML chunk (rows): You can set the maximum chunk size for the number of out-of-sync rows for the server-to-agent calls. Larger chunk size boosts perfomance by minimizing the server-to-agent calls, but utilizes more network bandwidth.
- Comparison Type: To enable comparison by row count, set this parameter to Row Count. To enable complete data comparison for a table, set this parameter to Full.
  - Default value: Full.
  - Supported Values: Full, Row Count.
- Sorting Method: Controls sorting method and memory management. Data is sorted to match keys (or a key specification) so that the correct source and target rows are compared.
  - **Sort Data Using:** Selecting the **Server** option sorts the data on the server disk, and is more efficient compared to database sorting, which sorts the data in database using the **order by** clause. Default value of this parameter is **Server**.
  - Temporary Storage Directory for Source Data: Creates the sort file for the source data in the specified directory on the Server disk. Default location is the /tmp directory. Ensure sufficient disk space is available in the given location to store the sorted data.
  - Temporary Storage Directory for Target Data: Creates the sort file for the target data in the specified directory on the Server disk. Default location is the /tmp directory. Ensure sufficient disk space is available in the given location to store the sorted data.
- **Initial Compare**: Controls parameters for the process that performs the initial compare step.
  - Output Out-of-Sync Record Details to Report File: Select this option to print all the out-of-sync records to Veridata report. Default value is NO.
  - Terminate when Maximum records Out-of-Sync: Stops the comparison when the out-of-sync records reaches the limit specified. This parameter is valid only if you do not select the **Perform Confirm Out-of-Sync Step** parameter under the **Confirm Out**of-Sync category.



#### (i) Note

If Terminate when Maximum COOS records out of sync number is not divisible by the batch size, then there are extra records in the report. The number of records in the report is always multiple of the batch size. For example, if the batch size is 10 and Terminate when Max COOS out of sync records is 15, and then the report will have 20 records.

- Update Report file every (records): Enter the batch size after which the report file should be updated.
- **Update Report file every (seconds)**: Enter the time interval after which the report file should be updated.



#### Note

Updating Report file every (seconds) takes precedence over Updating Report file every (records), when you set values for both the parameters.



- Limit Number of Input Rows: Limits the number of rows to process from the table.
   Useful for estimating the amount of time that a comparison will take. Limit the number of rows, then run a comparison for them. Afterward use the results to calculate how long the full comparison would take. This parameter is also useful for testing and debugging so that the run completes in a shorter time than if the entire table was processed.
- **Source Oracle Optimizer Hint**: (Oracle) Specifies an Oracle hint that is passed to the agent on the source database when a comparison starts.
- Target Oracle Optimizer Hint: (Oracle) Specifies an Oracle hint that is passed to the agent on the target database when a comparison starts.
- **Max Concurrent Comparison Threads**: Set the number of compare pairs to be run in parallel. The remaining compare pairs is set to waiting/ pending state.
- Event Reporting
  - Generate Messages: Select the type of message to be generated in the veridata report.

Supported values: Info, Warn, Both and None.

Default Value: None.

- Generate warning messages for Out-of-Sync rows after (differences): Set the number of out-of-sync records after which warning messages should be generated in Veridata report.
- Agent (This is applicable only for C-Agents)
  - Use Static Listening Port for Agent during Row Hash on Source (0 to use dynamic port list): Use Static Listening Port for Agent during Row Hash on Source (0 to use dynamic port list)
  - Use Static Listening Port for Agent during Row Hash on Target (0 to use dynamic port list): Use Static Listening Port for Agent during Row Hash on Target (0 to use dynamic port list)

These settings are for debugging. At the default of 0, the Oracle GoldenGate Veridata Agent is started by the Oracle GoldenGate Veridata Manager during the initial comparison step. By specifying a port number instead, you can direct Oracle GoldenGate Veridata Server to interact with Veridata Agent through that port.

### Non Stop Process

- Source Process CPU Number: : This field designate a CPU for the process on the source system. The valid range is -1 (the default) to 16. When set to -1, the Oracle GoldenGate Manager process will attempt to start the processes in a round-robin fashion.
- Source Process Priority: This field designate the priority at which the process will run on the source system.
- Source process name starting with = Must start with dollar symbol, followed by two letters, and end with \*: This value designate the name to use for the process on the source system. This name must start with a \$ sign followed by two letters and end with an asterisk, for example \$AA\*.
- Target Process CPU Number: This field designate a CPU for the process on the target system. The valid range is -1 (the default) to 16. When set to -1, the Oracle GoldenGate Manager process will attempt to start the processes in a round-robin fashion.
- Target Process Priority: This field designate the priority at which the process will
  run on the target system.



- Target process name starting with = Must start with dollar symbol followed by two letters, and end with \*: This value designate the name to use for the process on the target system. This name must start with a \$ sign followed by two letters and end with an asterisk, for example \$AA\*.
- Confirm-Out-Of-Sync: Controls parameters for the process that performs the confirmation step.
  - Output Out-of-Sync Record Details to Report File: Enable this option to get the Outof-Sync records in the compare-pair report file.
  - Terminate when Maximum records Out-of-Sync: Set the number of out-of-sync records, after which veridata should stop the (compare) job.
  - **Update Report file every (records)**: Enter the batch size after which the report file should be updated.
  - Update Report file every (seconds): Enter the time interval after which the report file should be updated.



#### Note

Updating Report file every (seconds) takes precedence over Updating Report file every (records), when you set values for both the parameters.

- Source Oracle Optimizer Hint: (Oracle) Specifies an Oracle hint that is passed to the agent on the source database during COOS comparison.
- Target Oracle Optimizer Hint: (Oracle) Specifies an Oracle hint that is passed to the agent on the target database during COOS comparison.
- Run Concurrently With Initial Compare: Enable this parameter to run the Confirm Out-of-Sync and Initial Compare phase concurrently. Setting the value to No ensures that Confirm Out-of-Sync is started only after Initial compare is completed.

Default value: Yes.

- Delay Confirm-Out-of-Sync By (seconds): Specify the time interval by which the Confirm Out-of-Sync phase should be delayed.
- Confirm-Out-of-Sync Batch Size: Set the batch size to be processed in the Confirm Out-of-Sync phase. Higher batch size increases the efficiency by reducing the network calls between Server and Agent, but it utilizes more network bandwidth.
- Perform Confirm Out-of-Sync Step: Enable this parameter to perform the Confirm Out-of-Sync step. By default it is enabled. You can disable this parameter if you are comparing static data.



### (i) Note

If you disable this option, you cannot view the Out-of-sync data.

- **Event Reporting** 
  - Generate Messages: Select the type of message to be generated in the veridata

Supported values: Info, Warn, Both and None.

Default Value: None.



- Generate warning messages for Out-of-Sync rows after (differences): Set the number of out-of-sync records after which warning messages should be generated in Veridata report.
- Agent (This is applicable only for C-Agents)
  - Use Static Listening Port for Agent during Row Hash on Source (0 to use dynamic port list): Use Static Listening Port for Agent during Row Hash on Source (0 to use dynamic port list)
  - Use Static Listening Port for Agent during Row Hash on Target (0 to use dynamic port list): Use Static Listening Port for Agent during Row Hash on Target (0 to use dynamic port list)

These settings are for debugging. At the default of 0, the Oracle GoldenGate Veridata Agent is started by the Oracle GoldenGate Veridata Manager during the initial comparison step. By specifying a port number instead, you can direct Oracle GoldenGate Veridata Server to interact with Veridata Agent through that port.

#### Non Stop Process

- Source Process CPU Number: : This field designate a CPU for the process on the source system. The valid range is -1 (the default) to 16. When set to -1, the Oracle GoldenGate Manager process will attempt to start the processes in a round-robin fashion.
- Source Process Priority: This field designate the priority at which the process will run on the source system.
- Source process name starting with = Must start with dollar symbol, followed by two letters, and end with \*: This value designate the name to use for the process on the source system. This name must start with a \$ sign followed by two letters and end with an asterisk, for example \$AA\*.
- Target Process CPU Number: This field designate a CPU for the process on the target system. The valid range is -1 (the default) to 16. When set to -1, the Oracle GoldenGate Manager process will attempt to start the processes in a round-robin fashion.
- Target Process Priority: This field designate the priority at which the process will
  run on the target system.
- Target process name starting with = Must start with dollar symbol followed by two letters, and end with \*: This value designate the name to use for the process on the target system. This name must start with a \$ sign followed by two letters and end with an asterisk, for example \$AA\*.
- Repair: Controls parameters for the repair process.

#### Repair

- Repair Batch Size: Enter the batch size of the records to be repaired. Higher batchsize results in higher efficiency.
   Default value: 1000.
- Repair Transaction Size: Specifies the number of repair operations included in the database transaction at the target. The special value "0" indicates that the transaction size should be the same as the repair batch size.
- Number of Concurrent Repair Operations: Set the number of repair operations to be run concurrently.
- Check changed values:



- \* If you enable this parameter, and the non-key column records are updated post comparison, the changed out-of-sync records will not be repaired.
- If you disable this parameter, and the non-key column records are updated post comparison, the changed out-of-sync records will also be repaired. This parameter is enabled by default.
- Terminate when Maximum Repair Warnings: Set the number of warnings after which the repair job should be stopped.
- Run Repair Automatically after Compare: Enable this parameter to run repair automatically after comparison.
   Default value: No.
- Repair SQL Path: This is the path where the Repair SQL gets generated on Veridata server before being copied to local disk. By default it will be generated under /tmp. Once the file is copied to local disk, it gets deleted from Veridata server location.

### Repair Reporting

 Write Repair Success messages to Report: Enable this parameter to write the repair success messages to the report file.

### Disable DB Triggers Session Based

 Disable DB Triggers Session Based: For databases that support disabling triggers (Oracle and Sybase), when you enable this setting, the Veridata Agent will issue the required commands to disable triggers while applying repair updates.
 Default value: No.

### **Specifying a Sorting Method**

The Sorting Method settings of the Profile Information page specify whether data sorting will be performed by the database or by Oracle GoldenGate Veridata Server. Specify this method in the Sort Data Using option on the **Profile Information** page.

By default, Oracle GoldenGate Veridata uses the database to sort data for comparison. This default is due to historical conditions that are no longer valid. Server-side sorting is the current recommended sorting method. Database sorting should only be considered when the ordering produced by the database is identical to the ordering produced by Oracle GoldenGate Veridata Server. Following is a list of the types of conditions that will produce differing sorted ordering of the rows:

- Character encoding conditions: Oracle GoldenGate Veridata compares character data as UTF-8 encoded bytes. To match server-side ordering, key columns that contain character data must contain only ASCII data or be encoded using UTF-8, and the database must use binary comparisons for character data (no comparisons that are case-insensitive or specific to a locale).
- Some datetime data types, such as Teradata TIME, may sort differently in the database and in Oracle GoldenGate Veridata Server.
- To make database ordering consistent with the ordering done by Oracle GoldenGate
   Veridata Server, the Oracle GoldenGate Veridata Agent may add ORDER BY clauses to the
   initial comparison SELECT statement that will make the database ignore indexes on the
   columns. An example is TIMESTAMP with TIMEZONE data, where Oracle GoldenGate
   Veridata orders the data by the string representation of the data rather than by the absolute
   time.

When Oracle GoldenGate Veridata Server performs the sort, the Oracle GoldenGate Veridata Agents return data in the natural order that is provided by each database, and then the data is



sorted by two server sort processes, one to sort source rows and the other to sort target rows. Server-side sorting supports a maximum row length of 32768 bytes. This limit normally is not exceeded when the hash comparison method is used.

### Specifying Temporary Storage Directory for Source and Target Data

Specifies a location on the source disk or target disk to use as temporary storage when there is not enough memory to process all of the data that is being sorted. If no locations are defined, the default is to use a directory under the Oracle GoldenGate Veridata Server home location.

Choosing locations on different physical drives might speed up comparisons in some circumstances. You can specify multiple locations for each process, separating each one with a semicolon (for example /tmp/sort1; /tmp/sort2). All locations specified must already exist. The drives used should have sufficient free disk space. To calculate the approximate amount of space needed, use this formula:

1.5 \* (Trows \* (Tkey + 20)) \* nTables

where:

Trows = the number of rows in table

Tkey = the average size of the table key, in bytes

nTables = the number of tables that are being compared

# 5.3.2 Using the Default Profile

The default Oracle GoldenGate Veridata profile is selected when a custom profile is not linked to a job or compare pair.

The name of the default profile is \$default.

You can view and edit the settings for this default profile on the **Profile Information** page.

To edit the default profile, you must have the Oracle GoldenGate Veridata administrator role.

#### To edit the default profile settings:

Click **Profiles** on the Navigation pane, to display the **Profile Information** page. Click the default profile name to display the **Profile Information** page. Make the required edits and click **Save**.

To reset to system default settings:

Click **Profiles** on the Navigation pane, to display the **Profile Information** page. Click **Reset to System Defaults** button and click **Save**.

# 5.4 Jobs

To run comparisons, you must run a job. The job configuration determines which compare groups are processed.



#### Note

Before creating a job, you must create at least one compare group (see <u>Groups and Compare Pairs</u>) to link to the job. To use customized runtime parameter settings, you must also create at least one profile (see <u>Profiles</u>). Otherwise, the job will use the default profile that is supplied by Oracle GoldenGate Veridata.

You must have the Administrator or Super User role to create, edit, or delete a job.

- Overview
- Creating a Job
- Running Jobs
- Editing a Job
- Monitoring Jobs
- Estimating Comparison Time
- Using the Comparison Report
- Repairing Out-Of-Sync Jobs

## 5.4.1 Overview

Once you have configured groups and compare pairs into jobs, you can run those jobs. Oracle GoldenGate Veridata enables you to control which groups and compare pairs are processed during any given job run, and with which runtime parameters. Once a job is running, you have easy access to views of current and finished runs.

#### **Related Topics**

Creating a Job

Editing a Job

**Monitoring Jobs** 

Running a Job

Repairing Out-of-Sync Jobs

**Estimating Comparision Time** 

**Using the Comparison Report** 

# 5.4.2 Creating a Job

Click **Jobs** in the navigation pane.

On the **Jobs** page that opens, click **Create** and enter the following parameters:

- Name: Enter a job name. This is a required field and you cannot edit the name once the
  job is created and saved.
- Description: Enter a relevant description for the job.
- **Profile**: Select the profile that is associated with this job, from the drop-down list. If you do not select a profile, the default profile will be used.



- Click Add Group, to add a group to this job.
- On the Add Group page that comes up, a list of available groups is displayed.
- Select the groups that you want to add the job, and click Add & Save.

### Note

A job cannot be created without any groups added to it. You should add at least one group to a job while creating it.

Click Submit.

# 5.4.3 Running Jobs

On the **Run Jobs** page, you can run any job that has been previously configured. Before running a job, you may want to estimate its duration. See <u>Estimating Comparison Time</u>.

- 1. Click **Run Jobs** in the navigation pane.
- 2. Select a job from the drop-down.
- 3. Select a profile from the drop-down.
- Select the Run Repair Automatically after Compare, if you want to run repair automatically after the compare job is completed.
- 5. Click **Expand All**, to expand the Compare Pairs list and to reset any parameters.
- 6. Click Run Job.

To reset the Compare Pair Parameters:

Click **Expand All**, to expand the Compare Pairs list.

The following information is displayed in the Compare Pairs list. This information can help you to refine the job configuration.

- Row Partitions: Enables you to configure or override manual row partitions.
- Compare Pair Name: The name of the compare pair. To view actual object names, hover the mouse over the name.
- **Delta Base Time**: Shows the end time of the previous comparison, which provides a start time on which to base delta processing for the next run.

#### (i) Note

You have to enable delta processing option in the **Groups and Compare Pairs** page.

- To override the selected Delta Base Time, click Delta Base Time.
- On the Delta Base Time Selection page that opens, select the required Delta Base Time.
- 3. Click Save.
- Source Table: The source table selected while generating the compare pair.
- Target Table: The target table selected while generating the compare pair.



Overriding Manual Row Partitions

### 5.4.3.1 Overriding Manual Row Partitions

- 1. To override a row partition, click **Override** in the **Row Partition** column.
- 2. In the Override page that comes up, select the partition you want to apply, or toggle the **Enable Source Partition** and **Enable Target Partition** buttons, to compare the full tables.
- Click Save.

## 5.4.4 Editing a Job

- 1. From the list of jobs, click the job that you want to edit.
- On the Job Details page, you can edit the description, profile, add or delete groups linked to the job.



You cannot edit the job name.

To add or delete Groups:

 Click Expand All, in the Linked Groups section, to display the list of groups that are linked to this job.

To add a group to this job, click Add Group.

- a. On the Add Group page that comes up, a list of available groups is displayed.
- b. Select the groups that you want to add to the job, and click Add & Save.
- 2. To delete a linked group, click the delete (trash can) icon next to the group.
- 3. Click **Save** to save the changes to the job.

### 5.4.5 Monitoring Jobs

- Viewing Completed Jobs
- Viewing Jobs that are Running
- Viewing Details of the Repair Jobs

## 5.4.5.1 Viewing Completed Jobs

To view the near-real-time information about completed jobs, click **Monitor Jobs** in the navigation pane, and select the **Completed Jobs** tab.

This page displays details of all the completed jobs:

- Comparison Status: The status of the comparison.
- Repair Status: The repair status of the comparison. It is Successful, Failed, or left blank when the repair job is not run.
- Name: The name of the job.
- Start Time: The job process start time.
- End Time: The job process end time.



- Run Duration: Time taken to process the job.
- Compare Pair Total: The total number of compare pairs in the job.
- Out-of-Sync Compare Pairs: the number of compare pairs that are out of sync.
- Failed Compare Pairs: The number of compare pairs failed in this comparison.
- Cancelled Compare Pairs: The number of compare pairs cancelled.
- Report: Click this link to view the job report.
- Actions: You can delete the Run information of the job.

To delete mutiple jobs, select the jobs from the list and click **Delete**.

By default, this page shows all the jobs that are processed. Use the Time Range Filter to constrain the view as needed.

#### To use the Time Range Filter

- Most Recent Comparison Run: Selects the most recent run for each job that exists.
- View Last: Selects all of the runs of all of the jobs that started within the last <n> Hours or Minutes.
- From: Selects all of the runs of all of the jobs that finished within a specific date range, with the option to include the time of day. Supply date and time values.

Click **Apply** to activate your selection.

#### **Basic filter options**

You can filter by:

 Comparison Status: show all jobs (default) or constrain the list based on jobs with a specific comparison status.

Possible comparison status conditions:

- In Sync
- Out-of-Sync
- Failed
- Skipped
- Cancelled
- Repair Status: show all jobs (default) or constrain the list based on jobs with a specific repair status.

Possible repair status conditions:

- Pending
- Running
- Out-of-Sync
- Successful
- Failed
- Warning
- Cancelled



#### **More Details**

Click an invidual job name in the list to view:

- Job Summary: Displays the overall job summary, Compare Pairs Run Summary chart, and the list of compare pairs. Click a compare pair name to view the **Compare Pair** Details, Performance, and the Report tabs.
  - Compare Pair Details section:
    - Displays a chart view of the comparion details and a graph view of the out-of-sync operations.
    - Displays the Out-of-Sync and Skipped Row Details, which you can filter by Status, Operation, or Columns.
    - You have an option to download Repair SQL or to repair the compare pair.



#### (i) Note

Download Repair SQL option is available only for Jobs that have been configured for datasource connections with the Oracle datatype.

- **Performance**: Displays the comparison performance details at different comparison stages such as Sorting, Initial Compare, and Confirm Out-of-sync. The page also displays a graph view with the performance details for each stage, which you can filter based on:
  - Run Duration
  - **Rows Processed**
  - Rows Per Second
- **Report**: Displays the following details about the selected comparison:
  - Overiew
  - **Profile**
  - **Parameters**
  - **Agent Info**
  - **Performance Stats**
  - **Sort Statistics**
  - **Summary**

You can download the individual comparison report for a selected compare pair, from this page.

- From the Job Summary page, click More Actions for the Repair and Download Repair SQL options.
- Click **Run Job**, to run the current job again.
- Out-of-Sync Summary: Displays the out-of-sync details at each group and compare pair levels. You can filter the details by:
  - Groups
  - Status
  - Operation



Columns

You can also download the Repair SQL or repair the target from this page.

- Report: Displays the the report for a specific job, group, or compare pair. The report shows processing results and the causes of errors:
  - Results
  - Overiew
  - Profile
  - Run Options
  - Summary

You can download the following reports from this page:

- Overall Job Report
- Job and Compare Pairs Reports

### 5.4.5.2 Viewing Jobs that are Running

To view the near-real-time information about running jobs, click **Monitor Jobs** in the navigation pane, and select the **Running Jobs** tab.

The list view shows statistics and status for the running comparisons. By default, the list is organized within the context of the job as a unit. You can filter the list to organize itby time range, or by comparison status.

- Comparison Status: The current run status of the comparison. For example, Running, Waiting.
- Name: The name of the job.
- Start Time: The job process start time.
- Run Duration: The duration for which the job has been running.
- Out-of-Sync Compare Pairs: The number of out-of-sync compare pairs.
- Waiting Compare Pairs: The number of compare pairs which are yet to be processed.
- Running Compare Pairs: The number of compare pairs that are currently being processed.
- Failed Compare Pairs: The number of compare pairs that have failed the comparison job.
- Cancelled Compare Pairs: The number of cancelled compare pairs.
- Processed Compare Pairs: The number of all the processed compare pairs. This
  includes in-sync, out-of-sync, and failed compares pairs.
- Actions: Click the stop (close) icon to the stop the job.

By default, this page shows all the jobs that are processed. Use the Time Range Filter to constrain the view as needed.

#### To use the Time Range Filter

- Most Recent Comparison Run: Selects the most recent run for each job that exists.
- View Last: Selects all of the runs of all of the jobs that started within the last <n> Hours or Minutes.



• From: Selects all of the runs of all of the jobs that finished within a specific date range, with the option to include the time of day. Supply date and time values.

Click **Apply** to activate your selection.

#### **Basic filter options**

You can filter by:

 Comparison Status: show all jobs (default) or constrain the list based on jobs with a specific comparison status.

Possible comparison status conditions:

- In Sync
- Out-of-Sync
- Failed
- Skipped
- Cancelled
- Running
- Cancelling

#### **More Details**

Click an invidual job name in the list to view:

- Running Job : Displays the following details.
  - Start Time
  - Run Duration
  - Comparison Status
  - Option to stop the running job
  - The Compare Pair Status section dispays the graph view of compare pairs in each stage.
  - Click an individual Compare Pair Name in the list, to view the Compare Pair Details page:
    - You can view the individual compare pair details and a graph view of total out-ofsync rows in the **Intial Compare**, **Persistent Out-Of-Sync** stages, and the rows **In-Sync After In Flights**.
    - \* You can stop an indivual compare pair comparison from this page.

## 5.4.5.3 Viewing Details of the Repair Jobs

To view the information about the repaired jobs, click **Monitor Jobs** in the navigation pane, and select the **Repair Jobs** tab.

- Repair Status: The status of the repair. For example, Successful, Failed.
- Name: The name of the job.
- Start Time: The repair job process start time.
- Run Duration: Time taken to process the repair job.
- Processed Compare Pairs: The total number of compare pairs in the repair job.



- Repaired Compare Pairs: The number of compare pairs which were successfully repaired.
- Failed Compare Pairs: The number of compare pairs that have failed the repair.
- Cancelled Compare Pairs: The number of cancelled compare pairs.
- Actions: Click the stop (close) icon to the stop the job.

#### To use the Time Range Filter

- Most Recent Comparison Run: Selects the most recent run for each job that exists.
- View Last: Selects all of the runs of all of the jobs that started within the last <n> Hours or Minutes.
- From: Selects all of the runs of all of the jobs that finished within a specific date range, with the option to include the time of day. Supply date and time values.

#### Click Apply.

#### **Basic filter options**

You can filter by

- Repair Status: show all jobs (default) or constrain the list based on jobs with a specific comparison status.
  - Pending
  - Running
  - Out-of-Sync
  - Successful
  - Failed
  - Warning
  - Cancelled

#### **More Details**

Click an invidual repair job name in the list to view:

- Repair Summary: Displays the list of all the Compare Pair in the repair job.
  - Click an individual compare pair name in the list, to view the Repair Details page.
  - You can filter the Repair Row Details by:
    - Status:
      - Pending
      - Running
      - Out-of-Sync
      - Successful
      - \* Failed
      - \* Warning
      - Cancelled
    - \* Operation:
      - \* Insert



- Update
- Delete
- \* Columns:
  - \* Key
  - Out-of-Sync
- Report: Displays the the report for a specific job, group, or compare pair. The report shows
  processing results and the causes of errors:
  - Results
  - Overiew
  - Profile
  - Run Options
  - Summary

You can download the following reports from this page:

- Overall Repair Job Report
- Job and Compare Pairs Repair Reports

### 5.4.6 Estimating Comparison Time

When tables are large, you might want to estimate the amount of time that a comparison will take before running a full comparison. To get an estimate, run a test comparison of a limited number of rows. For example, if there are 100 million rows in a table, you can run a comparison for the first million rows and then multiply that amount of time by 100.

To specify the number of rows compared

- 1. On the navigation pane, click **Profiles**.
- 2. In the list, click the name of the profile that is linked to the job (or select it, and then click edit (pencil) icon.
- 3. In the Profile Information section, Click Initial Compare.
- 4. Clear the **Use Default** toggle of the **Limit Number of Input Rows** parameter, and then type the number of rows that you want to compare.
- After running the test comparison, change Limit Number of Input Rows back to the default for the full comparison run.

# 5.4.7 Using the Comparison Report

To view a comparison report, use the Reports page.

A comparison report is generated for each job, group, and compare pair that is finished being processed. It contains summary details about out-of-sync row counts, the number of records processed, performance statistics, errors, and so forth.

The comparison report tells you how extensive an out-of-sync problem is. It also provides performance statistics and, optionally, column details. The comparison report can be viewed by any user role.



#### To view a report

To view a report, click the name of the report in the Report Name column of the Existing Reports list.

Alternately, you can click the **Select** button in the list next to the name of the report, and then click **View**. The Report View page will be displayed.

#### Viewing the report file on disk

The report is a text file that is stored on disk. It can be viewed from its location on disk.

# 5.4.8 Repairing Out-Of-Sync Jobs

To repair Out-of-Sync Comparisons

Click Monitor Jobs in the navigation pane, and click the Completed Jobs Tab.

Click a job name in the list to view the Job Summary.

- Click More Actions and click Repair.
- To repair an individual compare pair, click the compare pair name in the list to open the Compare Pair Details page, and click Repair.

Oracle GoldenGate Veridata allows repair of duplicate records for the tables that do not have Primary Key's in the following databases: Oracle, MySQL and MSSQL. The Details for Out-Of-Sync page is displayed.



If the table contains duplicate records, then it is suggested to use System Generated Mapping in Compare-Pair. If you decide to use User-defined mapping then, ensure that the user-defined column mappings have unique dataset. If the row has Skipped status icon, then the row is ignored from Repair.

The Repair Jobs page displays a summary of all repair jobs. You can use the Filters on this page to display older repair jobs and to filter repair jobs by repair status and job name. Following are the possible repair statuses:

- Pending
- Running
- Out-of-Sync
- Successful
- Failed
- Warning
- Cancelled
- <u>Downloading SQL Statements for Out-of-Sync Records</u>
   Oracle GoldenGate Veridata provides the Download Repair SQL functionality to view the SQL Queries generated as part of Veridata Repair. The Download SQL Queries is enabled

only when either Oracle or SQL Server is used as the target database.

Veridata 23c (23.4.0.0.0) F85008-04 Copyright © 2015, 2025, Oracle and/or its affiliates.



### 5.4.8.1 Downloading SQL Statements for Out-of-Sync Records

Oracle GoldenGate Veridata provides the Download Repair SQL functionality to view the SQL Queries generated as part of Veridata Repair. The Download SQL Queries is enabled only when either Oracle or SQL Server is used as the target database.

You can look at these SQL Statements before the Oracle GoldenGate Veridata executes them onto the target database, or execute these SQL statements by yourself on any of the other database tools. With the Download Repair SQL functionality, you can download the SQL statements for all your out-of-sync-records, and can also execute them at your convenience. To download SQL queries:

1. From the **Finished Jobs** page, select the out-of-sync comparisons.

You can select Jobs, groups, and compare pairs for downloading SQL statements.

2. To download SQL queries for out-of-sync data, select the job from the finished jobs and click **Download Repair SQL**.

The **Download Repair SQL** button is enabled only for users who have access to the repair access functionality.

The SQL files are downloaded to the browser where the application is being used. When you click **Download Repair SQL**, the file gets downloaded to the browser.

If the record is already repaired, then the check box against the record is replaced with date and time of repair. Those records are not available for downloading SQL statements.

Executing the SQL File

### 5.4.8.1.1 Executing the SQL File

To execute the SQL file:

1. When you click **Download Repair SQL**, a SQL file is generated. Copy this zip to the target system. In order to enable access to target Oracle DB, run the following query on the target database. This is a one-time operation only:

CREATE OR REPLACE DIRECTORY VDT\_LOB\_DIR AS '<path of lob files>';

For example:

CREATE OR REPLACE DIRECTORY VDT\_LOB\_DIR AS '/scratch/lobs';.

Here:

/scratch/lobs is the path on the target system where the generated SQL files and LOB files should be copied. When the SQL scripts are executed from the DB prompt, the LOB files are read from /scratch/lobs

2. Unzip the zip file.

LOB files are generated only when the table has BLOB datatypes and huge data for CLOB/NCLOB.

The directories are created as follows:

<JobName><Timestamp>

- <GroupName>
  - SQL file (format <ComparePairName>.sql)
  - LOB file (format <TableName>\_<ColumnName>\_<Timestamp>.lob)



- <Group2>
  - SQL file
  - LOB files
- o SQL file (format < Compare Pair Name > . sql)
- o LOB file (format <TableName>\_<ColumnName>\_<Timestamp>.lob)
- 3. Copy the SQL and LOB files to VDT\_LOB\_DIR. However, it is not mandatory to copy the sql file to VDT\_LOB\_DIR. But, the LOB files are required in VDT\_LOB\_DIR.
- 4. If the data has any multibyte characters, then ensure that the NLS\_LANG environment variable is set before launching the DB Terminal.

#### On Linux:

- a. Bourne/Bash Shell
  - export NLS\_LANG=<NLS\_LANGUAGE>\_<NLS\_TERRITORY>.<NLS\_CHARACTERSET>.
     For example, export NLS\_LANG=AMERICAN\_AMERICA.AL32UTF8
- b. C/TCSH Shell
  - setenv NLS\_LANG <NLS\_LANGUAGE>\_<NLS\_TERRITORY>.<NLS\_CHARACTERSET>.
    For example, setenv NLS\_LANG AMERICAN\_AMERICA.AL32UTF8

#### On Windows:

 Set the NLS\_LANG in System Variable. Value should be <NLS\_LANGUAGE>\_<NLS\_TERRITORY>.<NLS\_CHARACTERSET>.
 For example: AMERICAN\_AMERICA.AL32UTF8

Note: Launch the DB terminal after NLS LANG is set.

5. Login to DB terminal and execute the SQL file: SQL> @<sql file>. For example: SQL> @VeridataRepair.sql. If prompted, enter the value of NLS\_CHARACTERSET.

#### Note

If the SQL file is in a different location than DB terminal is open, then ensure to include the complete path of the SQL file while executing it. For example, SQL> @/ Home/User/filename.sql.

## 5.5 Users

You can manage users and user groups from the **User Management** page. You can create, delete, edit, users and user group.

- Configuring User Groups
- Configuring Users



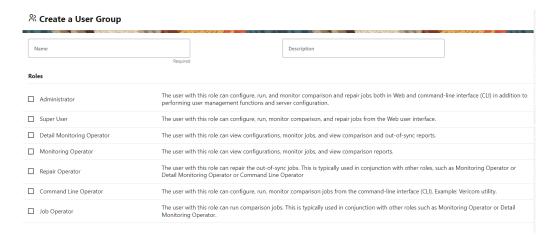
## 5.5.1 Configuring User Groups

#### Note

- You need to have an Admin User role to access this page.
- You can select from the exisiting default user groups.

#### To create a User Group:

- 1. Click **User Management** from the Navigation pane. On the **User Management** page, select the **User Groups** tab and click **Create**, to display the **Create a User Group** page.
- 2. Enter the following parameters and click **Next**:
  - Select a user role listed under the Roles tab, and click Submit.



#### (i) Note

- Configurations available to create, edit, view, delete:
  - Connections
  - \* Groups and Compare Pairs
  - \* Jobs
  - \* Users
  - \* Utilities
- Job Operator and Repair Operator role is used in conjunction with other roles such as the Monitoring Operator or Detail Monitoring Operator. You cannot login to the UI, having only the Job Operator or the Repair Operator role.

#### To view User Group details

1. Select a user group from the list under the **User Groups** tab.



You can view the user group details such as the Roles and Role descriptions under the **User Group Details** tab. You can view the users associated with the selected user group under the **Users** tab.

To update a User Group

- Select a user group from the list under the User Groups tab.
   You can edit the description and roles under the User Group Details tab.
  - You can edit the users associated with the selected user group under the **Users** tab.
- 2. Click Save.

To delete a User Group

- Under the User Groups tab, click the 3 dots (...) next to the user group that you want to delete.
- Click Delete.
- 3. Click **Delete** on the confirmation screen.

## 5.5.2 Configuring Users

Users with Adminstrator role can access this page.

To create a User:

- Click User Management menu option in the Navigation pane to display the User Management page.
- Select the Users tab and click Create.
- 3. On the User Details page, enter the following parameters and click Next:
  - Name: Enter an username. You cannot edit the name once created.
  - **Description**: A description is optional and can be edited as needed.
  - Email ID: Enter an email ID.
  - Password: Enter a password.
  - Confirm Password: Retype the password to confirm it.
- 4. On the **User Group** screen, all the available user groups are listed. Select and add all the user groups that you want to add the user to, and click **Submit**.

To view User

- Click User Management menu option in the Navigation pane to display the User Management page.
- Select a user from the list under the Users tab.
   You can view the user and the user group details on the right side of the screen, under the User Details and the User Groups tabs.

To update User details:

You can update the user and the user group details on the right side of the screen.

- User Details: You can edit the description and reset the password. After editing, click
  outside the text fields, to enable the Save option, and click Save. You cannot edit the user
  name, once created.
- User Groups: You can add or remove User Groups associated with the seleted user, and click Save.



#### To delete a User:

- Select a user from the list under the Users tab. Click the 3 dots (...) next to the user name and select Delete.
- 2. Click **Delete** on the confirmation screen.

# 5.6 Utilities

You can use the Oracle GoldenGate Veridata Import and Export utilities from the **Utilities** menu of the Veridata Web user interface. For more information, see <u>Running the Export and Import Utilities</u>.

- Export
- Import

## 5.6.1 Export

To export Jobs:

- Click Utilities from the menu to display Utilities page. The Export tab is displayed by default.
- 2. Select jobs from the **Available Jobs** list and move them to **Jobs to export**.
- Click Export to export these jobs.

The selected Jobs are exported to export.xml.

For more information, see Running the Export and Import Utilities

# 5.6.2 Import

To import Jobs:

- Click Utilities from the menu to display Utilities page. The Export tab is displayed by default.
- 2. Click **Import**. You can select either an exported file (.xml) or a GoldenGate parameter file (.prm)
- 3. Click Browse File, navigate to the location of the file, select the file, and then click Import.

A successful import message is displayed after the file import is complete.

# Administer

- Starting and Stopping the Veridata Server and the Repository
- Managing Veridata Agent
- Vericom
- Veridata Export and Import Utilities

You can use the Export and Import utilities, provided with the Oracle GoldenGate Veridata installation, to define portions of your configuration. Starting Oracle GoldenGate Veridata 23c, you can access these Utilities from the Web User Interface.

- Server Parameters
- Agent Parameters General
- Agent Parameters Connections

# 6.1 Starting and Stopping the Veridata Server and the Repository

To start the Oracle GoldenGate Veridata server:

- **1.** Go to <*VERIDATA HOME*>/bin
- 2. Run the following command: ./run.sh

To stop the Oracle GoldenGate Veridata server:

- 1. Go to < VERIDATA HOME > /bin
- 2. Run the following command: ./run.sh stop

To start and stop the repository:

- **1.** Go to <*VERIDATA HOME*>/bin
- 2. Run the following commands:
  - To start the repository: ./run.sh repoStart
  - To stop the repository: ./run.sh repoStop
  - To know the repository status: ./run.sh repoStatus

# 6.2 Managing Veridata Agent

This topic describes how to manage Veridata Agent: C-agent and Java-based components. It also includes controlling logging levels and connecting to the web user interface.

This topic includes the following sections:

- Starting and Stopping the C-Agent
- Managing Java-Based Components
- Reloading Logging Information
- Connecting to the Oracle GoldenGate Veridata Web Interface



# 6.2.1 Starting and Stopping the C-Agent

When the Oracle GoldenGate Veridata (server) initiates comparisons, the C-agent starts automatically. However, for Oracle GoldenGate Veridata Agent (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 automatic, you can stop the Manager process that controls the agent. Stopping Manager prevents the 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:

- From the agent installation location, run the Oracle GoldenGate software command-line interface.
- 2. Stop or start the Manager.

START MANAGER STOP MANAGER

# 6.2.2 Managing Java-Based Components

- Starting and Stopping the Java-Based Components
- Managing Bequeath Agent

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

The server and web user interface components are Java programs. The agent component is also available as a Java program for all platforms except NonStop.



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

To start and stop the Java agent component:

- Navigate to the AGENT\_DEPLOY\_HOME directory.
- 2. Start or stop the agent.

#### **UNIX or LINUX**

agent.sh {start | run} OR agent.sh stop

- In these commands:
- run starts the agent in the same command window where it is launched.
- start starts the agent in a separate command window.





#### (i) 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 to 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.

Configure the host to start and stop the processes automatically. Contact your system administrator if you need assistance.

## 6.2.2.2 Managing Bequeath Agent

Bequeath connection works only when the Oracle database and the agent are installed on the same server. You are required to set the following two environment variables before starting the agent:

```
export ORACLE_HOME=<path>
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

For the JDBC URL details, see Oracle OCI bequeath database connection URL:

Also, copy the ojdbc11.jar file from the installed location at \$ORACLE\_HOME/jdbc/lib to the <Agent\_Home>/drivers directory. Then, update the agent.properties file accordingly with the following entry:

server.jdbcDriver=ojdbc11.jar

## 6.2.3 Reloading Logging Information

You can reload logging information from the <VERIDATA\_AGENT\_HOME>/config/odl.xml configuration file to a running agent.

To reload logging information, start the agent and run the UNIX/Linuz reloadLog command as follows: agent.sh reloadLog

## 6.2.4 Connecting to the Oracle GoldenGate Veridata Web Interface

To connect to the Oracle GoldenGate Veridata Web user interface:

- In a Web browser, enter the following address: http://hostname:port/veridata.
  - In this example, hostname is the name of the system where the server is installed and port is the port number where it is running (default is 8831). Use localhost as the host name if you are connecting on the system that is local to the server installation.
- On the Oracle GoldenGate Veridata Web login page, enter your user name and password. For full instructions on using the Oracle GoldenGate Veridata web user interface, see the online help.



## 6.3 Vericom

The Vericom command-line interface provides a tool for you to run comparisons.

This topic includes the following sections:

- Overview of the Vericom Tool
- Supported Commands in Vericom
- Vericom Output Examples

### 6.3.1 Overview of the Vericom Tool

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

#### You can:

 Run an entire job or a specific group within a job or a specific compare pair of a group and job.

For specific compare pairs, you can:

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

You can also run comparisons from the Oracle GoldenGate Veridata web user interface. This interface provides greater control for configuring the objects to be compared and for controlling runtime parameter settings.

#### Note

If the **SSL / TLS Security for Web** option is selected during the Oracle GoldenGate Veridata installation, then do the following for Vericom to work:

Copy the <VERIDATA\_HOME>/config/vdtWebKeystore.p12 to <VERIDATA\_HOME>/cli/config/veridata-23c.p12

## 6.3.2 Supported Commands in Vericom

The following commands are supported in Oracle GoldenGate Veridata:

- 1. -u OR -user : Veridata user running veridata job
- 2. -host: Veridata helidon hostname, bydefault, it is localhost
- 3. -port: Veridata Server port. Optional paramter, if not passed then default port will be taken
- -j OR -job <jobname> : Veridata job name to be executed
- -j OR -job <jobname> -g OR -group <groupName> : If any specific group within a job to be executed
- 6. -j OR -job <jobname> -g OR -group <groupName> -c OR compare\_pair <CPName>: If any specific Compare Pair needs to be executed



- 7. -repair: If repair has to be run along with Compare
- 9. -pS: Source Partition Name (It takes default partition name)
- 10. -pS <source partition name>: To override source manual row partition at runTime
- **11. -pT:** Target partition Name (It takes default partition name)
- 12. -pT <target partition name> : To override target manual row partition at runTime
- 13. -delta: Run with delta enabled. When not passed, all rows will be compared in every run
- 14. -help:Parameter to provide Vericom usage
- 15. -v: Should provide Veridata Server version details.
- **16. -monitorStats** < runId>: This command is to monitor the Job status. It takes the runId as input.

#### Examples

 When running the Job via Vericom, it is necessary to show the job status as follows: / vericom.sh -j <jobName>.

#### Output:

Run ID: (2258, 0, 0)

2. To fetch the Run stats, run the command as follows: /vericom.sh -monitorStats <runId>:

#### Output:

Run ID: (2257, 0, 0) Job Start Time: 2008-03-21 22:48:05 Job Stop Time: 2008-03-21 22:48:20 Number of Compare Pairs: 3 Number of Compare Pairs With Errors: 0 Number of Compare Pairs With OOS: 1 Number of Compare Pairs Cancelled: 0 Job Completion Status: SUCCESSFUL

RUN ID: <CP RUN ID> <CP Name> <Group Name>

RUN ID: <CP RUN ID> <CP Name> <Group Name>

RUN ID: <CP RUN ID> <CP Name> <Group Name> and so on.

To View Compare Pair details, run ./vericom.sh -monitorStats < comparepair\_runld>
 Output

Run ID: (2257, 0, 0) Job Start Time: 2008-03-21 22:48:05 Job Stop Time: 2008-03-21 22:48:20 Number of Compare Pairs: 3 Number of Compare Pairs With Errors: 0 Number of Compare Pairs With OOS: 1 Number of Compare Pairs With No OOS: 1 Number of Compare Pairs Cancelled: 0 Job Completion Status: SUCCESSFUL

Compare Pair RUN ID:

Compare Pair Name:

Compare Pair ID:

Phase:

Status:

Total Rows Compared:

Rows In-sync:

Rows out-of-sync:

Compare-pair report:



#### ① Note

The Vericom utility fails when a special character, such as the \$ symbol, is present in the name fields of Job, Group, or Compare Pairs. To resolve this issue, an escape character needs to be added, as follows:

#### **Actual Compare Pair Name:**

 $\verb|\THOR.\$DATA08.IRFASRC.EACCT=\\|\THOR.\$DATA08.IRFASRC.BACCT||$ 

#### With Escape Character:

\THOR.\\$DATA08.IRFASRC.EACCT=\THOR.\\$DATA08.IRFASRC.BACCT

Run the command as follows:

./vericom.sh -wluser veridata -j NskTest -g nskENscribeTesting -c \THOR.\\$DATA08.IRFASRC.EACCT=\THOR.\\$DATA08.IRFASRC.BACCT

# 6.3.3 Vericom Output Examples

To view the results of a comparison that you run with the vericom tool, you can use the Oracle GoldenGate Veridata web user interface to view the comparison report. You can also view the output that is returned by the tool to the terminal. If a run finishes successfully, statistics for the job are displayed.

The following examples use the TestJob job:

#### **Example 1**

The example shows a run on a Linux system. The process exits with status 0, and finished job statistics are not displayed.

./vericom.sh -user veridata -port 8089 -job jobBetaBug

JAVA\_HOME is set to: /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home

Java version:17.0.8 [Enter user password:]

Veridata job submitted successfully!

Run ID: 325865/0/0

Execute ./vericom.sh -user <username> -monitorStats <runId> to fetch the run statistics.

#### **Example 2**

This example shows how to get statistics from the Run Id of a Job.

/vericom.sh -user veridata -monitorStats 325865/0/0

JAVA\_HOME is set to: /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home

Java version:17.0.8 [Enter user password:] Run ID: 325865,0,0

Job Start Time: 2024-06-19T05:32:42.870Z Job End Time: 2024-06-19T05:32:42.990Z

Number of Compare Pairs: 1

Number of Compare Pairs With Errors: 1 Number of Compare Pairs With Out-Of-Sync: 0 Number of Compare Pairs With No Out-Of-Sync: 0

Number of Compare Pairs Cancelled: 0 Number of Compare Pairs Running: 0 Job Completion Status: ERRORS Compare Pair Run ID: 325865,0,0

Compare Pair Name: ETMSP\_TIMESTAMPTEST=ETMSP\_TIMESTAMPTEST



Compare Pair Id: 1,893 Compare Pair Phase: FINISHED Compare Pair Status: FAILED Number of Rows Compared: 0 Number of Rows In Sync: 0 Number of Rows Out Of Sync: 0

# 6.4 Veridata Export and Import Utilities

You can use the Export and Import utilities, provided with the Oracle GoldenGate Veridata installation, to define portions of your configuration. Starting Oracle GoldenGate Veridata 23c, you can access these Utilities from the Web User Interface.

- Introduction to the Export and Import Utilities
- Running the Export and Import Utilities
- Configuration File Element Reference

## 6.4.1 Introduction to the Export and Import Utilities

Using the Export and Import utilities, you can create XML documents that are used to configure Oracle GoldenGate Veridata. The DTD (Document Type Definition) that governs these XML documents is stored in the *VERIDATA 23C HOME*/cli/vdt-export-client.jar file.

The Export utility helps you to either selectively or completely export the compare configuration data to an XML file. Additionally, you can use it to export configurations between different Veridata repository types using the import functionality. For example, from a SQL Server configuration to an Oracle configuration.

The Import utility allows you to configure database connections, comparison groups including compare pairs, comparison jobs, and profiles. It takes an XML document as input then creates comparison objects in Oracle GoldenGate Veridata. Typically, the XML document matches the inputs on the configuration pages in the user interface.

You should have an understanding of basic XML and its rules.

These utilities provide the following advantages:

- It can reduce the time required to define repetitive tasks
- · It allows you to create reusable configurations
- It can ensure that your test configuration mirrors the one you use for production
- Supported Configurations

## 6.4.1.1 Supported Configurations

Oracle GoldenGate Veridata import and export utilities support configuring:

- Database connections
- Comparison groups (jobs, groups, and compare pairs)
- Profiles



# 6.4.2 Running the Export and Import Utilities

The Export and Import utilities run from the *VERIDATA\_23C\_HOME*/cli/bin directory of the Oracle GoldenGate Veridata installation location. The UNIX and Linux scripts for Import and Export are vdt-import.sh and vdt-export.sh respectively.

- Using the Export Utility
- Using the Import Utility
- Processing the Configuration
- SSL Configuration for Export and Import Utilities

### 6.4.2.1 Using the Export Utility

The syntax for running the export utility is:

vdt-export [-host < host>] [-port < port>] -user < user Name> -export < fileName>

#### **Required Parameters**

- -host <host>: The host for Veridata server. The default is localhost.
- -port <port>: The port number for the Veridata Web server. The default value is 8089.
- -user <user Name>: The Oracle GoldenGate Veridata user. For example, veridata.
- -export <fileName>: The file where the Veridata config file gets exported to.

#### **Optional Parameters**

- One of these optional operations can be requested at run time:
  - -help: Prints the usage of all the Export Config file flags.
  - version v: Prints version.
  - -jobs [<job1> <job2> ... <jobN>]: Job Names space separated for exporting Configuration.
  - groups [<group1><group2> ... <groupN>]: Export all groups in the repository or add group names separated by a space, such as group1, group2, or group3.
  - -connections [<conn1> <conn2> ... <connN>]: Export all connections in the repository or add connections separated by a space, such as conn1, conn2, or conn3.
  - profiles [<profile1> <profile2> ... <profileN>]: Export all profiles in the repository or add profiles separated by a space, such as profile1, profile2, or profile3..
  - -all: All the repository configuration will be exported, takes precedence over above options. Also this is the default behavior when none of the optional flag values are entered.
  - exportPassword: Passwords for connections that are exported. By default, the Connection passwords are not exported.
  - -ssl: To communicate with Oracle GoldenGate Veridata 23c using the https protocol.

### 6.4.2.2 Using the Import Utility

The syntax for running the Import utility is:



#### **Required Parameters**

You need to specify one of the following required parameters while running the Import utility:

- -create <fileName>: Create Oracle GoldenGate Veridata config or parameter from file <fileName>.
- -delete < fileName>: Deletes Oracle GoldenGate Veridata config from file < fileName>. All named items that exist for the configuration are removed from the repository.
- -update <fileName>: Updates Veridata config from file <fileName>. New items are added to the
  repository and existing items are modified. Items existing in the repository and not listed in
  the configuration are deleted.
- -replace <fileName>: Replaces Veridata config from file <fileName>. All items listed to be replaced in the configuration are replaced as specified.
- -user <username>: Oracle GoldenGate Veridata user.
- -host <hostname>: Server host where Oracle GoldenGate Veridata is running.
- -port <port>: Server port where Oracle GoldenGate Veridata is running.
- <fileName>: Veridata config.xml or GoldenGate parameter file path to perform create, delete, update, or replace operation.

#### **Optional Parameters**

- One of these optional operations can be requested at run time:
  - -p: Properties file path.
  - -extract: Extract file path and name.
  - -ssl: To communicate with the veridata 23c using https, which is the tls protocol.
  - -help: Prints usage.
  - -version v: Prints version.

### 6.4.2.3 Processing the Configuration

The import utility first parses the *configuration.xml* file attempting to complete the entire file before aborting due to the errors. Any errors it finds are logged in the following location:

DOMAIN\_HOME/<VERIDATA\_23C\_INSTALL\_LOCATION>/cli/veridata/logs/vdt\_import.log

If it does not abort because of errors, it makes a second parsing pass, this time processing the configuration.

#### **Matching Object Names**

Database object names, such as catalogs, schema, tables, indexes, and columns will be matched according to these rules:



- The matching is case insensitive
- The hyphen (-) is considered a match to the underscore (\_) to support matching Enscribe DDL and SQL columns
- Wildcard expressions for table names and source column names match against the exact name and against the uppercase version of the name.
- Quoted names for schemas and wildcards match everything within the quotations must be matched exactly. A wildcard character within quotes is treated as an ordinary character. An example of a quoted name as it would appear in the XML is:

```
source-table=""CHAR_TYPES"*"
```

This would match CHAR TYPES, CHAR TYPES2, and CHAR TYPES NOTNULL.

Filters can either include or exclude schemas and tables. If include filters are used, at least one filter must be matched before a table can be included in a compare pair. If exclude filters are used, a table is excluded if it matches any exclude filter. Include filters can include a COLFILTER element that contains a list of columns to include or exclude. When a table matches a include filter, the include filter's COLFILTER is used to specify the columns for the generated compare pair. The schema and table name can use wildcards.

Filters can be used to exclude tables with specific names. For example, you can use the following pattern to exclude tables containing FIRST in their names:

```
<filter type="exclude" table="*FIRST"/>
```

Filters can be used to exclude views by specifying as follows in the XML:

```
<filter exclude-views="true"/>
```

For NonStop Enscribe files, file pattern filters are used. The file pattern is any valid NonStop file name pattern.

A compare pair may have a column specification with the Boolean attribute "optional".
 When this attribute is true, the column is only included in the compare pair if the source table includes the specified source column.

#### **Determining Key Columns**

The key columns are selected in the following order:

- Explicit key column definitions if they are available. In this case if source-pkey and target-pkey compare-pair element attributes are set it will generate an error.
- 2. Columns in the index specified by source-pkey and target-pkey attributes of the compare-pair element. The number of columns and all data types must match and the data types must be compatible.
- Columns in the system-selected primary key.

#### **Generating Compare Pairs**

Compare pair generation has the following characteristics:

- Generating from wild cards works the same as the user interface generation except that regular expressions can be used.
- Compare pairs are processed in the order specified in the configuration.xml file



- The compare pairs generated by a single compare pair element are generated in alphabetical order of the source table name.
- When compare pairs are generated by more than one compare pair element, the first one will be used.

As a general rule, the order of the compare pair elements should be:

- Compare pairs with specialized configuration requirements, such as user-defined keys.
- Compare pairs that match general patterns.
- Exclusions of compare pairs that would otherwise match general patterns.

### 6.4.2.4 SSL Configuration for Export and Import Utilities

The Export and Import utilities are now supported to communicate with Oracle GoldenGate Veridata server using SSL protocol.

#### To configure SSL for Export and Import Utilities:

- Check whether the SSL ceritificate is self-signed or not in the Oracle GoldenGate Veridata 23c Server. If its a self-signed certificate, then the client certificate is shipped as part of the installer under <install\_location>/config/vdtWebKeystore.p12.
- Copy the *<install location>*/config/vdtWebKeystore.p12 to *<install location>*/cli/config/veridata-23c.p12.



#### Note

To enable SSL, there is no modification required for the self-signed certificate. The Oracle GoldenGate Veridata 23c server should start with SSL.

Access the CLI utility through the SSL mode as follows:

```
./vericom.sh -j <job> -user <user> -ssl
```

Here -ssl is the option for ssl communication between the cli and Server.

If you have your own ssl certificate, replace the <install\_location>/cli/config/veridata-23c.p12 to the client certificate and rename it to veridata-23c.p12.

### 6.4.3 Configuration File Element Reference

The configuration is defined by the top level configuration element and several nested elements. Most of these elements have attributes that define their characteristics, such as the operation attribute for the configuration element or the port attribute for the connection element.

The following is the high-level element hierarchy in the configuration XML file. For more information about an element and its attributes, click the element name in the hierarchy.

#### configuration

connection conn-properties group description filter sql-partition enscribe-key compare-pair



enscribe-info
enscribe-key
sql-partition
column
excluded-column
delta-config
job
profile

- configuration
- <u>column</u>
- colfilter
- colfiltercol
- compare-pair
- connection
- conn-properties
- delta-config
- description
- enscribe-info
- enscribe-key
- <u>excluded-column</u>
- <u>expandddl</u>
- filter
- group
- job
- profile
- key-column
- profile-general
- sorting-method
- initial-compare
- confirm-out-of-sync
- param
- <u>repair</u>
- sql-partition
- table partition

The table-partition element helps in specifiying the partition name details in the Oracle GoldenGate Export and Import tools.

# 6.4.3.1 configuration

The root element is configuration.

The following elements can be nested within the configuration element:



Table 6-1 configuration Elements

Elements	Description
connection	One or more Veridata database connection definitions.
group	One or more Veridata comparison group definitions.
job	One or more comparison job definitions.
profile	One or more profile definitions.

The following attributes describe the configuration element:

Table 6-2 configuration Attributes

Attribute	Description
validation	Specifies the type of validation that is used for the configuration. The options are:
	"required" - All compare pairs must be successfully validated before any pairs are added to the repository. This is the <i>default</i> value.
	"omit-failures" - Successfully validated compare pairs are added to the repository and compare pairs that cannot be validated are ignored.
	"none" - Compare pairs are added to the repository without any validation. If this option is selected, the Oracle GoldenGate Veridata Web User Interface should be used to review and fix validation problems.
operation	Specifies how data is applied to the repository. The options are:
operation	"create" - All items listed in the configuration are new. If any item in the list exists in the repository, nothing is added. This can be used to prevent unintended modification to existing repository items. This is the default value.
	"update" - New items are added to the repository and existing items modified. Items existing in the repository and not listed in the configuration are deleted.
	"delete" - All named items in the configuration are removed from the repository.
	You can use a command line flag to override the value entered for this attribute.
wildcard	Specifies the pattern matching method that is used. The options are:
	"ggs" - Use the typical Oracle GoldenGate pattern using an asterisk (*). See the Oracle GoldenGate Veridata Web User Interface help for details on this type of matching. This is the default value.
	"regex" - Use regular expressions for matching.

#### Example

The following example adds compare pairs that can be validated and ignores those that cannot; uses regular expressions for wild carding; and uses the "create" default to adds all items as new items, adding nothing if any item already exists.



<configuration validation="omit-failures" wildcard="regex">
.
.
.</configuration>

### 6.4.3.2 column

The column element defines a set of columns to be included or excluded from the compare pair. The column element has no nested elements or text data.

The following attributes describe the column element:

Table 6-3 column attributes

Attribute	Description
source-name	A regular expression that defines a set of source column names. This value is required.
target-name	A regular expression that defines a set of target column names. It can include references to groups captured by the source-name expression.
exclude	Indicates whether or not the matched columns should be excluded from the compare pair. The options are:
	"true" - The matched columns should be excluded.
	"false" - The matched columns should be included. This is the default.
type	Indicates the type of the column. The options are:
type	"key" - The column is used as a key.
	"hash" - The column is compared using a hash value. This is the default value.
	"literal"- The column is a literal value.
format	Specifies a format to override the comparison format that would normally be used.
scale	Specifies a scale to override the default scale for the comparison.
precision	Specifies a precision to override the default precision used for the comparison.
timezone	Specifies a time zone to override the default time zone of the comparison.
optional	Indicates whether the column mapping is optional. For example, mapping will not fail if the base tables do not have the column patterns specified. Default is "false".

### 6.4.3.3 colfilter

The colfilter element defines a set of columns to be included or excluded. It is used to specify the names of the columns to use as filtering criteria.

The following element describes the colfilter element:



Table 6-4 colfilter Element

Attribute	Description
colfiltercol	Specifies a set of columns to be included or excluded.

The following attribute describes the colfilter element:

Table 6-5 colfilter Attribute

Attribute	Description
type	Specifies whether to include the columns or exclude them. The options are include or exclude; the default is include. This is a required attribute.

#### **Example**

This example excludes COL3 and COL5 for the table TABLE\_NAME from the generated compare pair.

```
<filter type="include" table="TABLE_NAME">
  <colfilter type ="exclude">
        <colfiltercol name="COL3" />
        <colfiltercol name="COL5" />
        </colfilter>
</filter>
```

### 6.4.3.4 colfiltercol

The colfiltercol element defines a set of columns to be included or excluded. It is used to specify the names of the columns to use as filtering criteria.

The following attribute describes the colfiltercol element:

Table 6-6 colfiltercol Attribute

Attribute	Description
name	A regular expression that defines a set of source column names. This is a required attribute.

# 6.4.3.5 compare-pair

The compare-pair element specifies a set of compare pair items. As in the Oracle GoldenGate Veridata Web User Interface, the compare pairs default to system mapped keys and columns.

The following elements can be nested within the compare-pair element:

Table 6-7 compare-pair Elements

Element	Description
enscribe-info	One or more sets of information used when comparing NonStop Enscribe files.



Table 6-7 (Cont.) compare-pair Elements

Element	Description
sql-partition	One or more specifications of a subset of rows within the table.
table-partition	Specifies the database table partitions.
enscribe-key	One or more specifications of a subset of records within an Enscribe file.
key-column	A set of columns to be used as the user-defined key for the comparison.
column	One or more definitions of a set of columns to be included.
excluded-column	Defines a set of columns to excluded from the a compare pair when the compare pair uses system mapped columns.
delta-config	Defines the delta processing configuration for the compare pair. The maximum is to add it once per compare pair.

The following attributes describe the compare-pair element:

Table 6-8 compare-pair Attributes

Attribute	Element
name	An expression defining the name of the compare pair. This expression can include groups captured with source-table expressions and target table group \$0.
source-table	A regular expression that defines the table or tables to be compared.  See "Regular Expression Grouping" later in this section for more detail.  The default is to match all tables.
target-table	A regular expression that defines the target tables for the comparison. This may contain references to groups captured by the source table expression. The default is \$0 for the full source table name.
source-schema	The name of the default schema for the source tables referenced for the compare pair. The default is the value specified for the group. For SQL/MP, this is the subvolume of the SQL catalog. This is not used with Enscribe files.
target-schema	The name of the default schema for the target tables referenced for the compare pair. The default is the value specified for the group. For SQL/MP, this is the subvolume of the SQL catalog. This is not used with Enscribe files.
source-catalog	The default catalog for the source tables referenced in this compare pair. For SQL/MP, this is the volume of the SQL catalog. This is not used for the following databases: Oracle (Non-PDB), DB2, Enscribe, or Teradata. For Oracle PDB, this holds the PDB name.
target-catalog	The default catalog for the target tables referenced in this compare pair. For SQL/MP, this is the volume of the SQL catalog. This is not used for the following databases: Oracle (Non PDB), DB2, Enscribe, or Teradata. For Oracle PDB, this holds the PDB name.



Table 6-8 (Cont.) compare-pair Attributes

Attribute	Element
exclude	Indicates whether or not the compare pair should be included in the group element. This can be used to remove a compare pair generated by an earlier compare pair element. The options are:
	"true" - Exclude the compare pair.
	"false" - Include the compare pair. This is the default.
source-file-pattern	The default file pattern for the source if the data source is Enscribe or SQL/MP.
target-file-pattern	The default file pattern for the target if the data target is Enscribe or SQL/MP.
source-pkey	The name of the unique index to use as the source portion of the user- specified primary key. The default is no user-specified index name.
target-pkey	The name of the unique index to use as the target portion of the user- specified primary key. The default is the value of the source-pkey.
delta-processing	Indicates whether or not delta processing is enabled for this compare pair. The options are:
	"true" - delta processing is enabled.
	"false" - delta processing is not enabled. This is the default.
profile-name	The name of the profile to use when running the compare-pair comparison.
system-key	If the compare pair has no column elements and no specified source- pkey, Oracle GoldenGate Veridata will select the most appropriate primary key or unique index to use. The options are:
	"true" - Oracle GoldenGate Veridata selects the key if it is not defined. This is the default.
	"false" - Oracle GoldenGate Veridata does not select the key.
system-columns	Indicates that the compare pair contains column elements with the type attribute set to key, so the generated compare pair will have user-defined columns for the key. The options are:
	"true" - Compare pair has key column elements. This is the default.
	"false" - Compare pair does not have key column elements.
wildcard	Specifies the pattern matching method that is used. The options are:
Wildcard	"ggs" - Use the typical Oracle GoldenGate pattern that matches an asterisk (*) to any number of characters.
	For Oracle GoldenGate HP Nonstop, the following is supported: source-table=""target-table="", and the following is not supported: source-table="TBL" target-table="". "regex" - Use regular expressions for matching.
	"default" - Use the setting for the configuration. This is the default.
is-auto	Specifies whether or not the Automatic Partitioned compare pairs need to be created in Oracle GoldenGate Veridata.  "true" - Oracle Goldengate Veridata creates the auto partitioned compare pairs. The number of compare-pairs to be created is based or the value specified in no-of-auto-partitions.
	"false" - Oracle GoldenGate Veridata does not create auto-partitioned compare pairs. This is the default.



Table 6-8 (Cont.) compare-pair Attributes

Attribute	Element
no-of-auto-partitions	Oracle GoldenGate Veridata creates the Auto Partitioned compare pairs. No of compare-pairs to be created is based on the value specified in no-of-auto-partitions.
use-source-keys	Defines key column as Source Key Columns OR Target key columns as key columns, when there are no keys selected for column mapping.
use-target-keys	Defines column values from target key to source columns, when there are no keys selected for column mapping.
use-all-columns	Defines key column values from All Columns, when there are no keys selected for column mapping.

#### **Regular Expression Grouping**

Regular expression grouping can be used to capture the parts of the source table names to be used for matching the target table name. You can do this by changing the wildcard attribute should be changed to regex. Groups to be matched are referenced as \$1, \$2, \$3 and so on. Group \$0 matches the entire source table name.

Examples of matching groups include:

- P(.\*) Matches table names that begin with P. It captures the variable portion in \$1. This matches table PROSPECTS.
- [^PV].\* Matches table names that do *not* begin with P or V. This does not match the table PROSPECTS, but does match the table REGIONS.
- ([P-R])(.\*) Matches table names starting with P, Q, or R and captures the initial letter in group \$1 and the rest of the name in group \$2. Groups are defined by parenthesis pairs. Group numbers are defined by the count of left parenthesis. Group \$1 starts at the first left parenthesis and group \$2 starts at the second parenthesis.

Captured groups ( $\$_n$ ) are then used in expressions for selecting the target tables.

#### **Example**

The following example describes the key-only compare-pair. It's source tables are defined in the "test" schema and target tables in the "other" schema. It creates a compare pair in which the source table name begins with S and target table name begins with T. For example, S\_TABLE and T\_TABLE, where S\_TABLE is a table in schema "test" and T\_TABLE is table in schema "other". It also excludes all non-key columns in the generated compare pairs.



### 6.4.3.6 connection

The connection element defines a connection to a source or target comparison database through an Oracle GoldenGate Veridata agent.

The following elements can be nested within the connection element:

Table 6-9 connection Elements

Element	Description
description	Provides a description of the connection.
conn-properties	Defines the connection properties for a connection.

The following attributes describe the connection element:

Table 6-10 connection Attributes

	Providence of the control of the con
Attribute	Description
name	A name that identifies the connection. This is a required attribute.
host	The name of the system on which the Oracle GoldenGate Veridata agent is running.
port	The port number of the system on which the agent is running.
user	The user name the agent uses to connect to the database.
password	The password the agent uses to connect to the database.
repairUser	The database user with privileges to perform repair operations. See Database Privileges for the Agent Component.
repairPassword	The password for the repairUser.
agent-timeout	The amount of time Oracle GoldenGate Veridata will wait before timing out when sending requests to the agent.
truncate-spaces	Either "true" or "false" to indicate whether or not spaces will be removed from the end of character columns. The default is "true" to truncate spaces.
fetch-size	(Oracle only) The number of rows fetched in each batch.
use-ssl	Defines using SSL communication between the Veridata Agent and the Server. The default is "true".
use-source-keys	Defines key column as Source Key Columns OR Target key columns as key columns, when there are no keys selected for column mapping.
use-all-columns	Defines key column values from All Columns, when there are no keys selected for column mapping.



#### **Example**

The following example identifies the connection named source.

## 6.4.3.7 conn-properties

The conn-properties element provides additional connection to a source or target comparison database elements.

The following attributes can be nested within the conn-properties element:

Table 6-11 conn-properties

Element	Description
datatype-name	Specifies the data type for which properties have changed.
format	Specifies the Veridata comparison format to be used for comparison.
precision	Specifies the precision to be applied to the comparison.
scale	Specifies the scale to be applied to the comparison.
timezone	Timezone name is same as in the Veridata GUI.

## 6.4.3.8 delta-config

The delta-config element defines the delta processing configuration for the specified compare pair. It can be used once per compare pair. This element can appear once or not at all depending on the type of configuration you want. When the source or target configuration specified, the corresponding column-name attribute and query element are mandatory.

The following elements describe the delta-config:

Table 6-12 delta-config Elements

Attribute	Description
source-config	Provides source side configuration for delta processing.
target-config	Provides target side configuration for delta processing.
query	Specifies the query for delta processing.



#### **Example**

This example creates a compare pair with delta processing enabled. Delta processing is enabled on COL1 of SYSMAPPING1 table for both source and target side. The SQL query is defined within the "query" tag.

### 6.4.3.9 description

The description element is free-form text that can be used to attach a description to the containing element. It has no associated attributes.

#### Example

The following example provides a description for the connection named source.

### 6.4.3.10 enscribe-info

The enscribe-info element provides additional information used to compare NonStop Enscribe records at the field level.

The following elements can be nested within the enscribe-info element:



Table 6-13 enscribe-info Elements

Element	Description
expandddl	Describes the rules that are used when applying the DDL.

The following attributes describe the enscribe-info element:

Table 6-14 enscribe-info Attributes

Attribute	Description
side	Indicates whether the information applies to the source or the target table. The options are:
	"source" to specify the source table. This is the default.
	"target" to specify the target table.
dictionary	The volume and subvolume containing the data dictionary.
record	The name of the record in the data dictionary.

# 6.4.3.11 enscribe-key

The enscribe-key element defines the key that is to be used for Enscribe files. The enscribe-key element defines a delta processing that can used in a where clause on the initial comparison query.

The following attributes describe the enscribe-key:

Table 6-15 enscribe-key Attributes

Description
A name that identifies the key. This is a required attribute.
The key that is to be used to begin reading the Enscribe file. This is a required entry.
The key of the last Enscribe record that should be read. This is a required entry.
Specifies the format of the Enscribe key. The options are:
"ascii" - The format of the key is ASCII. This is the default.
"hexadecimal" - The format of the key is hexadecimal.
Indicates whether the partition should be applied at the source database, the target database, or both databases.
Indicates whether this is the default partition. This is equivalent to the "use at run time" indicator on the UI. The default is both.



#### **Examples**

```
<enscribe-key name = "Part1" end-key = "1000" format = "hexadecimal" default = "false" side="source"/>
<enscribe-key name = "Part1" start-key = "001" format = "hexadecimal" default = "false" side="target"/>
<enscribe-key name = "Both" start-key = "001" end-key = "1000" default = "true"/>
```

### 6.4.3.12 excluded-column

The excluded-column element defines a set of columns to be excluded from a compare pair when the compare pair uses system mapped columns.

The following attribute describes the excluded-column element:

Table 6-16 excluded-column Attributes

Attribute	Description
name	A regular expression that defines a set of source column names. This is a required attribute.

## 6.4.3.13 expandddl

The expandddl element describes the rules used when applying the DDL.

The following attributes describe the expandddl element:

Table 6-17 expandddl Attributes

Attribute	Description
expandGroupArrays	Whether or not to expand group arrays. The options are:
	"true" to expand the array. This is the default.
	"false" not to expand the array.
redefined-columns	Whether or not to include redefined columns. The options are:
	"include" - Includes redefined columns
	"omit" - Leaves out redefined columns. This is the default.
resolvedups	Specifies how to resolve duplicates that result when the array is expanded. The options are:
	"append $Index$ " - Adds a unique numeric index to the end of the duplicate. This is the default.
	"append Alpha Index" - Adds an alpha character index to the end of the duplicate.
	"prependGroup" - Prefixes the name of the array group to the duplicate.
ddl-separator	The character separator for defining array output into columns. An example is the dash used in FIELDX-3, which is the third occurrence of FIELDX in the array. The options are:
	"none" - There is no separator. This is the default.
	"dash" - Use a dash (-) as the separator.
	"bracket" - Use brackets [] as the separator.
	"underscore" - Use underscore (_) as the separator.
	"double-underscore" - Use double underscore () as the separator.



Table 6-17 (Cont.) expandddl Attributes

Attribute	Description
zero-fill-length	Prepends zeros to adjust the number of the occurrence. The value is the number of digits enclosed in quotation marks. "0" is the default.
fix-long-names	Whether to fix the names that result from resolving duplicates if they exceed the max-col-name-length. The options are:
	"true" - Fix the names that exceed the maximum. This is the default.
	"false" - Do not change the names that exceed the maximum.
max-col-name-length	The maximum length allowed for a column name. The entry is a number within quotation marks. The default is "120".

# 6.4.3.14 filter

The filter element defines a set of schemas and tables to either be included or excluded.

When using include filters, at least one filter must be matched before a table can be included in a compare pair. When a table matches a include filter, the include filter's colfilter is used to specify the columns for the generated compare pair.

When using exclude filters, a table is excluded if it matches any exclude filter. Include filters can include a colfilter element, which contains a list of columns to include or exclude.

Instead of schema and table filters, NonStop platforms use file pattern filters. The file pattern is any valid NonStop platform file name pattern.

The schema and table name can use wildcards.

The following attribute describes the filter element:

Table 6-18 filter Attributes

Attribute	Description
type	Specifies either to include or exclude schemas and tables. Valid values are include or exclude.
catalog	Specifies the default catalog name.
exclude-views	Excludes all the views while generating compare pairs. Valid values are true or false. Default value is false.
schema	Specifies the schema name.
table	Specifies the table name.
file-pattern	For NonStop platforms only, specifies the file patter filter.

#### Example

When the source and target schemas have CHAR\_TYPES3, INT\_TYPE1, and INT\_TYPE2 tables, then the following filters only create compare pairs for tables CHAR\_TYPES1 and CHAR\_TYPES3. The CHAR\_TYPES2 table is excluded because of exclude filter and INT\_TYPE1 and INT\_TYPE2 are excluded because they were not part of include filter.

<group ..

<filter type="include" table="CHAR\_TYPES\*" />



# 6.4.3.15 group

The group element defines a set of compare pairs that all have the same source and target database connections. These compare pairs also have other properties in common.

The following elements can be nested within the group element.:

**Table 6-19 Group Elements** 

Element	Description
description	Provides a description of the group.
filter	One or more filter specifications, which allows table name filtering at the group level.
sql-partition	One or more specifications of a subset of rows within the table.
enscribe-key	One or more specifications of a subset of records within an Enscribe file.
compare-pair	Defines one or more compare pairs. The compare-pair elements are added to the group in the order they are specified. If the same compare pair fits the criteria of another specification in the group, the first compare pair will be used.

The following attributes describe the group element:

Table 6-20 Group Attributes

Attribute	Description
name	A name that identifies the group. This value is required.
source-conn	The name of the connection to the source database. This can reference an existing connection in the Oracle GoldenGate Veridata repository or a connection previously defined in this configuration. This attribute is required if it references an existing connection in the repository.
target-conn	The name of the connection to the target database. This can reference an existing connection in the Oracle GoldenGate Veridata repository or a connection previously defined in this configuration. This attribute is required if it references an existing connection in the repository.
source-schema	The name of the default schema for the source tables referenced in the compare pairs that make up the group.
target-schema	The name of the default schema for the target tables referenced in the compare pairs that make up the group.
source-catalog	The default catalog for the source tables referenced in this group.
target-catalog	The default catalog for the target tables referenced in this group.



Table 6-20 (Cont.) Group Attributes

Attribute	Description
validation	Specifies the type of validation that will be used for the configurations. The options are:
	"required" - All compare pairs must be successfully validated before any pairs are added to the repository.
	"omit-failures" - Successfully validated compare pairs are added to the repository and compare pairs that cannot be validated are ignored.
	"none" - Compare pairs are added to the repository without any validation. If this option is selected the Oracle GoldenGate Veridata Web User Interface should be used to review and fix validation problems.
	"default" - Use the type of validation specified for a higher level, such as the configuration element. This is the default.
source-file-pattern	The default file pattern for the source if the data source is Enscribe or SQL/MP.
target-file-pattern	The default file pattern for the target if the data target is Enscribe or SQL/MP.

# **Example**

<group name="weekly-tables" source-conn="source" target-conn"="target">
<description>

.

</description> <sql-partition>

.

</sql-partition>

<compare-pair>

.

</compare-pair>
</group>

# 6.4.3.16 job

The job element defines an Oracle GoldenGate Veridata comparison job.

The following elements can be nested within the job element:

Table 6-21 job Elements

Element	Description
description	Provides a description of the job.
group	The name of the group associated with the job. This can be a new group or a previously defined group.



The following attributes describe the job element:

Table 6-22 job Attributes

Attribute	Description
name	A name that identifies the job. This is a required attribute.
source-conn	The name of the connection to the source database. This can reference an existing connection in the Oracle GoldenGate Veridata repository or a connection previously defined in this configuration. This attribute is required if it references an existing connection in the repository.
	The job source-conn is used to override the source connection specified for the groups included in the job.
target-conn	The name of the connection to the target database. This can reference an existing connection in the Oracle GoldenGate Veridata repository or a connection previously defined in this configuration. This attribute is used to override the target connection for the groups included in the job.
profile	The default profile to use when running the job.

## **Example**

```
<job name="all-groups" profile="server-sort">
  <group name="all-tables"/>
    <group name="selected-tables"/>
  </job>
```

# 6.4.3.17 profile

The profile element defines the connection properties of a comparison job connection.

The following elements can be nested within the profile element:

Table 6-23 profile Elements

Element	Description
description	Provides a description of the profile.
profile-general	Defines the profile parameters that control the output options.
sorting-method	Defines the profile parameters that control the sorting method and memory management. The data is sorted to match keys (or a key specification) so that the correct source and target rows are compared.
initial-compare	Defines the profile parameters that control the parameters for the job that performs the initial compare step
confirm-out-of-sync	Specifies the profile parameters that control the parameters for the job that performs the confirmation step
repair	Specifies the profile parameters that control the parameters for the repair job.



The following attributes describe the profile element:

Table 6-24 profile Attributes

Attribute	Description
name	A name that identifies the profile. This is a required attribute.

# **Example**

This example creates profile named "userDefinedProfile". The parameter names like "oosformat", "sort-method" are described in the table (link for table is in another pin)

# 6.4.3.18 key-column

The key-column element defines a set of columns to be used as the user defined key for the comparison job.

The following attributes describe the key-column element:

Table 6-25 profile Attributes

Attribute	Description
source-name	A regular expression that defines a set of source column names. This value is required.
target-name	A regular expression that defines a set of target column names. It can include references to groups captured by the source-name expression.
format	Specifies a format to override the comparison format that would normally be used.
scale	Specifies a scale to override the default scale for the comparison.
precision	Specifies a precision to override the default precision used for the comparison.
timezone	Specifies a time zone to override the default time zone of the comparison.



# 6.4.3.19 profile-general

The profile-general element provides parameters to control the output options.

The data is sorted to match keys (or a key specification) so that the correct source and target rows are compared.

The following elements can be nested within the profile-general element:

Table 6-26 profile-general Element

Element	Description
param	Defines the parameter to change for the profile.

# 6.4.3.20 sorting-method

The sorting-method element provides parameters for sorting method and memory management. The data is sorted to match keys (or a key specification) so that the correct source and target rows are compared.

The following elements can be nested within the sorting-method element:

Table 6-27 sorting-method Element

Element	Description
param	Defines the parameter to change for the profile.

# 6.4.3.21 initial-compare

The initial-compare element provides parameters for the process that performs the initial compare step.

The following elements can be nested within the initial-compare element:

Table 6-28 initial-compare Element

Element	Description
param	Defines the parameter to change for the profile.

# 6.4.3.22 confirm-out-of-sync

The confirm-out-of-sync element provides parameters for the process that performs the confirmation step.

The following elements can be nested within the confirm-out-of-sync element:



Table 6-29 confirm-out-of-sync Element

Element	Description
param	Defines the parameter to change for the profile.

# 6.4.3.23 param

The param element defines the parameters that are used for configuring profile options.

The following attributes describe the repair element:

Table 6-30 param Attributes

Attribute	Description
name	The name of the parameter. This is a required attribute.
value	The value of the parameter

# 6.4.3.24 repair

The repair element provides parameters for the repair process.

The following elements can be nested within the repair element:

Table 6-31 repair Element

Element	Description
param	Defines the parameters that are used to configure the profile options.

# 6.4.3.25 sql-partition

The sql-partition element defines a boolean SQL expression that can be used in a where clause in the initial comparison query.

The following attributes describe the sql-partition element:

Table 6-32 sql-partition Attributes

Attribute	Description
name	A name that identifies the partition. This is a required attribute.
side	Indicates whether the partition should be applied at the source database, the target database, or both databases. The default is "both".
default	Indicates whether this is the default partition. This is equivalent to the "use at run time" indicator on the UI. The default is "false".



Table 6-32 (Cont.) sql-partition Attributes

Attribute	Description
type	Distinguishes between Manual and Automatic Row Partition. When type is set to sql, it defines Manual Partition and type is set to auto, defines Automatic Row Partition. This attribute is displayed during export of compare-pairs using Oracle GoldenGate Veridata Export tool.

#### **Example**

```
<sql-partition name="replicate" default="true" side="source" type="sql">
<![CDATA[ replicated='false']]>
</sql-partition>
<sql-partition name="replicate" default="true" side="source" type="sql">
<![CDATA[ replicated='true']]>
</sql-partition>
<sql-partition name="AutoPartition0" default="true" side="source" type="auto>
<![CDATA[2,0]]>
</sql-partition>
```

# 6.4.3.26 table partition

The table-partition element helps in specifiying the partition name details in the Oracle GoldenGate Export and Import tools.

For more information, see Mapping Database Table Partitions for Manual and Automatic Row Partitioning. To create compare-pairs using existing Database Table partitions, include the following lines in the XML when you execute the Import and Export tools:

```
<configuration operation="create" validation="required">
<group name="oracle_oracle_grp" source-conn="oracle_src" target-conn="oracle_tgt">
<compare-pair name="SALES_SRC_Q1_2006=SALES_GT_SALES_Q1_2006_AutoPartition0"
source-table="SALES_SRC" target-table="SALES_TGT" source-schema="SYSTEM" target-schema="SYSTEM" use-source-keys="true" use-target-keys="true" use-all-columns="true">
<table-partition name = "SALES_Q1_2006" default="true" side="target"/>
<table-partition name = "SALES_Q1_2006" default="true" side="target"/>
</compare-pair>
</compare-pair>
</configuration>
```

# 6.5 Server Parameters

This section provides a consolidated overview of key server configuration parameters used in Oracle GoldenGate Veridata. These parameters, typically defined in the <code>veridata.cfg</code> configuration file, impacts the server behavior, including performance tuning, memory management, sorting, email notifications, reporting, repair operations, and handling of out-of-sync (OOS) records. Understanding and tuning these parameters appropriately improves system performance, optimizes comparison jobs, and ensures reliable operation of Veridata in diverse deployment environments.

Server configuration in Oracle GoldenGate Veridata can be modified through parameters defined in the <code>veridata.cfg</code> file, <code>located</code> at <code>DOMAIN\_HOME/config/veridata/veridata.cfg</code>.

Refer How to Set a Parameter for details on how to set a parameter.



#### Note

Modification in these parameters will require server restart.

- Overview of the Server Memory
- Estimating Memory Usage
- How to Set a Parameter
- Parameter Descriptions

# 6.5.1 Overview of the Server Memory

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

- **Server memory for basic operation**: This is the size of the virtual memory that the Veridata server and web components need to operate. It stores object pools, database access libraries, and other information. This is typically about 200 MB.
- **Sort memory**: This is the memory that is used when server-side sorting is used. The virtual memory for sorting is allocated for the entire comparison, not per thread. The rows are read from the agent and submitted to be sorted. The sorting occurs in a thread that is separate from the thread that reads from the agent, and the sort may use more threads to work in parallel. After all the rows from the agent are submitted to the sort process, the server process retrieves the sorted rows from the sort for comparison.
- Row hash queue memory: This is the memory that buffers data between the agent processes, the sort process, and the server process. A comparison that uses database sorting requires a single queue each for the source and target. Each queue has a capacity of 20 MB. The memory usage by the queues is affected by the relative speed of the comparison and by the data coming from the agent. The relative speed between the two agents also affects the memory usage. A larger differential in speed increases the amount of memory that is used, because the queue needs to buffer the data.
- MOOS queue memory: This is the memory that holds potentially out-of-sync records between the initial comparison and confirmation steps of a comparison. The size of the MOOS queue is limited to 50K of records. Memory usage is also dependent on the width of each record.
- **IPC buffer memory**: This is the memory that is used to exchange messages between the server and the agent.
- Scratch runtime transient memory: This is virtual memory space.

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

- System physical memory
- Available memory in swap
- Java boot option -Xmx maximum memory setting

# 6.5.2 Estimating Memory Usage

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



When a comparison is run, two buffers are allocated from the reserved sort memory. Each of these is equal to the size specified as Maximum Memory Usage (MB). Refer server.max sort memory.

## To Estimate the Amount of Memory Used per Row

For more information about the estimating the amount of memory used per row, see <u>Disk and Memory Requirements for the Server Component</u>.

# 6.5.3 How to Set a Parameter

To set a parameter, edit its entry in the <code>veridata.cfg</code> file. This file is stored in the <code>DOMAIN\_HOME/config/veridata</code> directory within the Oracle GoldenGate Veridata Server installation directory.

Open a Oracle service request before changing these parameters.

# 6.5.4 Parameter Descriptions

Paramete r Name	Syntax / Example	Possible Values	Default Value	Description
database. hash	database.hash = false	true/false	false	Defines whether data hashing is done in the database. Supported only for Oracle.
max_lob_ key_len	max_lob_key_len = 1	1 to 8000 (bytes)	0 (A value of 0 disables the feature)	Sets max length of LOB data types used as keys (in bytes). Supported for Postgres' citext.
	server.convert_em pty_to_null = true	true, false	true	If true, treats empty string and NULL as in-sync.
server.ma x_concurr ent_comp arison_thr eads	server.max_concur rent_comparison_t hreads = 100		4	This parameter sets the maximum number of concurrent comparisons that can be executed. You can lower this number to reduce the impact of the server on your system. When this limit is reached, no new comparisons, starts until an active comparison completes.
server.me mory_ma pped_sort _directory	server.memory_m apped_sort_direct ory = /u01/tmp2	Directory path	-	This parameter specifies the directory path where .map files are generated. By default, it uses temp directory of the operating system. If your operating system temp directory is less than 50GB, then you should consider specifying a larger directory.
	server.meta_sessi on_handle_timeou t = 600	Numeric (seconds)	900	Timeout (in seconds) for metasession handle.
	server.veridata_da ta = /u01/veridata/ reports	Directory path	veridata/ reports	Output directory for Veridata reports.
socket.tim eout.mins	s = 0	Numeric (minutes)	0	Timeout for inactive sockets (in minutes).



Paramete r Name	Syntax / Example	Possible Values	Default Value	Description
server.use Ssl	server.useSsI = true	true, false	false	This parameter specifies whether SSL is enabled for communication from Veridata Server to Veridata Agents.
truncate_s paces_len	truncate_spaces_I en = 0	0, 1	1	Use the truncate_spaces_len parameter to control how trailing spaces in a String or Binary column are truncated. Space (U+0020) and Ideographic Space (U+3000) are the truncate targets. You can disable the truncate feature with the Truncate Trailing Spaces When Comparing Values parameter of the Connection Settings tab in the Oracle GoldenGate Veridata UI of the connection configuration. This parameter also truncates the column padding character if the padding character is one of the target spaces. If the value equals 0, then all spaces are trimmed. For example, data = "". If the value equals 1, then at least one space gets retained. For example data = "".
jvm.memo ry.xms	jvm.memory.xms = <min_heap_size></min_heap_size>	Memory value (e.g., 8g)	-	Minimum JVM heap size. This can be modified by editing <veridata_home>/config/ oggvdt_cainput.properties. See section "What should be the optimum memory for server?"</veridata_home>
jvm.memo ry.xmx	jvm.memory.xmx = <max_heap_size></max_heap_size>	Memory value (e.g., 28g)	-	Maximum JVM heap size. This can be modified by editing <veridata_home>/config/ oggvdt_cainput.properties. See section "What should be the optimum memory for server?"</veridata_home>



Paramete r Name	Syntax / Example	Possible Values	Default Value	Description
server.ma x_sort_m emory	server.max_sort_ memory = 1000M	Memory value (e.g., 1000M)	size based on the available virtual memory.	memory that is available to all running comparisons that use server-side sorting. The value of this parameter varies depending on the value of server.mapped_sort_buffers. server.max_sort_memory {default   number{M   m}} default is illustrated in the follows: When server.mapped_sort_buffers is set to true, the default value is 2 GB. The maximum number supposed to enter is the maximum size of the memory mapped file. When server.mapped_sort_buffers is set to false, the default value is the JVM available heap size minus 200 MB, which is allocated for basic tasks. The maximum number supposed to enter is the value of java -Xmx.
server.ma pped_sort _buffers	server.mapped_so rt_buffers = true	true, false	true, false	Indicates whether sort buffers are allocated as a memory mapped file or allocated on the JVM heap.
server.nu mber_sort _threads	server.number_sor t_threads = <number></number>	Max of 4 or 1/4 of CPU cores	4	This parameter specifies the number of threads used to sort input buffers from the Veridata Agent.
mail.enabl ed	mail.enabled = false	true, false	false	Enables/disables email notifications.
mail.alert. only	mail.alert.only = true	true, false	true	If true, send only alert emails (not completion).
mail.smtp. server	mail.smtp.server = smtp.host.com	SMTP server hostname	-	SMTP server hostname.
mail.smtp. port	mail.smtp.port = 25	Numeric port number	-	SMTP server port.
mail.smtp. connectio n.ssl.tls	mail.smtp.connecti on.ssl.tls = true	true, false	false	Enables SSL/TLS for mail server connection.
mail.from	mail.from = john@mail.com	Valid email address	-	Sender email address.
mail.to	mail.to = john@mail.com,m ary@mail.com	Comma- separated email addresses	-	Recipient email addresses (commaseparated).
server.enc ryption	server.encryption = false	true, false	false	Enables encryption for comparison and repair reports.
server.enc ryption.bit s	server.encryption.b its = 128	128, 192, 256	128	Encryption strength in bits.



Paramete r Name	Syntax / Example	Possible Values	Default Value	Description
repair.ora cle.tag.en able	repair.oracle.tag.e nable = false	true, false	true	Enables tagging of repair sessions in Oracle redo logs.
repair.ora cle.tag	repair.oracle.tag = 00	Value can be up to 2000 hexadeci mal digits (0-9A-F) or the plus sign (+).	00	Hexadecimal tag used in Oracle redo logs.
coos.join. strategy	coos.join.strategy = nokey	nokey, always, never	nokey	nokey - Use COOS Join strategy only when Use all columns as Key columns (there is no primary or unique constraint on the table) is used to select key columns for the compare pair. All columns as key columns in the compare pair exception XML, UDT and Lobs. always - Use COOS Join always for COOS step. never - Don't use COOS Join for COOS Step.
skiprows.e nabled	skiprows.enabled = true	true, false	true	Skips rows with invalid numbers to allow comparison to continue.

# 6.6 Agent Parameters - General

This topic defines the following configurable parameters for your Oracle GoldenGate Veridata Agent:

- compare.xmldatatype.format
- coos.batch.fetch
- · database.characterSet
- database.transaction.isolation
- pool.maxIdleTime
- pool.checkInterval
- pool.maxIdle
- pool.maxSize
- pool.maxStatements
- server.port
- <u>rowscn</u>
- zlib.buffer.flush.size



# 6.6.1 compare.xmldatatype.format

To use the INDENT/NO INDENT functionality of XMLSerialize, update the compare.xmldatatype.format to either true or false. By default, the comparison is done with INDENT. To select NO INDENT set compare.xmldatatype.format=false.



#### (i) Note

The compare.xmldatatype.format parameter is only for Oracle Agent.

#### **Syntax**

compare.xmldatatype.format=true

#### **Default**

# 6.6.2 coos.batch.fetch

Parameter to fetch COOS data in batches, instead of running individual gueries, in turn accelerating the performance.

#### **Default Value**

false

# 6.6.3 database.characterSet

The database characterSet parameter is used for overriding the source database character for

The parameter value should be the name of the character set used to encode the character data (CHAR, VARCHAR2, CLOB and LONG).

This parameter should match the SOURCECHARSET OVERRIDE in the replicat parameter file at the target database.

This is supported only for Oracle databases.

# 6.6.4 database.transaction.isolation

The database transaction isolation property controls the transaction isolation level used during initial compare.

The default value for Sybase, DB2, SQL Server and Teradata is READ\_UNCOMMITTED.

The only value supported for Oracle is READ\_COMMITTED.

SQL Server versions 2005 above also support the value SNAPSHOT which requires that ALLOW\_SNAPSHOT\_ISOLATION is enabled in the database.

Confirm out of sync always uses the READ\_COMMITTED transaction isolation level.

#### **Default Value**

database.transaction.isolation=READ UNCOMMITTED



# 6.6.5 pool.maxIdleTime

Timeout for idle connections in the connection pool. The value is in seconds.

#### **Default Value**

300

# 6.6.6 pool.checkInterval

Time interval to check the timeouts for idle connections.

#### **Default Value**

1/4th of pool.maxIdleTime

# 6.6.7 pool.maxIdle

Max number of idle database connections present in pool.

#### **Default Value**

20

# 6.6.8 pool.maxSize

Database connection pool maintained by Veridata. It defines the max number of database connections held by this pool. There will a separate pool maintained by Veridata for each user and for each PDB in-case of Container DB.

## **Default Value**

20

# 6.6.9 pool.maxStatements

The number of statements cached per connection. The default value is 20.

#### **Default Value**

20

# 6.6.10 server.port

The server.port property is the port where the Oracle GoldenGate Veridata Agent listens for connection requests.

#### **Syntax**

server.port=Port Number

#### For example:

server.port=7862



# 6.6.11 rowscn

Use this property if you want to skip initial delta comparison. Only the rows greater than the given SCN value will be compared.

This property is enabled only for Oracle Agent.



#### (i) Note

This property works either with or without delta. However if you enable delta, then ensure that ORA\_ROWSCN is used as delta column which is the default delta column for Oracle tables.

#### **Syntax**

rowscn=<scn number>

#### **Default Value**

None

# 6.6.12 zlib.buffer.flush.size

The zlib.buffer.flush.size property is used to decide the extra flush() call.

Decrease the value to a lower value (for example, 700000) if the agent appears to be in hanged state and the CPU usage for agent process is about 100%.

The value should be between 10000 and 2000000.

#### **Default Value**

1000000

# 6.7 Agent Parameters - Connections



#### Note

For all the connections, the downloaded Driver can be added to the default location: server.driversLocation=\$ORACLE\_HOME/agent/drivers

Or to a custom location that you have to specify in agent.properties. Example:

server.driversLocation=/custom/driver/directory

This information is also available in the agent.properties.sample.

- # The server.driversLocation property is the directory
- # containing the JDBC driver jar file(s). The JDBC drivers
- # included with the Veridata agent will be retrieved from the
- # standard locations.
- # The path is relative to the Veridata agent
- # deployment directory. OR An absolute path to the directory containing



# drivers can be specified. server.driversLocation=/scratch/nextgen/vdt23c/agent/drivers

- IBM Db2 for i
- IBM Db2 for LUW
- IBM Db2 for z/OS
- Apache Hive
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle Database
- PostgreSQL
- Sybase Adaptive Server Enterprise
- Teradata Vantage

# 6.7.1 IBM Db2 for i

## Db2 i sample database connection URL

LocationName is usually the i server name.

database.url=jdbc:veridata:db2://DB2HOST:446;LocationName=<DATABASE\_NAME>server.jdbcDriver=vddb2-5.1.4.jar

## Settings for using the JTOpen driver for Db2 i

database.url=jdbc:as400:DB2HOST server.jdbcDriver=jt400Native.jar

## Settings for using the IBM Native driver for Db2 i

database.url=jdbc:db2i:DB2HOST:portNumber=446;preFetch=true server.jdbcDriver=db2\_classes.jar

# 6.7.2 IBM Db2 for LUW

#### Db2 LUW sample database connection URL

database.url=jdbc:veridata:db2://localhost:50000;DatabaseName=sample server.jdbcDriver=vddb2-5.1.4.jar

# 6.7.3 IBM Db2 for z/OS

## DB2 z/OS sample database connection URL

 $\label{local-data-data-data} database.url=jdbc:veridata:db2://localhost:447;LocationName=DB2 \\ server.jdbcDriver=vddb2-5.1.4.jar$ 

# 6.7.4 Apache Hive

Hive sample database connection URL for Apache Hive2 JDBC Driver



database.url=jdbc:hive2://localhost:10000 server.jdbcDriver=hive-jdbc-3.1.2-standalone.jar

# 6.7.5 MariaDB

#### MariaDB sample JDBC URL with SSL

database.url=jdbc:mariadb://localhost:3306/ database?sslMode=verify-full&serverSslCert=/path/to/cert.pem server.jdbcDriver=mariadb-java-client-3.5.2.jar

#### MariaDB sample JDBC URL without SSL

database.url=jdbc:mariadb://localhost:3306/database server.jdbcDriver=mariadb-java-client-3.5.2.jar

You can download the Driver from https://mariadb.com/kb/en/installing-mariadb-connectorj/.

# 6.7.6 Microsoft SQL Server

#### SQL Server sample database connection URL for SSL/TLS

database.url=jdbc:sqlserver://localhost:1443;databaseName=<dbName>;encrypt=false;trustServerCertificate=false; server.jdbcDriver=mssql-jdbc-11.2.0.jre17.jar

### SQL Server database connection URL

database.url=jdbc:veridata:sqlserver://localhost:1433 server.jdbcDriver=vdsqlserver-6.0.0.jar

# 6.7.7 MySQL

## MySQL sample database connection URL

database.url=jdbc:mysql://localhost:3306/<db\_name>?serverTimezone=UTC& zeroDateTimeBehavior=CONVERT\_TO\_NULL&sessionVariables=sql\_mode='PAD\_CHAR\_TO\_FULL\_LENGTH' server.jdbcDriver=mysql-connector-j-8.3.0.jar

# MySQL sample JDBC URL for MySQL Server Authentication via server certificate

(For MySQL version 8.0.12 and earlier):

 $database.url=jdbc:mysql://localhost:3306?useSSL=true\&verifyServerCertificate=true\\server.jdbcDriver=mysql-connector-j-8.3.0.jar$ 

#### For MySQL version 8.0.13 and later:

database.url=jdbc:mysql://localhost:3306?sslMode=<VERIFY\_CA or VERIFY\_IDENTITY> server.jdbcDriver=mysql-connector-j-8.3.0.jar

# 6.7.8 Oracle Database

#### Oracle sample database connection URL

database.url=jdbc:oracle:thin:@localhost:1521:orcl database.url=jdbc:oracle:thin:@localhost:1521/PDB\_service\_name server.jdbcDriver=ojdbc11-23.2.0.0.jar

#### Oracle OCI bequeath database connection URL



#### The OCI libraries must be available and the JDBC driver must match the OCI libraries.

database.url=jdbc:oracle:oci:@FREE server.jdbcDriver=ojdbc11-23.2.0.0.jar

# Oracle sample database connection URL for SSL/TLS

database.url=jdbc:oracle:thin:@tcps://<host>:<port>/<service name>?wallet\_location=<wallet directory path> database.url=jdbc:oracle:thin:@tcps://localhost:2484/service\_name?wallet\_location=/path/WALLET server.jdbcDriver=ojdbc11-23.2.0.0.jar

# 6.7.9 PostgreSQL

# Postgresql sample database connection URL for native driver

database.url=jdbc:postgresql://localhost:5432/target server.jdbcDriver=postgresql-42.7.4.jar

## Postgresql sample database connection URL for SSL/TLS

database.url=jdbc:postgresql://<host>:5432/postgres?sslmode=verify-ca&sslrootcert=<crt\_path>/<crt\_filename>server.jdbcDriver=postgresql-42.7.4.jar

# 6.7.10 Sybase Adaptive Server Enterprise

#### Sybase database connection URL

JDBCBehavior=0 is required to repair Sybase UNITEXT columns.

 $\label{lem:database.url=jdbc:veridata:sybase://localhost:5000; Application Name=Veri Agent; MaxPooled Statements=20; JDBCB ehavior=0 \\ server.jdbcDriver=vdsybase-5.1.4. jar$ 

# 6.7.11 Teradata Vantage

# Teradata database connection URL

database.url=jdbc:teradata://localhost/DBS\_PORT=1025,CHARSET=UTF8 server.jdbcDriver=terajdbc4.jar

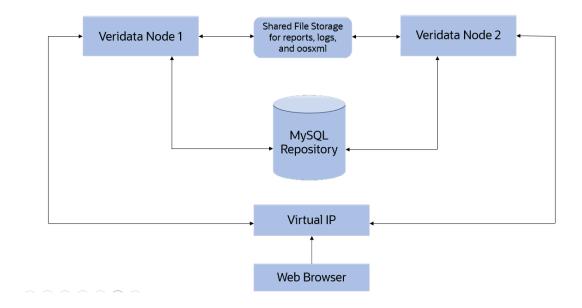
# Configure

High Availability

# 7.1 High Availability

High Availability (HA) refers to the ability of a system, application, or services to remain operational and accessible with minimal downtime. It ensures that an application continues to function even in the event of failures, such as hardware crashes, network issues, or software bugs. You can configure High Availability for the Oracle GoldenGate Veridata Server and Agent in several ways. This article explains one such method.

Figure 7-1 High Availability for Veridata Server



To establish High Availability in Oracle GoldenGate Veridata 23c:

- 1. Deploy Veridata Server on the nodes that share a Virtual IP between them.
- 2. Connect to Veridata Server using a VIP to establish an Active-Passive setup. In such a set up, even when one Server node goes down, the other node serves the purpose.

Ensure that there is a shared file system for storing the logs, reports, and oosxml files from the active server, so that when the Standby server turns active it can have access to all the files.

For the MySQL repository, ensure that the repository is highly available.

- Setting Up the High Availability for Oracle GoldenGate Veridata Server 23c
- Configuring a Shared File Path



Setting up High Availability for Oracle GoldenGate Veridata Agent

# 7.1.1 Setting Up the High Availability for Oracle GoldenGate Veridata Server 23c

To set up HA for Veridata Server 23c:

- Install Veridata Server on all the nodes which share a common Virtual IP between them.
- 2. Ensure that all nodes use the same port for Veridata Server.
- 3. Edit the repository.url property in <veridata\_home>/config/oggvdt\_cainput.properties file, to ensure that both the servers point to the same shared repository.
- 4. For more information about Oracle GoldenGate Veridata installation, see <u>Installing and</u> Running Oracle GoldenGate Veridata.

# 7.1.2 Configuring a Shared File Path

- Reports
- Wallet

# 7.1.2.1 Reports

Ensure that Oracle GoldenGate Veridata reports are stored in a shared file system which can be accessed by all the nodes configured using the VIP. Shared storage path can be given in veridata.cfg as follows:

server.veridata\_data property must be updated to the new shared storage path:

server.veridata\_data new\_path/veridata/reports

# 7.1.2.2 Wallet

The wallet that stores the passwords of connections or users must also be stored in the shared file path.

To configure wallet to shared path:

- Nagivate to the config in the Veridata installation directory.
- 2. Open jps-config-jse.xml and update the shared path in the below xml tag in location attribute. <serviceInstance name="credstore" provider="credstoressp" location=". ">

For example: <serviceInstance name="credstore" provider="credstoressp" location="/u01/config ">

In the Veridata 23c server, the wallet path is hard-coded in the JPS configuration file.

To change it to a user-preferred directory path, follow the steps below:

- Set the environment variable to override the wallet path to export CREDENTIAL\_LOCATION=<path>
- 2. Restart the server after setting the environment variable.



# 7.1.3 Setting up High Availability for Oracle GoldenGate Veridata Agent

In Oracle GoldenGate Veridata, the agent is deployed as a Java Standalone application. To establish agent HA, deploy the agent on nodes sharing a Virtual IP (VIP) between them and then connect the Veridata Server to agent via VIP.

Ensure that agents running on different nodes are using the same port number. The agent.propreties from node1 must be copied to node2 as well and must be kept in sync for the changes.

Veridata Agent HA Active/Passive

Figure 7-2 High Availibility for Oracle GoldenGate Veridata Agent

# Agent on Node1 Oggvdtagt-IP1 Oggvdtagt-VIP Veridata Server

# Secure

- Configuring Workflow for Two-Way SSL in Oracle GoldenGate Veridata 23c
- Configuring Two-Way SSL for the NSK C-Agent on the Veridata Server
- Configuring Oracle GoldenGate Veridata Agent Using Kerberos to Connect to Oracle Database
- Configuring Oracle GoldenGate Veridata Agent Using Kerberos to Connect to Hive
- Connecting Oracle GoldenGate Veridata to SSL-Enabled Oracle Database
- Connecting Oracle GoldenGate Veridata to SSL-Enabled MySQL Database
- Connecting Oracle GoldenGate Veridata to SSL-Enabled SQL Server Database
- Connecting Oracle GoldenGate Veridata to SSL-Enabled PostgreSQL Database

# 8.1 Configuring Workflow for Two-Way SSL in Oracle GoldenGate Veridata 23c

#### **Prerequisites**

- Ensure PATH environment variable contains the path to JDK 17 bin directory, where keytool is located.
- Keystore is also known as Identity Store in older releases.
- For HP NonStop, see <u>Configuring Two-Way SSL for the NSK C-Agent on the Veridata Server</u>.

#### This topic comprises the following:

- Enabling SSL in the Agent Properties File
- Generating Agent Keystore and Certificate
- Generating Server Keystore and Certificate
- Importing Agent Certificate to Server Truststore
- Saving Server Keystore/Truststore Passwords to Server Wallet
- Importing Server Certificate to Agent Truststore
- Saving Agent Keystore/Truststore Passwords to Agent Wallet
- Creating an Agent Connection in UI

# 8.1.1 Enabling SSL in the Agent Properties File

## To enable SSL in the Agent properties file:

**1.** Open the <*AGENT\_DEPLOY\_LOCATION*>/<*AGENT PROPERTIES*> file. The default <*AGENT PROPERTIES*> would be agent.properties.



Find the entry server.useSsl, set it to true and save the changes.

# 8.1.2 Generating Agent Keystore and Certificate

To generate agent keystore and certificate file:

- Go to <AGENT\_DEPLOY\_LOCATION>/config/certs directory.
- Build Agent Keystore with the following command, you will also need the same unlock for later step.

keytool -genkeypair -keyalg RSA -keystore vdtAgentKeystore.p12 -storepass <unlock-password>

3. Export Agent Keystore to a certificate with following command:

keytool -exportcert -keystore vdtAgentKeystore.p12 -storepass <unlock-password> -file vdtAgent.crt

# 8.1.3 Generating Server Keystore and Certificate

## Use the following keytool commands to generate the server keystore and certificate:

- Go to <Server\_installation\_location>/config directory.
- Build Server Keystore with the following command:

keytool -genkeypair -keyalg RSA -keystore vdtServerKeystore.p12 -storepass <unlock-password>

Export Server Keystore to a certificate with the following command:

keytool -exportcert -keystore vdtServerKeystore.p12 -storepass <unlock-password> -file vdtServer.crt

# 8.1.4 Importing Agent Certificate to Server Truststore

To import agent certificate to Server truststore:

- Go to <Server\_installation\_location>/config directory, and copy <AGENT\_DEPLOY\_LOCATION>/config/ certs/vdtAgent.crt to this directory.
- 2. Run the following command to create a server truststore and import agent certificate into this truststore.



#### Note

When importing multiple agent certificates into a server truststore, assign a unique -alias value to each agent certificate.

keytool -importcert -file vdtAgent.crt -alias vdtAgent.crt.<unique-id> -keystore vdtServerTruststore.p12 storepass <unlock-password>

Delete vdtAgent.crt in < Server\_installation\_location > /config directory.



# 8.1.5 Saving Server Keystore/Truststore Passwords to Server Wallet

Run the following script configure\_server\_ssl.sh under <Server\_installation\_location>/config directory:

./configure\_server\_ssl.sh

OGGV-80056: Copyright (c) 2013, 2024, Oracle and/or its affiliates. All rights reserved.

OGGV-80057: Veridata Server SSL Configuration Utility

OGGV-80058: Notes:

OGGV-80059: OGGV-80059: This utility allows Veridata to access keystore and truststore. When entering the passwords below,

unlock password should match the one used in keytool -storepass option

[OGGV-80060: Enter Server Keystore unlock password:] [OGGV-80062: Enter Server Truststore unlock password:]

# 8.1.6 Importing Server Certificate to Agent Truststore

To import Server certificate to agent truststore:

- Go to <AGENT DEPLOY LOCATION>/config/certs directory, and copy <Server installation location>/ config/vdtServer.crt to this directory.
- Run the following command to create an agent truststore and import server certificate into this truststore.



#### (i) Note

When importing multiple Server certificates into an agent truststore, assign a unique -alias value to each Server certificate. This is less common, because there is typically only one Oracle GoldenGate Veridata server.

keytool -importcert -file vdtServer.crt -alias vdtServer.crt.<unique-id> -keystore vdtAgentTruststore.p12 storepass <unlock-password>

Delete vdtServer.crt in <aGENT\_DEPLOY\_LOCATION>/config/certs directory.

# 8.1.7 Saving Agent Keystore/Truststore Passwords to Agent Wallet

parameter AgentID is the name of the agent properties file, without the .properties extension.:

./configure\_agent\_ssl.sh AgentID

2024-08-30 11:21:25.782 TRACE OGGV-80018 Wallet messages are installed correctly.

OGGV-80028: Copyright (c) 2013, 2024, Oracle and/or its affiliates. All rights reserved.

OGGV-80029: Veridata Agent SSL Configuration Utility

OGGV-80030: Notes:

OGGV-80031: OGGV-80031: This utility allows Veridata to access keystore and truststore. When entering the passwords

below, unlock password should match the one used in keytool -storepass option.

[OGGV-80022: Enter Agent Keystore Store unlock password:]

[OGGV-80024: Enter Agent Trust Store unlock password:]

OGGV-80037: SSL Configuration of Veridata Agent is successful.

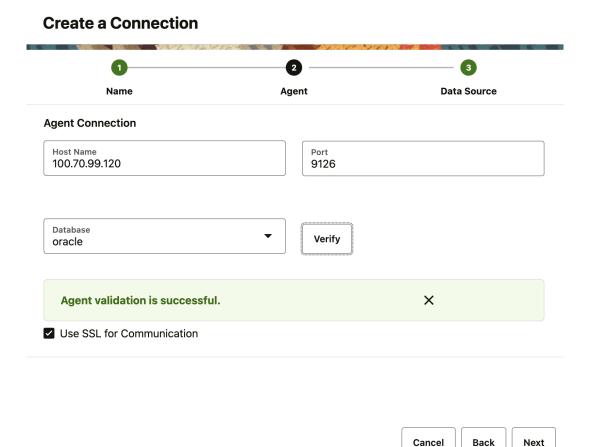
Delete vdtServer.crt in <a href="https://documents.com/delete-vdtServer.crt">AGENT\_DEPLOY\_LOCATION</a>/config/certs directory.



# 8.1.8 Creating an Agent Connection in UI

- Log in to the Veridata website, navigate to the Connections page, and click Create on the right side of the page.
- 2. Enter the required connection name and agent host machine address/port. Check the Use SSL for communication checkbox to enable SSL for the connection, and click Verify to verify the connection:

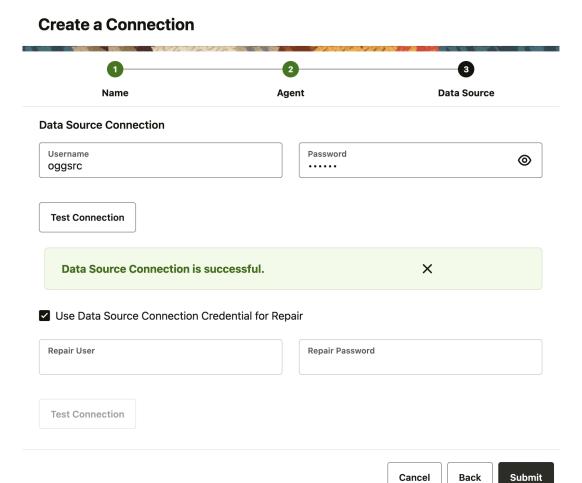
Figure 8-1 Create a Connection



3. Enter the database username and password, then click the **Test Connection** to validate the credentials. If different credentials are needed for repair, uncheck the **Use Data Source Connection Credential for Repair** checkbox and enter the repair credentials.



Figure 8-2 Data Source Connection



Click Submit to save the connection.

# 8.2 Configuring Two-Way SSL for the NSK C-Agent on the Veridata Server

#### **Prerequisites for Veridata Server**

- Ensure PATH environment variable contains the path to JDK 17 bin directory, where keytool is located.
- Keystore is also known as Identity Store in older releases

# SSL Configuration on Veridata Server

#### **Generate Server Keystore and Certificate**

If the Oracle GoldenGate Veridata server keystore has not been created, then use the keytool command as follows to generate the server keystore.

**1.** Go to *<Server installation location>/*config directory.



Build Server Keystore with the following command:

keytool -genkeypair -keyalg RSA -keystore vdtServerKeystore.p12 -storepass <unlock-password>

Export Server Keystore to a certificate with the following command

keytool -exportcert -keystore vdtServerKeystore.p12 -storepass <unlock-password> -file vdtServer.crt

Convert the certificate from CRT format to PEM format with the following command:

openssl x509 -in vdtServer.crt -outform PEM -out vdtServer.pem -inform DER

#### Import NSK C-Agent Certificate to Server Truststore

- Go to <Server\_installation\_location>/config directory, and copy the NSK C-agent certificate to this directory.
- 2. Run the following command to create a server truststore and import the NSK C-agent certificate into this truststore:

keytool -importcert -file <nsk-agent-certificate-file> -alias vdtAgent.crt.<unique-id> -keystore vdtServerTruststore.p12 -storepass <unlock-password>



#### (i) Note

When importing multiple nsk c-agent certificates into a server truststore, assign a unique -alias value to each nsk c-agent certificate.

Delete the nsk c-agent certificate in <Server\_installation\_location>/config directory

# Save Server Keystore/ Truststore passwords to Server Wallet

Run the following script configure\_server\_ssl.sh under <Server\_installation\_location>/config directory.

./configure\_server\_ssl.sh

OGGV-80056: Copyright (c) 2013, 2024, Oracle and/or its affiliates. All rights reserved.

OGGV-80057: Veridata Server SSL Configuration Utility

OGGV-80058: Notes:

OGGV-80059: This utility allows Veridata to access keystore and truststore. When entering the passwords below, unlock password should match the one used in keytool -storepass option.

[OGGV-80060: Enter Server Keystore unlock password:] OGGV-80062: Enter Server Truststore unlock password:]

#### SSL Configuration on NSK C-agent

The SSLCA file is the Certificate Authority file distributed with the NSK C-agent. The **SSLCERT** file is the default certificate for the NSK C-agent.

For a successful two-way SSL configuration, the NSK C-agent's certificate (for example, SSLCERT) must be:

- Imported into the Veridata Server's truststore.
- Appended into the SSLCA file of NSK C-agent.



#### Update SSLCA file with Veridata Server Certificate

- Open the SSLCA file using the vi editor in OSS (POSIX environment).
- Append the contents of Veridata Server certificate pem file, for example, vdtServer.pem to the end of the SSLCA file. For more information about steps to generate a new Veridata Server's certificate, see Generate Server Keystore and Certificate.

#### Create a Agent Connection in UI

- Log in to the Veridata website, navigate to the Connections page from the left side panel, and click Create on the right side of the page.
- Enter the required connection name and agent host machine address/port. Select the Use SSL for communication checkbox to enable SSL for the connection, and click Verify to verify the connection.
- The Username, Password, Repair User and Repair Password are optional for NSK Cagent connection. The Test Connection buttons are disabled for now.
- Click Submit button to save the connection.

# 8.3 Configuring Oracle GoldenGate Veridata Agent Using Kerberos to Connect to Oracle Database

To configure Oracle GoldenGate Veridata Agent using Kerberos to connect to Oracle database:

- 1. Complete the steps detailed in #unique 266.
- 2. Initiate initial ticket granting ticket for the principal using okinit. To request an initial ticket, run okinit username. The username is the user created or configured to use kerberos.
- 3. Login database instance with an Oracle Net Service service name. Run sqlplus / @service\_name to login to the db instance, and then run show user. The displayed user should be the user granted the initial ticket before.
- 4. Copy Kerberos configuration file and ticket cache file into Veridata agent deploy directory. Absence of either file in agent deploy directory disables the kerberos use of the Oracle GoldenGate Veridata agent.
- 5. Edit agent.properties. For example: database.url=jdbc:oracle:thin:@host1.us.oracle.com:1522:vdtkbr. The database.url is the same as the url that is in a non-Kerberos configuration.
- 6. In the agent.properties file, add, uncomment, and edit the entries, kerberos.configuration.file.name and oracle.kerberos.ticket.cache.file.name. A missing entry or an incorrect entry disables the Kerberos use of Oracle GoldenGate Veridata agent. If Kerberos use is not desired, then comment out or delete either of the entries.

#### For example:

#Kerberos configuration file name.

Comment the entry to disable veridata agent to use kerberos.

#To make veridata agent to use kerberos, the file must be in the agent install directory.

kerberos.configuration.file.name=krb.conf

#Kerberos ticket cache file name for Oracle.

#To make veridata agent to use kerberos, the file must be in the agent install directory. oracle.kerberos.ticket.cache.file.name=krb.cc

7. Start the Oracle GoldenGate Veridata Agent: ./agent.sh.



Verify connection in UI. Note that you do not have to enter the username and password in the Database details.

# 8.4 Configuring Oracle GoldenGate Veridata Agent Using Kerberos to Connect to Hive

To configure Oracle GoldenGate Veridata Agent using Kerberos to connect to Hive database:

- Complete the steps detailed in <u>#unique 266</u>.
- 2. Obtain the Kerberos configuration file from the kerberos server, for example, krb5.conf.
- 3. Copy the Kerberos configuration file in the OS default location. For example, in Linux, it is /etc/.
- For Hive specific configurations, review: VERIDATA\_AGENT\_HOME/veridata/agent/sample\_properties/ agent.properties.hive.
- 5. Obtain the the keytab file from Kerberos enabled Hive environment.
- 6. Copy the keytab file into the Oracle GoldenGate Veridata Agent deploy directory.
- 7. In the agent.properties file, add/uncomment, and edit the entries hive.kerberos.keytab.file.name and enter the keytab file name.
- 8. Edit agent.properties and add the database.url. For Kerberos authentication principal argument is required. For example, in a Cloudera Hive, following is the jdbc url syntax: .
  - database.url=jdbc:hive2://<*Hive server host*>:10000/default;principal=hive/HiveServerHost@YOUR-REALM.COM
- Edit agent.properties.hive and add the server.jdbcDriver appropriately. For example, for Cloudera Hive:

server.jdbcDriver=commons-collections-3.2.1.jar hadoop-common-2.4.1.jar hive-service-0.14.0.jar hadoop-mapreduce-client-core-2.4.1.jar hive-shims-common-0.14.0.jar commons-logging-1.1.3.jar hive-exec-0.14.0.jar log4j-1.2.17.jar hive-jdbc-0.14.0-standalone.jar slf4j-api-1.7.5.jar hadoop-auth-2.4.1.jar hive-metastore-0.14.0.jar slf4j-log4j12-1.7.5.jar commons-configuration-1.6.jar commons-dbcp2-2.5.0.jar commons-pool2-2.6.0.jar

Obtain the appropriate versions of these jars from Hive environment.

10. Initiate an initial ticket granting ticket for the principal using kinit. Go to the agent installation directory and run kinit and verify using:

```
klist: kinit
-k -t {keytab file} {principal name}
```



For auto renewal of Kerberos ticket, add the hive.kerberos.principal.override property in the agent.properties file as follows: hive.kerberos.principal.override=<*Principal name*>. If you do not mention the Principal name, then Oracle GoldenGate Veridata uses the principal name form the database.url.

- 11. Start the Oracle GoldenGate Veridata Agent.
- 12. Verify connection in UI.





#### (i) Note

In case you have permission-related issues, you may have to enter the Hadoop username and password in Database details. First, try without using the username and password.

# 8.5 Connecting Oracle GoldenGate Veridata to SSL-Enabled **Oracle Database**

**SSL Enabled JDBC URL format**: jdbc:oracle:thin:@tcps://<host>:<port>/servicename? wallet\_location=<wallet path>, where wallet\_location is the Directory path for client wallet files, which are cwallet.sso and ewallet.p12.

To connect Oracle GoldenGate Veridata to SSL-Enabled Oracle Database:

- Create an auto-login wallet in the database as follows: \$ orapki wallet create -wallet create -wallet path> pwd <wallet password> -auto\_login.
- Create a self-signed certificate and load it into the wallet: \$ orapki wallet add -wallet <wallet path> pwd <wallet password> -dn "CN=<database hostname>" -keysize 1024 -self\_signed -validity 3650
- Export the certificate so that you can load it into the client wallet: \$ orapki wallet export -wallet <wallet path> -pwd <wallet password> -dn "CN=<hostname>" -cert <server certificate path>
- Repeat step 1 to 3.
- Exchange Client and Server Certificates:
  - Load the server certificate into the client wallet: \$ orapki wallet add -wallet <client wallet path> pwd <wallet password> -trusted\_cert -cert <server certificate path>
  - b. Load the client certificate into the server wallet: \$ orapki wallet add -wallet <server wallet path> pwd wallet password -trusted\_cert -cert <client certificate path>
- Check the contents of the server/client wallet: \$ orapki wallet display -wallet <server wallet path> -pwd <wallet password>

For more information, see Create JKS Wallets for a TLS Connection to a DB System that has Client Authentication Enabled in the Administering Oracle Data Safe guide.

# 8.6 Connecting Oracle GoldenGate Veridata to SSL-Enabled MySQL Database

#### Setting up Agent Authentication via server certificate

- Copy ca.pem file from MySQL database server to veridata agent machine.
- Run the keytool command in the veridata agent machine to import the ca.pem file: \$> keytool importcert -alias <Set\_Your\_Alias> -file ca.pem -keystore truststore -storepass <Set\_Your\_Password>



# Note

If the truststore file does not already exist, then a new one is created; else the certificate gets added to the existing file.



- 3. Append the following connection parameters to MySQL JDBC URL in the <AGENT DEPLOY LOCATION>/agent.properties file.
  - For MySQL version 8.0.12 and earlier: database.url=jdbc:mysql://abc.com:3306? useSSL=true&verifyServerCertificate=true.
  - For MySQL version 8.0.13 and later: database.url=jdbc:mysql://abc.com:3306? sslMode=<VERIFY\_CA or VERIFY\_IDENTITY>.



Note

The VERIFY IDENTITY option does not work with the default, self-signed certificate ca.pem from mysql.

4. Export Java parameters to read the truststore you just created or modified. Export JAVA\_OPTS="\$JAVA\_OPTS -Djavax.net.ssl.trustStore=<path\_to\_truststore\_file> -Djavax.net.ssl.trustStorePassword=<your\_truststore\_password>"

#### Setting up Client Authentication via client certificate

- Copy client-cert.pem and client-key.pem two files from MySQL database server to veridata agent machine.
- 2. Run the opensal command in the veridata agent machine to convert the client key and certificate files to a PKCS #12 archive: \$> openssl pkcs12 -export -in client-cert.pem -inkey clientkey.pem -name "<Set Your name>" -passout pass:<Set Your Password> -out client-keystore.p12
- Run the keytool command in the veridata agent machine to import the PKCS file: \$> keytool importkeystore -srckeystore client-keystore.p12 -srcstoretype pkcs12 -srcstorepass <Set\_Your\_Password> destkeystore keystore -deststoretype JKS -deststorepass < Set\_Your\_Password>.



#### (i) Note

If the keystore file does not already exist, then new one is created; else, the certificate is added to the existing file.

After the step, you can delete the PKCS #12 archive (client-keystore.p12 in the example).

- 4. Export java parameters to read the keystore you just created or modified: export JAVA OPTS="\$JAVA OPTS -Djavax.net.ssl.keyStore=<path to keystore file> -Djavax.net.ssl.keyStorePassword=<your\_keystore\_password>".
- Authentication via client certificate does not require connection parameters in MySQL JDBC URL as opposed to via server certificate.

#### 2-Way Authentication

Apply the steps outlined in both Setting up Server Authentication via server certificate and Setting up Client Authentication via client certificate topics.

Export Java parameters to enable both authentication schemas for server and client at the same time:

export JAVA\_OPTS="\$JAVA\_OPTS -Djavax.net.ssl.trustStore=<path\_to\_truststore\_file> -Djavax.net.ssl.trustStorePassword=<your\_truststore\_password> -Djavax.net.ssl.keyStore=<path\_to\_keystore\_file> -Djavax.net.ssl.keyStorePassword=<your keystore password>"



# 8.7 Connecting Oracle GoldenGate Veridata to SSL-Enabled SQL Server Database

**SSL SYNTAX**: database.url=jdbc:sqlserver://

<host>:<port>;databaseName=<dbName>;encrypt=false;trustServerCertificate=false;

**Example URL**: :database.url=jdbc:sqlserver://

phoenix007.dev3sub2phx.databasede3phx.oraclevcn.com:1433;databaseName=atssrc;encrpt=false;trustServerCertificate=false:

To connect Oracle GoldenGate Veridata to SSL-EnabledSQL Server:

- **1.** Import the mssql crt file to the truststore: keytool-importert-alias ms\_sql keystore \$JAVA\_HOME/jre/lib/security/cacerts -storepass changeit -file <path>/mssql.pem.
- 2. Update the jdbc url in agent.properties.
- 3. Set the path of truststore and truststore password in \$JAVA\_OPTS variable export JAVA\_OPTS="\$JAVA\_OPTS -Djavax.net.ssl.trustStore=<path\_to\_truststore\_file> Djavax.net.ssl.trustStorePassword=<put>your\_truststore\_password>"
- 4. Start the agent.

# 8.8 Connecting Oracle GoldenGate Veridata to SSL-Enabled PostgreSQL Database

**SSL SYNTAX**: database.url=jdbc:postgresql://<machine\_name>:5432/<db\_name>?sslmode=verify-ca&sslrootcert=<full path of the cert>/root.crt

**Example URL**: database.url=jdbc:postgresql://phoenix007:5432/postgres?sslmode=verify-ca&sslrootcert=<path>/
root.crt

To connect Oracle GoldenGate Veridata to SSL-Enabled PostgreSQL:

- Copy the root certificate root.crt in your agent location and provide full path of it for sslrootcert parameters.
- 2. For sslmode there can be two values verify-ca and verify-full.

9

# Migrate

 Migrating Veridata Configuration from Veridata 12.2.1.4.0 to Oracle GoldenGate Veridata 23c

# 9.1 Migrating Veridata Configuration from Veridata 12.2.1.4.0 to Oracle GoldenGate Veridata 23c

You cannot migrate users and user groups from Oracle GoldenGate Veridata 12c to 23c. You need to create users and user groups manually. It is a one-time task .The history will not be available in the new Oracle GoldenGate Veridata 23c post migration. You can view the historical information only from the previous version of Veridata 12c server.

To migrate the configurations, you need to use the Export utility in Oracle GoldenGate 12.2.1.4.0 and then import the xml by running the Import utility in Oracle GoldenGate Veridata 23c.



Prior to migration, ensure that you have installed the Oracle GoldenGateVeridata 23c binaries on a new server or in a new directory path of the existing server.

To migrate from Oracle GoldenGate Veridata 12c to 23c:

- Go to the Oracle GoldenGate Veridata 12.2.1.4.0 installation location and run the Export utility. See <u>Using the Veridata Import and Export Utilities</u> in *Administering Oracle GoldenGate Veridata 12c*. An xml file gets created. You need to import this xml file by running the Import utility in Oracle GoldenGate Veridata 23c.
- 2. From the Oracle GoldenGate Veridata Web UI, run the Import utility. See <u>Utilities</u>. You can also run the Import utility as follows:

./vdt-import.sh -user <username> <-create| -delete | -update | -replace > <fileName>

See Using the Import Utility.

# Performance

Improving the Performance of Oracle GoldenGate Veridata

# 10.1 Improving the Performance of Oracle GoldenGate Veridata

You can use the following parameters and configurations to improve the performance of Oracle GoldenGate Veridata, when processing large volumes of data.

#### **Adding Primary Key Columns**

Refer <u>Specifying Primary Key (PK) Columns</u> section to know how adding primary key can help improve the performance.

### **Partitioning**

Refer <u>Partitioning</u> section to know how dividing data into comparable units chunks can help improve performance.

# **Delta Comparison**

Refer Delta Comparison section to know how delta compare can help improve performance.

# **Excluding Columns**

Refer <u>Exclude Columns</u> section to know how and when to remove columns to help improve performance.

# **Server Configurations**

#### COOS join

Refer <u>Server Configurations</u> to know when to use COOS join and other server configurations that can improve performance.

#### **Agent Configurations**

#### COOS Batch Fetch

You can use COOS Batch Fetch instead of fetching data individually, so that the agent fetches data in batches. Refer Coos Batch Fetch.

#### ROWSCN

You can use ROWSCN to skip full table comparison and compare data after a particular SCN value.

Refer **ROWSCN** to know how to set the ROWSCN value.

# Change the Database Transaction Isolation Level

Each Oracle GoldenGate Veridata agent has an agent.properties file in the root of its installation folder that contains environment parameters. One of those parameters is database.transaction.isolation. This property controls the transaction isolation level that is used during the initial comparison step.



The default value for SQL Server and Teradata is READ UNCOMMITTED. This means that the guery that is issued to select rows can read rows that have been modified by other transactions but not yet committed. It does not issue shared locks to prevent other transactions from modifying the data, nor is it blocked by the locks of other transactions.

The advantage of using the READ UNCOMMITTED option is that the initial reads of the data are faster because they are not affected by locking. The disadvantage is that more records could be labelled as possibly out-of-sync, because the data can change over the course of the transaction, and will be compared again during the confirmation step.

If there are too many rows being compared in the COOS step, you can edit the properties file and set database.transaction.isolation to COMMITED, which only permits the fetching of committed records. You must weigh any improvement against the possibility that the initial comparison step becomes slower due to the effect of locks to preserve read consistency.

#### Note

The only value that is supported for Oracle is READ COMMITTED, and the COOS step always uses READ COMMITTED, because at this stage uncommitted records reads are not accepted.

Refer Agent Configurations to learn more about setting Database Transaction Isolation Level.

### **Profile Configurations**

You can control certain parameters for the sorting, initial compare, COOS and repair that can help increase performance.

## **Sorting Configurations Server Sort**

The following factors affect the performance of the sorting mechanism:

- The number of rows in the tables being compared
- The indexes that are defined on the tables
- The keys that are being used
- The way that the database is tuned

Refer Sorting Configuration, to know how to use server-side sorting.

#### **Initial Comparison Configurations**

- Max Concurrent Comparison Threads: Refer Max Concurrent Comparison Threads, to know how to use comparison threads.
- Optimizer Hints: Refer Source and Target Optimizer Hint to know how to use optimizer hints.
- **Limit Number of Input Rows**: Limit the number of rows that are compared, by using this parameter. You can constrain the number of rows that are fetched for processing, for a specific job profile. This enables you to process a smaller number of rows to find out how out-of-sync (or not) the entire table is. Based on the results, you can run a complete comparison or just resynchronize the data. This parameter is a general parameter for the initial comparison process.

To set this parameter:

- In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- Click the Initial Compare tab and then update value for Limit Number of Input Rows



Increase process priority (NonStop only): Assign the Oracle GoldenGate Veridata
Agent the highest process priority possible on a NonStop system. You can assign a
priority and a process name and CPU number, by using the NonStop settings of the
initial and confirmation steps in the job profile.

#### Out-of-Sync Configurations

- COOS Batch Size: Refer COOS Batch Size, to know how to use COOS batching.
- Optimizer Hints Refer Optimizer Hints, to know how to use optimizer hints.

#### Run COOS in Parallel with Initial Compare

By default, Oracle GoldenGate Veridata runs the initial compare and confirm-out-ofsync processes concurrently. If you run them in sequence, fewer system resources are used, but it will take longer to get results.

To verify this:

In the Veridata UI, click **Profiles** and select a profile.

Click Confirm-Out-of-Sync. Check the value for Run Concurrently With Initial Compare. This option should be selected (checked).

#### Skip COOS step

The default setting is to always perform a COOS step. You can skip this step if the database is guiesced or if replication is not actively replicating data changes.

To skip COOS step:

- 1. In the Oracle GoldenGate Veridata UI, click **Profiles** and select a profile.
- 2. Click the Confirm-out-of-sync tab and then uncheck the Use Default and Value for Perform Confirm Out-of-Sync Step.

#### Increase process priority (NSK)

Assign the Oracle GoldenGate Veridata Agent the highest process priority possible on a NonStop system. Use the **NonStop Process Setting** under **Initial compare and Confirm-out-of-sync** in Profile Configuration.

Refer <u>Out-of-Sync Configuration</u> section to know other COOS configurations that can help improve performance.

#### **Connection Configurations**

#### **Compare Fetch Size**

Increase the batch size of fetched rows to increase throughput. Refer <u>Compare Fetch Size</u> to know how to increase fetch size.

#### **Repair Configurations**

Refer <u>Repair Configuration</u>, to know how to set the following parameters to improve performance.

- Number of Concurrent Repair Operations
- Repair Batch

# Accessibility

- About this Accessibility Article
- Using Oracle GoldenGate Veridata with Assistive Technologies and Keyboard

# 11.1 About this Accessibility Article

This article consists of information about accessing Oracle GoldenGate Veridata with a assistive technologies and a keyboard, and guidelines for developing Oracle GoldenGate Veridata applications that are accessible and highly usable to all users.

"Oracle is committed to creating accessible technologies and products that enhance the overall workplace environment and contribute to the productivity of our employees, our customers, and our customers' customers." —Safra Catz, Chief Executive Officer, Oracle.

- Accessibility Overview
- What is Web Accessibility?
- Why Accessibility is Important?
- · About Building for Accessibility

### 11.1.1 Accessibility Overview

Web accessibility means people with disabilities can effectively use and contribute to the web. To ensure people with disabilities can use Oracle GoldenGate Veridata, we need to design and develop more accessible web applications and software.

Accessibility is not just a checklist, but an ongoing and planned effort. The aim of this guide is to help you start the planning process, and understand how to build accessible applications with Oracle GoldenGate Veridata.

### 11.1.2 What is Web Accessibility?

This article refers to the <u>Web Content Accessibility Guidelines 2.0 (WCAG 2.0)</u> to back up some of the guidelines with the relevant standard and provide further reading.

According to W3C, web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the web, and that they can contribute to the web. The disabilities that affect access to the web can be categorized as: visual, auditory, speech, cognitive, physical, and neurological. The goal of web accessibility is to provide equal access to users with disabilities. That is, developers need to focus on building more accessible web applications and products. Apart from people with disabilities, the elderly with increasing impairments, and people in a limiting situation (for example, slow internet connection, or no audio) can also benefit with accessible applications and products.

#### **Principles of Accessibility**

WCAG 2.0, which is created by W3C defines the four principles of accessibility as:



- Perceivable: Information and user interface components must be presentable to users in ways they can perceive. That is, users must be able to perceive the information presented.
- Operable: User interface components and navigation must be operable. That is, users
  must be able to operate the interface.
- Understandable: Users must be able to understand the information as well as the operation of the user interface.
- Robust: Content must be robust enough that it can be interpreted reliably by a wide variety
  of user agents, including assistive technologies.

Each principle includes a list of guidelines that address the principle and there are a total of 12 guidelines. Each guideline includes one or more testable success criteria, which are at three levels: A, AA, and AAA. To learn more about these principles, guidelines, and success criteria, see Web Content Accessibility Guidelines 2.0 (WCAG 2.0).

### 11.1.3 Why Accessibility is Important?

Up to 16% of the world population is disabled, either through birth, ageing, illness, or the result of an accident. Accessible products remove obstacles between a company and its existing or potential customers. Accessibility enables employers to recruit from a broader pool of talent. Overall, accessibility generally improves product for everyone (Universal Design).

Developing accessible products is very important for companies in order to adhere to Procurement and Discrimination laws such as:

- Section 508 of the Rehabilitation Act of 1973
- Mandate 376 (E.U.)
- Accessibility for Ontarians with Disabilities Act (Canada)
- Americans with Disabilities Act (U.S.)
- Disability Discrimination Act, and Equality Act (U.K.)
- Twenty-First Century Communications and Video Accessibility Act (U.S.)

### 11.1.4 About Building for Accessibility

Building for accessibility means coding to standards and not for a specific technology. Building to standards often leads to better, standardized HTML and cleaner code. The process of building accessible applications at Oracle involves considerable amount of time and resources at all stages of the development cycle. Accessibility guidelines and standards are considered, implemented, and reviewed through the Voluntary Product Accessibility Template (VPAT, a product's statement of conformance to Section 508 of the U.S. Federal Rehabilitation Act).

The Oracle Accessibility Program Office, reporting to the office of the Chief Corporate Architect, is responsible for defining the corporate standards for accessibility, and developing materials to train all employees so that they can successfully create products that meet those standards. To learn more about Oracle's Accessibility Program, see <a href="Oracle Accessibility Information and Resources">Oracle Accessibility Information and Resources</a>.

Oracle uses the VPAT to represent the degree of conformance to various accessibility standards and guidelines, including Section 508, WCAG 1.0, and WCAG 2.0. Depending on when a product was developed and released, different standards may be listed.

.



# 11.2 Using Oracle GoldenGate Veridata with Assistive Technologies and Keyboard

You can navigate and perform user interface (UI) actions without a mouse by using keyboard controls. Common keyboard controls are available in all applications. Additional keyboard controls are used for complex interactions and vary by component type.

Table 11-1 Frequently used keyboard controls

Key(s)	Action	
Enter	Trigger the activity, when the focus is on a link, an image, or a button with an associated URL or activity.	
	Toggle the focus or selection.	
	For input fields and menus, submit or enter a typed value or select the highlighted item.	
	In Tables, make the current row editable or enter Actionable Mode.	
Shift + Enter	For cells in editable Tables, make the previous row editable and move the focus to the editable cell in the current column of the previous row. If the first row is editable, make it read-only.	
Space	Trigger the activity, when the focus is on a link, an image, or a button with an associated URL or activity.	
	Toggle the focus or selection.	
Shift + Space	In components such as Card View, Data Grid, List View, and Tree View, selects contiguous items from the last selection to the current item.	
Ctrl / Cmd + Space	Multi-select item with focus.	
	In List View and Tree View, toggles the selection of the current item while maintaining previous selections.	
Tab	Sets focus to the first focusable element or cycles focus through next focusable elements.	
	For some components, such as Card View, Data Grid, List View, Table, or Waterfall Layout, when in Tabbable or Actionable Mode, if the focus is on the last element, the focus goes back to the first focusable element. When not in Tabbable or Actionable mode, the focus goes to the next focusable item on the page.	
Shift + Tab	Cycles focus through previous elements.	
	For some components such as Card View, Data Grid, List View, Table, or Waterfall Layout, when in Tabbable or Actionable Mode, if the focus is on the first element, the focus goes back to the last focusable element. When not in Tabbable or Actionable mode, the focus goes to the previous focusable item on the page.	



Table 11-1 (Cont.) Frequently used keyboard controls

Koy(o)	Action	
Key(s)	Action	
Up Arrow	Moves focus and selection to previous item or scrolls the content up.	
	For drop down lists, expands the list when closed.	
	For numerical input fields, increases the number by one in the segment.	
	For slider components, scrolls right on horizontal sliders and up on vertical sliders.	
Down Arrow	Moves focus and selection to next item or scrolls the content down.	
	For drop down lists, opens the list when closed.	
	For numerical input fields, decreases the number by one in the segment.	
	For slider components, scrolls left on horizontal sliders and down on vertical sliders.	
Left Arrow	Moves focus and selection to the previous item or the item on the left. Can also scroll the content left.	
	For collapsible groups inside collection components collapses the current expanded item.	
	For slider components, scrolls left on horizontal sliders and down on vertical sliders.	
Right Arrow	Moves focus and selection to the next item or the item on the right. Can also scroll the content right.	
	For collapsible groups inside collection components expands the current item if it has children.	
	For slider components, scrolls right on horizontal sliders and up on vertical sliders.	
Shift + Arrow	For selectable elements within collection components, moves focus and extends the selection in the direction of the arrow	
Shift + Ctrl / Cmd + Arrow	In Card View, reorders the current item in the direction of the arrow.	
F2	For some components, such as Cards, Data Grid, List View, Table, or Waterfall Layout, enters Tabbable or Actionable Mode allowing navigation and use of controls in the item.	
	Exits Tabbable or Actionable Mode.	
Alt + F10	For Rich text input, moves focus to the text editing toolbar.	
Esc	Exits Tabbable or Actionable Mode.	
	Collapses drop down lists.	
= or +	Zoom in one level (if zooming is enabled).	
- or _	Zoom out one level (if zooming is enabled).	
Page Up	For Data grid and data visualization components, such as Diagram and Timeline, pans up (if scrolling is enabled).	



Table 11-1 (Cont.) Frequently used keyboard controls

Key(s)	Action
Page Down	For Data grid and data visualization components, such as Diagram and Timeline, pans down (if scrolling is enabled).
Shift + Page Up	For Data grid and data visualization components, such as Diagram and Timeline, pans left (in left-to-right locales).
	Pans right (in right-to-left locales).
Shift + Page Down	For Data grid and data visualization components, such as Diagram and Timeline, pans right (in left-to-right locales).
	Pans left (in right-to-left locales).
Home or End	For Data grid, moves focus to the first or last focusable item.
Ctrl / Cmd + X	For draggable elements, marks the selected items to move if reorder is enabled.
Ctrl / Cmd + C	For draggable elements, marks the selected items to copy if reorder is enabled.
Ctrl / Cmd + V	For draggable elements, pastes selected items directly before the current item (or as the last item if the current item is a folder).

#### (i) Note

For more information about Oracle's commitment to accessibility, see <u>Oracle's Accessibility Program</u>.

Component-Specific Keyboard Controls

# 11.2.1 Component-Specific Keyboard Controls

Some components use additional or alternative keyboard controls. For more details, review the specific component information on the <u>Oracle JavaScript Extension Toolkit (JET) Keyboard and Touch Reference</u> or navigate the links listed in the following table.



Table 11-2 Component-Specific Keyboard Controls

Collections	Controls	Forms and Inputs	Layouts and Navigation	Visualizations
Data grid	Button	Checkbox	Accordion	Area chart
<u>Indexer</u>	File picker	Checkbox set	Action card	<u>Chart</u>
<u>List view</u>	Film strip	Color palette	<u>Collapsible</u>	<u>Diagram</u>
Row expander	<u>Menu</u>	Color spectrum	Conveyor belt	<u>Gantt</u>
Stream list	Menu button	<u>Date picker</u>	Dialog	<u>Legend</u>
Swipe actions	<u>Message</u>	<u>Input date</u>	<u>Drawer layout</u>	<u>Line chart</u>
<u>Table</u>	Message banner	<u>Input date mask</u>	Navigation list	<u>Meter bar</u>
<u>Tree view</u>	Message toast	Input date text	<u>Popup</u>	Meter circle
Waterfall layout	<u>Train</u>	Input date time	<u>Tab bar</u>	Nbox
		Input number		<u>Picto chart</u>
		Input password		Rating gauge
		Input search		<u>Sunburst</u>
		Input sensitive text		Tag cloud
		Input text		Thematic map
		Input time		<u>Timeline</u>
		<u>Radioset</u>		<u>Treemap</u>
		Select multiple item		
		Select single item		
		Slider		
		Range slider		
		<u>Switch</u>		

#### (i) Note

For more information about Oracle's commitment to accessibility, see  $\underline{\text{Oracle's}}$  Accessibility Program.

# Troubleshooting

- Configuration Assistant
- MySQL Operation
- Comparison and Repair Performance

# 12.1 Configuration Assistant

Issue: install\_configure\_mysql.sh fails with "Permission denied" error

#### Error Message:

[ERROR] [MY-013276] [Server] Failed to set datadir to '<veridata\_installation\_dir>/mysql-commercial-\*.\*.\*\*-linux-glibc2.17-x86\_64-minimal/data/'
(OS errno: 13 - Permission denied)

#### **Troubleshooting Options**

- 1. Check Directory Permissions:
  - a. Run the following command to check access permissions (Note: "/ogg/app/oracle/veridata/23 is an example path of <veridata\_installation\_dir>):
    - ls -ld /ogg /ogg/app /ogg/app/oracle /ogg/app/oracle/veridata /ogg/app/oracle/veridata/23 /ogg/app/oracle/veridata/23 /mysql-commercial-\* \* \* \*\*-linux-glibc2.17-x86\_64-minimal
  - **b.** If any parent directories lack the x (execute) permission, then vdtrepouser is unable to access the data directory.
- Verify SELinux Policy:
  - a. Switch to the root user.

  - c. Run the command: ls -lt ../data
  - d. If the directory permission output does not include a trailing "." (for example, drwxr-x---.), then SELinux is not the issue. Otherwise, set SELinux to Permissive or Disabled mode.

# 12.2 MySQL Operation

**Issue**: Error when trying to log into MySQL (./bin/mysql):

#### Error Messages:

- error while loading shared libraries: libtinfo.so.5: cannot open shared object file: No such file or directory
- error while loading shared libraries: libraries. So.5: cannot open shared object file: No such file or directory

#### **Troubleshooting Options**

Create Symlinks for Missing Libraries (if libtinfo.so.6 and libncurses.so.6 exist in the OS):



#### **Commands**

ln -s /usr/lib64/libncurses.so.6 /usr/lib64/libncurses.so.5

ln -s /usr/lib64/libtinfo.so.6 /usr/lib64/libtinfo.so.5

Install the Required Package using the following command:

yum install neurses-compat-libs

# 12.3 Comparison and Repair Performance

- Slowed Comparison
- Slowed Repair

### 12.3.1 Slowed Comparison

When you initiate a compare job, it runs in five phases - Pending, Initialising, Sorting, Initial Compare & COOS, and Finished. Compare job slows down because of lag in any one of these phases.

#### To find out the phase which is stalled or slow:

- 1. On the Veridata UI, select Running Jobs.
- 2. Select the job and check the status of the **Run Phase** column.

Based on the Run Phase column value, follow the corresponding steps to resolve the issue.

- Sorting Phase
- Initial Compare Phase
- COOS Phase
- Check Stalled Queries in Database

### 12.3.1.1 Sorting Phase

If the Run Phase column value is SORTING, then follow the troubleshooting steps below.

Most common issues that can occur in the **Sorting** phase and their solutions are listed in this section.

#### Sorting is slow or stalled

This occurs when there is a lag in fetching the data from the source or target databases, due to:

#### Sorting done at Database

If you are sorting at the database, it can be slow. We recommend sorting at the server (Veridata server), which is faster.





#### (i) Note

Server sorting has a higher server memory usage, but it is faster.

To enable server sorting and to know more about sorting configurations, refer Sorting Configuration.

#### Insufficient memory allocated to the Sort Directory

To know about the sort directory and its location, refer **Sort Directory**.

To know the memory requirement for the sort directory, refer Memory for Sorting and Sort Directory.

To change memory mapped sort directory:

Memory mapped sort directory is where .map files are generated. By default it is temp directory of the operating system, but it can be changed from veridata.cfg file:

server.memory\_mapped\_sort\_directory directorypath

Example: server.memory\_mapped\_sort\_directory directorypath/tmp2

Apart from server.max\_sort\_memory, below parameters in the veridata.cfg file can also be modified to accelerate the sorting process.

- server.concurrent.writers: The number of writer threads per sort directory.
- server.number sort threads: The number of threads used to sort input buffers from the agent. This number should not be larger than the number of available processes.
- server.concurrent.readers: The number of reader threads for entire server.

Refer for more details.

#### **Compare Fetch Size**

To accelerate the compare process, the number of rows fetched from database can be increased in the connection settings, by changing the Compare Fetch Size. Start with 10,000 and increase up to 100,000.

Refer Compare Fetch Size.

Increasing this value accelerates the fetch process during the initial comparison. Run the comparison again after changing the value.

#### Queries Stalled in the Database

To find out if queries are stalled:

On the Veridata UI, select Monitor Jobs, and select the job.

Check the values in the Last Source Fetch and Last Target Fetch columns. If these values remain unchanged, despite not all rows from the source and target being fetched, then check values of the Source Fetch Latency and Target Fetch Latency column.

- If the latency values are high, then it can be due to slow network between server and agent or between agent and DB.
- If all the values look good, then database might have some running query that is not getting completed.

Refer Check Stalled Queries in Database, to check if any queries are running or stalled in both source and target DB.



#### **Compare Pair is Stalled**

If the compare pair remains stalled, despite changing all parameters, cancel the existing run, and rerun that particular compare pair. Verify the successful completion of the run.

### 12.3.1.2 Initial Compare Phase

If the **Run Phase** column value is **INITCOMPARE** or **INITCOMPARECOOS**, then follow the troubleshooting steps below.

The issues that can occur in the **Initial Compare** or the **Initial Compare and COOS** phases, are due to the lag in the comparison between the source and the target data. The possible solutions to these issues are listed in this section.

#### Check if the Table has Key Columns

Refer Specifying Primary Key (PK) Columns, to improve performance with key columns.

#### **Use Delta Compare**

Refer Delta Comparison, to enable delta compare to accelerate the comparison.

#### **Source and Target Oracle Optimizer Hint**

If the database is Oracle, you can use an optimizer hint to improve the processing of the queries.

Refer Optimizer Hints.

#### **Managing Max Concurrent Comparison Threads**

Running multiple comparison pairs in parallel can improve the overall compare time.

Refer Max Concurrent Comparison Threads.

#### **Use Auto Partitions or Row Partitions or Table Partitions**

You can divide large source and target tables into partitions. Refer Partioning.

#### **Setting ROWSCN**

Set this parameter to skip full table comparison and compare data after a particular SCN value. Refer ROWSCN.

#### **Compare Pair is Stalled**

If the compare pair remains stalled, despite changing all parameters, cancel the existing run, and rerun that particular compare pair. Verify the successful completion of the run.

#### 12.3.1.3 COOS Phase

If the **Run Phase** is **COOS** or **INITCOMPARECOOS**, but optimisations suggested in <u>Initial</u> <u>Compare Phase</u> did not resolve the issue, follow the solutions listed in this section.

#### **COOS Join Or COOS Batch Fetch**

If DB is Oracle DB and temp table feature is supported in it, then use COOS join, else use Coos Batch Fetch.



To verify if the temp table is supported, run the command create private temporary table in DB.

Refer COOS join and COOS Batch Fetch, for more information.



#### Note

Do not use both COOS Join and COOS Batch Fetch together. Use one option.

#### **Updating COOS Batch Size**

Refer COOS Batch Size.

#### **Source and Target Oracle Optimizer Hint**

If the database is Oracle, you can use an optimizer hint to improve the processing of the queries.

Refer Optimizer Hint.

#### **Check Stalled Queries in the Database**

Refer Check Stalled Queries in Database, to learn how to check for stalled queries in different databases.

#### **Compare Pair is Stalled**

If the compare pair remains stalled, despite changing all parameters, cancel the existing run, and rerun that particular compare pair. Verify the successful completion of the run.

#### **Hardware Scaling (Memory and Processors)**

If the above steps do not accelerate the compare, scale up the hardware.

RAM: Adding more memory to a single machine enables multiprocessing, leading to improved performance of applications that require significant memory. Based on the database size, approximately 60% of the data size is required for memory.

Processors: Upgrading to a more powerful CPU or adding additional cores to the existing CPU enhances the processing capabilit, accelerating the system speed and enabling more complex computations.

### 12.3.1.4 Check Stalled Queries in Database

Login to the database and check if any queries are running or stalled.

Table 12-1 Queries Stalled in Database

Database	Query	Output
MySQL/ MariaDB	SHOW PROCESSLIST;	Lists active queries and their status.
		Look for queries that are inLocked status.



Table 12-1 (Cont.) Queries Stalled in Database

Database	Query	Output
PostgreSQL	SELECT pid, state, query, age(clock_timestamp(),	Lists all running queries and their states.
pg_	query_start) AS duration FROM pg_stat_activity WHERE state <> 'idle';	You can indetify long-running queries by the duration.
Microsoft SQL Server	SELECT session_id,status,command,wait_ type,wait_time,wait_resource,cpu _time,total_elapsed_time,query_t ext.text AS query_text FROM sys.dm_exec_requests CROSS APPLY sys.dm_exec_sql_text(sql_handle ) AS query_text WHERE session_id > 50; Exclude system sessions	Displays detailed information about running queries, including any that are waiting on resources.
Oracle DB	SELECT s.sid, s.serial#, s.status, s.username, s.sql_id, t.sql_text FROM v\$session s JOIN v\$sqltext t ON s.sql_id = t.sql_id WHERE s.status = 'ACTIVE';	Lists all active sessions and their SQL text.
SQLite	PRAGMA database_list;PRAGMA locking_mode;	Lists all active sessions and their SQL text.

# 12.3.2 Slowed Repair

You can use batching and concurrent repair options to accelerate the repair process.

#### **Batching**

Set the **Repair Batch Size** parameter to enable batch repairing.

#### **Running Concurrent Repair**

Set the Number of Concurrent Repair Operations to enable the required number of concurrent repairs.

Refer Repair Configuration, to know how to set the above repair parameters.