1

Oracle® GoldenGate

Upgrading Oracle GoldenGate Veridata

12*c* (12.1.3)

E49223-01

June 2014

Oracle GoldenGate Upgrading Oracle GoldenGate Veridata, 12*c* (12.1.3)

E49223-01

Primary Author: Savija Vijayaraghavan

2 Contents

3 Preface

This document describes how to upgrade an existing Oracle GoldenGate Veridata 11*g* (11.2.1) environment to 12*c* (12.1.3).

Audience

This document is intended for system administrators who are responsible for installing, maintaining, and upgrading Oracle GoldenGate Veridata.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documentation set:

- *Installing and Configuring Oracle GoldenGate Veridata*
- *Administering 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.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Preparing for the Oracle GoldenGate Veridata Upgrade

This chapter provides a summary of points you should understand before you upgrade Oracle GoldenGate Veridata from the 11.2.1 version to 12.1.3 version.

This chapter includes the following sections:

- Section 1.1, "Overview of the Oracle GoldenGate Veridata Upgrade Process"
- Section 1.2, "Differences Between Oracle GoldenGate Veridata 11g and Oracle GoldenGate Veridata 12c"
- Section 1.3, "Oracle GoldenGate Veridata Server Upgrade Roadmap"
- Section 1.4, "Pre-Upgrade Requirements for Oracle GoldenGate Veridata Upgrade"
- Section 1.5, "Understanding the Upgrade Prerequisites"

1.1 Overview of the Oracle GoldenGate Veridata Upgrade Process

The upgrade process described in this document applies only for an upgrade from Oracle GoldenGate Veridata Release 11.2.1 to Oracle GoldenGate Veridata Release 12.1.3.

If you are using versions prior to 11.2.1 of Oracle GoldenGate Veridata, you must first upgrade Veridata to Release 11.2.1 before upgrading to Release 12.1.3. For more information, see "Upgrading Oracle GoldenGate Veridata" chapter in the 11.2.1 Administrator's Guide at:

`http://docs.oracle.com/cd/E35209_01/doc.1121/e29092/upgrading.htm#CHDDJHGC`

1.2 Differences Between Oracle GoldenGate Veridata 11*g* and Oracle GoldenGate Veridata 12*c*

To know about the new features in Oracle GoldenGate Veridata 12.1.3, see the "What's New" chapter in *Administering Oracle GoldenGate Veridata*.

1.3 Oracle GoldenGate Veridata Server Upgrade Roadmap

Follow the steps below to upgrade Oracle GoldenGate Veridata from 11*g* to 12*c*:

1. Before you upgrade to 12c, back up the existing 11g environment and ensure that all pre-upgrade requirements are met. See "Pre-Upgrade Requirements for Oracle GoldenGate Veridata Upgrade".

2. Ensure that all upgrade prerequisites are met. See Section 1.5, "Understanding the Upgrade Prerequisites".
3. Install Oracle GoldenGate Veridata 12c. See Section 2.1, "Installing Oracle GoldenGate Veridata 12c (12.1.3)".
4. Create repository schemas for your database by using RCU. See Section 2.2, "Creating Repository Schemas using Repository Creation Utility".
5. Upgrade the Veridata repository using the Upgrade Assistant. See Section 2.3, "Using Upgrade Assistant to Upgrade Veridata Schemas".
6. Configure the WebLogic Domain for your Veridata installation. See *Installing and Configuring Oracle GoldenGate Veridata*.
7. Use the Upgrade Assistant to upgrade the Veridata configuration. See Section 2.5, "Upgrading Veridata Configuration by Using the Upgrade Assistant".

1.4 Pre-Upgrade Requirements for Oracle GoldenGate Veridata Upgrade

Before you upgrade, you must do the following:

- Ensure that no jobs are running in the Veridata 11g instance.
- Stop the Veridata 11g Server.
- Back up the Veridata Agent configuration files.
- Back up the Veridata repository.
- Back up old report files. Do not delete the 11g report location if you want to retain old reports.

1.5 Understanding the Upgrade Prerequisites

Following are the prerequisites for upgrading Oracle GoldenGate Veridata Server from 11g to 12c:

- JDK 1.7 must be installed on the target machine. For more information about JDK installation, see the JDK installation guide at: `http://docs.oracle.com/javase/7/docs/webnotes/install/index.html`.
- Oracle WebLogic Server 12*c* (12.1.3) with JRF installed on the target machine. For more information, see *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
- If required, upgrade the Veridata repository database. For more details about the supported databases, see the system requirements document at `https://support.oracle.com/`.

2 Upgrading Oracle GoldenGate Veridata

This chapter describes the tasks you perform to upgrade Oracle GoldenGate Veridata.

This chapter includes the following sections:

- Section 2.1, "Installing Oracle GoldenGate Veridata 12c (12.1.3)"
- Section 2.2, "Creating Repository Schemas using Repository Creation Utility"
- Section 2.3, "Using Upgrade Assistant to Upgrade Veridata Schemas"
- Section 2.4, "Creating the Veridata Domain using the Upgraded Schema"
- Section 2.5, "Upgrading Veridata Configuration by Using the Upgrade Assistant"

2.1 Installing Oracle GoldenGate Veridata 12*c* (12.1.3)

Oracle GoldenGate Veridata 12*c* (12.1.3) runs on Oracle WebLogic Server. Therefore, to upgrade Oracle GoldenGate Veridata 11*g* (11.2.1), which was using Apache Tomcat Web Server, you must first install Oracle GoldenGate Veridata 12c using the Oracle Fusion Middleware installer. After installing, follow the upgrade process described in this document.

For more information about installation, see *Installing and Configuring Oracle GoldenGate Veridata*.

2.2 Creating Repository Schemas using Repository Creation Utility

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

- Oracle
- MySQL
- Microsoft SQL Server

After installing Oracle GoldenGate Veridata 12*c* (12.1.3) and configuring your database, create the necessary schema for your repository database by using RCU. Navigate to the `ORACLE_HOME/oracle_common/bin` directory and start RCU.

RCU creates the `schema_version_registry` table in the database and it is used by Veridata and Upgrade Assistant. Following sections explain how to create schema for various databases:

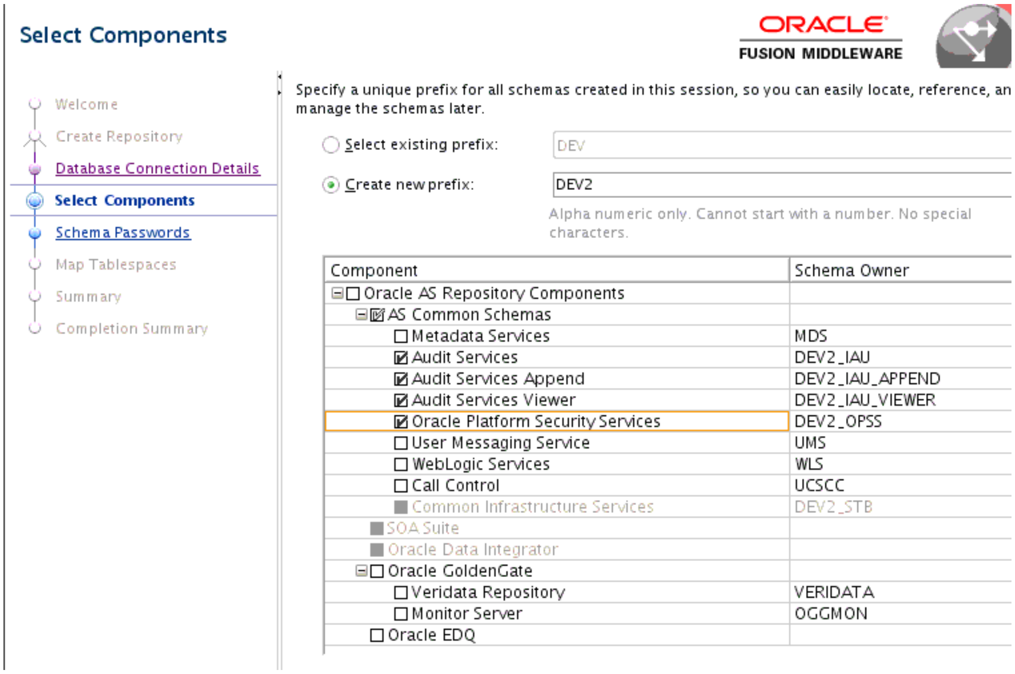
> **Note:** In the RCU pages, when you create the Service Table (STB), specify a unique prefix for the schema for all databases. You need to provide this prefix while running the Upgrade Assistant for upgrading Veridata schema.
>
> To drop the Veridata repository, you should use the same prefix.

- Section 2.2.1, "Creating Schema for Oracle Database"
- Section 2.2.2, "Creating Schema for Microsoft SQL Server"
- Section 2.2.3, "Creating Schema for MySQL"

2.2.1 Creating Schema for Oracle Database

If you are using Oracle database, create the Oracle Platform Security Services (OPSS) schema using RCU. In the Select Components page of RCU, select OPSS as follows:

Figure 2–1 OPSS schema selection for Oracle Database



Specify a unique prefix for all schemas created in this session, so you can easily locate, reference, an manage the schemas later.

○ Select existing prefix: DEV

● Create new prefix: DEV2

Alpha numeric only. Cannot start with a number. No special characters.

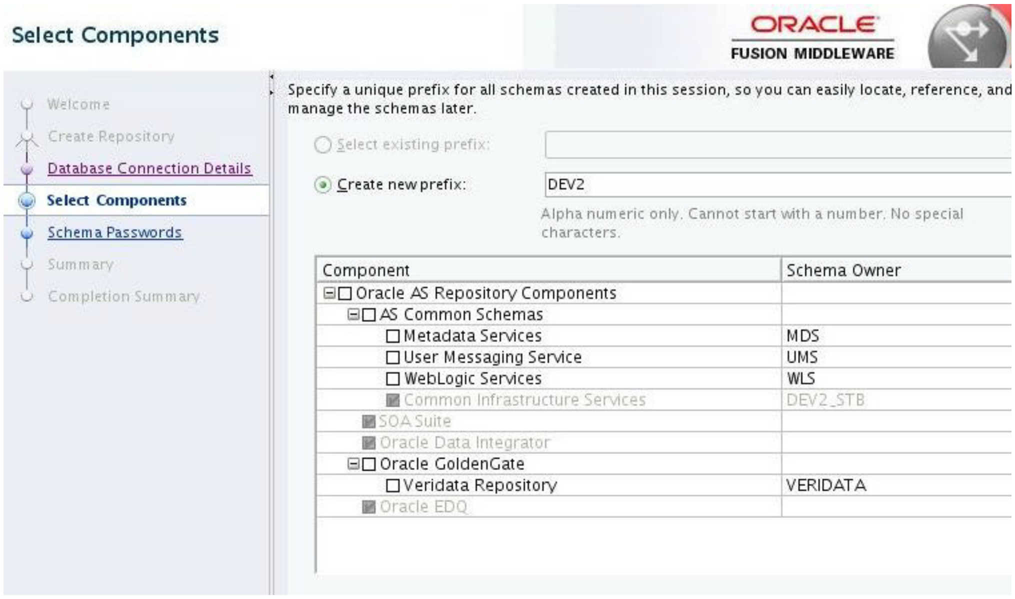
Component	Schema Owner
⊟☐ Oracle AS Repository Components	
⊟☑ AS Common Schemas	
☐ Metadata Services	MDS
☑ Audit Services	DEV2_IAU
☑ Audit Services Append	DEV2_IAU_APPEND
☑ Audit Services Viewer	DEV2_IAU_VIEWER
☑ Oracle Platform Security Services	DEV2_OPSS
☐ User Messaging Service	UMS
☐ WebLogic Services	WLS
☐ Call Control	UCSCC
■ Common Infrastructure Services	DEV2_STB
■ SOA Suite	
■ Oracle Data Integrator	
⊟☐ Oracle GoldenGate	
☐ Veridata Repository	VERIDATA
☐ Monitor Server	OGGMON
☐ Oracle EDQ	

See *Creating Schemas with the Repository Creation Utility* for more information.

2.2.2 Creating Schema for Microsoft SQL Server

For upgrading to version 12.1.3, MS SQL Server users should run RCU to create the Service Table (STB). Select only Service Table in the Select Components page:

Figure 2–2 Service Table selection for Non-Oracle database



Select Components

Specify a unique prefix for all schemas created in this session, so you can easily locate, reference, and manage the schemas later.

○ Select existing prefix:

● Create new prefix: DEV2

Alpha numeric only. Cannot start with a number. No special characters.

Component	Schema Owner
⊟□ Oracle AS Repository Components	
⊟□ AS Common Schemas	
□ Metadata Services	MDS
□ User Messaging Service	UMS
□ WebLogic Services	WLS
☑ Common Infrastructure Services	DEV2_STB
☑ SOA Suite	
☑ Oracle Data Integrator	
⊟□ Oracle GoldenGate	
□ Veridata Repository	VERIDATA
☑ Oracle EDQ	

See *Creating Schemas with the Repository Creation Utility* for more information.

2.2.3 Creating Schema for MySQL

For upgrading to version 12.1.3, MySQL users should run RCU to create the Service Table. Select only STB in the Select Components page of the RCU (See Figure 2–2).

See *Creating Schemas with the Repository Creation Utility* for more information about using RCU.

Setting the configuration parameters for MySQL

Before running RCU for MySQL, execute the following commands at the MySQL command prompt:

```
mysql>SET GLOBAL INNODB_FILE_PER_TABLE="ON";
mysql>SET GLOBAL INNODB_FILE_FORMAT="Barracuda";
mysql>SET GLOBAL INNODB_LARGE_PREFIX="ON";
mysql>SET GLOBAL LOG_BIN_TRUST_FUNCTION_CREATORS="ON";
```

Note: Before you run the RCU, you must ensure that the database name is compliant with the Oracle Fusion Middleware standards. The database name and the user name must be the same and in capital letters.

Renaming the database user name

The `lower_case_table_names` system variable in MySQL determines if the MySQL table lookup is case sensitive or not.

If `lower_case_table_names` is "1", the table lookup is case insensitive.

If `lower_case_table_names` is "0", the table lookup is case sensitive.

Run the following command at the MySQL prompt to see the value of the `lower_case_table_names` variable:

```
mysql > SHOW VARIABLES LIKE "lower_case_table_names";
```

Depending on the value of `lower_case_table_names`, follow the steps below to rename the database user name in uppercase.

- If `lower_case_table_names`=1, then rename the MySQL user name as follows:

  ```
  RENAME USER old_user@'host' TO new_user@'host';
  ```

  where

  `old_user` is the user name of the current Veridata repository.

  `new_user` is the user name that is same as the database name and this should be in uppercase.

  `host` is the host name of the machine from which you need access to the database. To allow access from all machines, specify '%' for the host name.

  Example:

  If the database name is `veridataDB` and user name is `veridataUser`, execute the following command:

  ```
  RENAME USER veridataUser@'localhost' to VERIDATADB@'localhost';

  or
  RENAME USER veridataUser@'%' to VERIDATADB@'%';
  ```

- If `lower_case_table_names`=0 and if the database name is in uppercase, then rename the MySQL user name to uppercase as same as the database name:

  ```
  RENAME USER old_user@'host' TO new_user@'host';
  ```

  where

  `old_user` is the user name of the current Veridata repository.

  `new_user` is the user name that is same as the database name and this should be in uppercase.

  `host` is the host name of the machine from which you need access to the database. To allow access from all machines, specify '%' for the host name.

- If `lower_case_table_names`=0 and if the database name is in lowercase or mixed case, then create a new database and a new user with both the user name and database name in uppercase. See "Creating a new database and user for MySQL database".

  Example:

  If the user name is `veridataUser` and old database name is `veridataDB`, then create a new database with the name as `VERIDATAUSER` and copy contents of `veridataDB` to `VERIDATAUSER`. You should also create a new user `VERIDATAUSER` with access to the new database.

Creating a new database and user for MySQL database

To create a database:

1. Enter the following commands to create a new database from the old database dump:

```
bash>mysqldump -u[username] -p[pass] -v [olddatabase] > [file path to create
sqldump olddbdump.sql]

bash>mysql -u[username] -p[pass]

mysql>create schema [NEWDATABASE]

mysql>use [NEWDATABASE]

mysql>source [file path for sqldump olddbdump.sql]
```

2. To grant access to the database:

```
mysql>grant all on `[NEWDATABASE]`.* to [NEWDATABASE]@'localhost' identified by
'[Password]';

mysql>grant grant option on `[NEWDATABASE]`.* to [NEWDATABASE]@'localhost';

mysql>grant all on `[NEWDATABASE]`.* to [NEWDATABASE]@'%' identified by
'[Password]';

mysql>grant grant option on `[NEWDATABASE]`.* to [NEWDATABASE]@'%';

mysql>flush privileges;
```

Example:

Old database: `veridata18_9`

New database: `VERIDATA18_9`

New user name: `VERIDATA18_9`

Note that both user name and database name are the same.

```
bash> mysqldump -uroot -pwelcome -v veridata18_9 > /home/veridata18_9.sql
bash> mysql -uroot -pwelcome
mysql> create schema VERIDATA18_9;
mysql> use VERIDATA18_9;
mysql> source /home/veridata18_9.sql;
mysql> grant all on `VERIDATA18_9`.* to VERIDATA18_9@'localhost' identified by
'VERIDATA18_9';
mysql> grant grant option on `VERIDATA18_9`.* to VERIDATA18_9@'localhost';
mysql> grant all on `VERIDATA18_9`.* to VERIDATA18_9@'%' identified by
'VERIDATA18_9';
mysql> grant grant option on `VERIDATA18_9`.* to VERIDATA18_9@'%';
mysql> flush privileges;
```

2.3 Using Upgrade Assistant to Upgrade Veridata Schemas

Use Oracle Fusion Middleware Upgrade Assistant to upgrade Oracle GoldenGate Veridata schema to 12c (12.1.3). For more information about the features of Upgrade Assistant, see *Upgrading with the Upgrade Assistant*.

To upgrade the Veridata repository schema follow the instructions below:

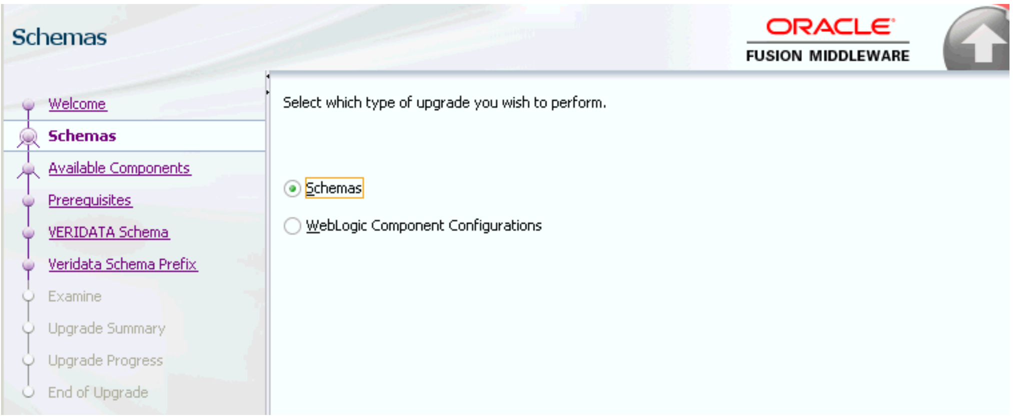
1. To start Upgrade Assistant, go to the `*ORACLE_HOME*/oracle_common/upgrade/bin` directory, and enter the following command:

   On UNIX: `./ua`

   On Windows: `ua.bat`

The Welcome screen of the Upgrade Assistant appears. Click **Next**.

2. Select **Schemas** as the upgrade type. Click **Next**.

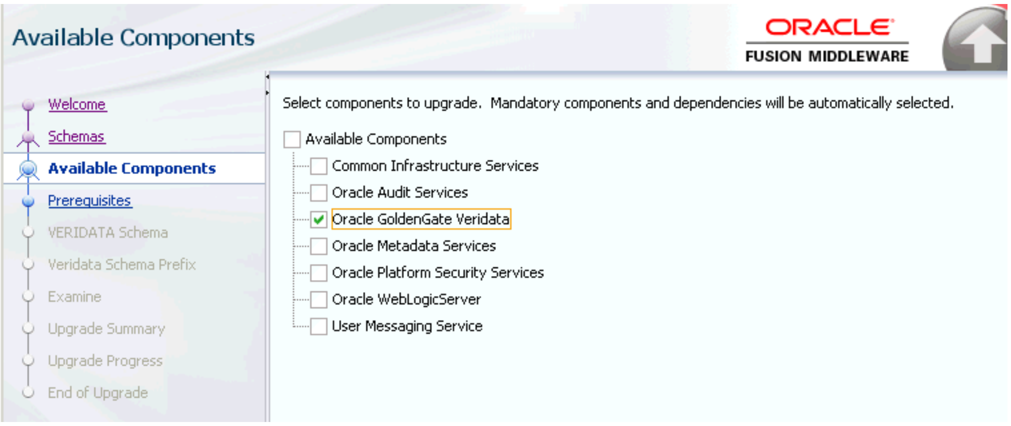


Select which type of upgrade you wish to perform.

● Schemas

○ WebLogic Component Configurations

3. The Upgrade Assistant lists all schemas available for upgrade. Select only Oracle GoldenGate Veridata. Ensure that no other schemas are selected for upgrade.

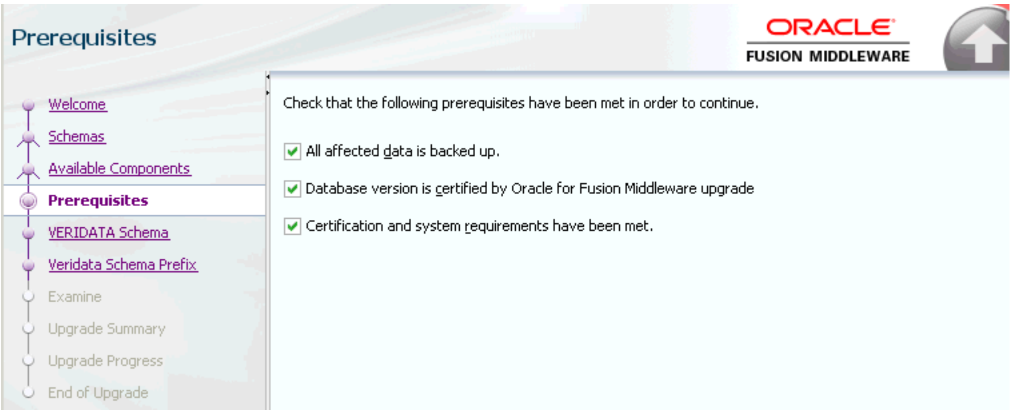


Select components to upgrade. Mandatory components and dependencies will be automatically selected.

- □ Available Components
  - □ Common Infrastructure Services
  - □ Oracle Audit Services
  - ☑ Oracle GoldenGate Veridata
  - □ Oracle Metadata Services
  - □ Oracle Platform Security Services
  - □ Oracle WebLogicServer
  - □ User Messaging Service

4. The Prerequisites screen lists items that you must check and verify before you can proceed.

   You must check the boxes before you can continue. The Upgrade Assistant will not verify that the prerequisites have been met.



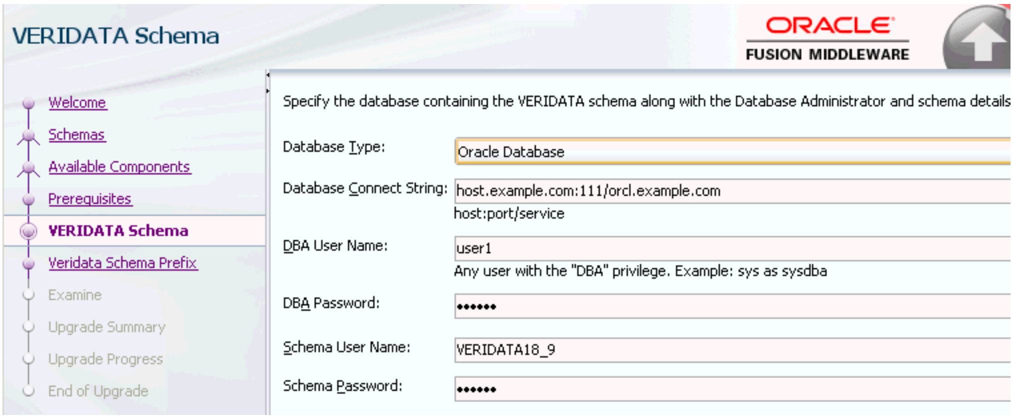
Check that the following prerequisites have been met in order to continue.

☑ All affected data is backed up.

☑ Database version is certified by Oracle for Fusion Middleware upgrade

☑ Certification and system requirements have been met.

5. On the **VERIDATA Schema** page, enter the connection credentials for the database containing the schema you want to upgrade.



Specify the database containing the VERIDATA schema along with the Database Administrator and schema details

Database Type: Oracle Database

Database Connect String: host.example.com:111/orcl.example.com
host:port/service

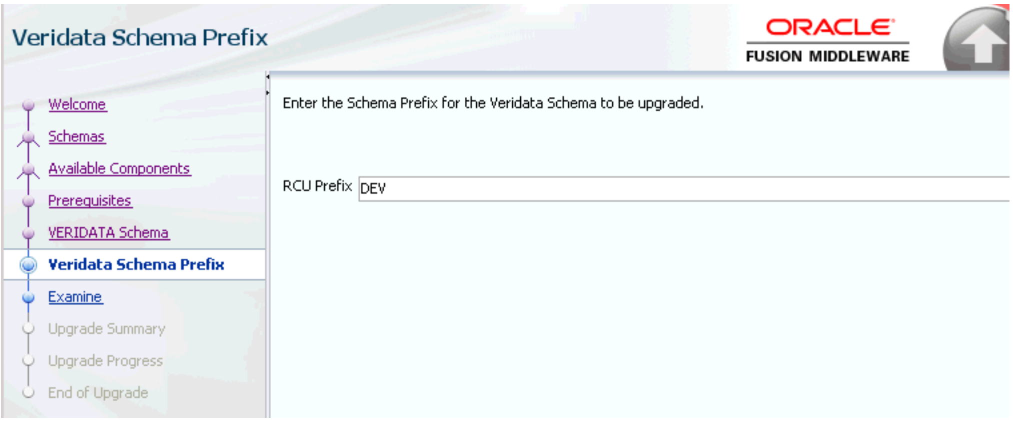
DBA User Name: user1
Any user with the "DBA" privilege. Example: sys as sysdba

DBA Password: ••••••

Schema User Name: VERIDATA18_9

Schema Password: ••••••

6. On the Veridata Schema Prefix page, enter the prefix that was entered while creating the schema using RCU. See Section 2.2, "Creating Repository Schemas using Repository Creation Utility".



Enter the Schema Prefix for the Veridata Schema to be upgraded.

RCU Prefix DEV

7. On the Examine screen, the Upgrade Assistant performs a series of validations before upgrading the selected components. Ensure that all validations have succeeded. Click Next.

8. Click Upgrade on the Upgrade Summary screen to begin the upgrade. The Upgrade Progress screens shows information about the progress of the upgrade, and the Upgrade Success screen summarizes the upgrade.

To verify the schema upgrade, follow the procedure as described in Section 3.2, "Verifying the Schema Upgrade".

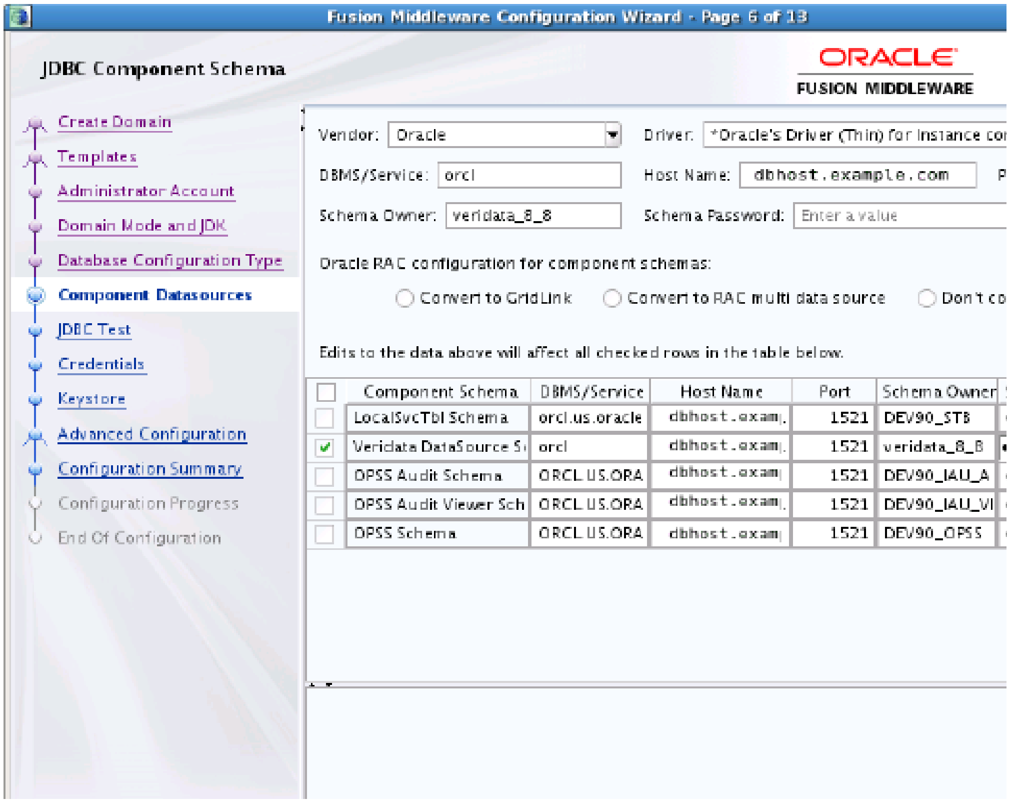
To troubleshoot issues that might occur during schema upgrade, review the logs as described in Section 3.6, "Troubleshooting your Upgrade".

2.4 Creating the Veridata Domain using the Upgraded Schema

To create a WebLogic Server domain for Veridata, follow the procedure described in "Configure a Domain for Oracle GoldenGate Veridata" in *Installing and Configuring Oracle GoldenGate Veridata*.

Note: Provide the upgraded Veridata schema details on the JDBC Component Schema page of the Configuration Wizard as shown in Figure 2–3.

Figure 2–3 Schema Selection in the Configuration Wizard



Vendor: Oracle

Driver: *Oracle's Driver (Thin) for Instance co

DBMS/Service: orcl

Host Name: dbhost.example.com

Schema Owner: veridata_8_8

Schema Password: Enter a value

Oracle RAC configuration for component schemas:

○ Convert to GridLink ○ Convert to RAC multi data source ○ Don't co

Edits to the data above will affect all checked rows in the table below.

□	Component Schema	DBMS/Service	Host Name	Port	Schema Owner
□	LocalSvcTbl Schema	orcl.us.oracle	dbhost.exam	1521	DEV90_STB
☑	Veridata DataSource S	orcl	dbhost.exam	1521	veridata_8_8
□	OPSS Audit Schema	ORCL.US.ORA	dbhost.exam	1521	DEV90_IAU_A
□	OPSS Audit Viewer Sch	ORCL.US.ORA	dbhost.exam	1521	DEV90_IAU_VI
□	OPSS Schema	ORCL.US.ORA	dbhost.exam	1521	DEV90_OPSS

2.5 Upgrading Veridata Configuration by Using the Upgrade Assistant

To upgrade the Veridata repository schema using the Upgrade Assistant follow the instructions below:

1. To start the Upgrade Assistant, go to the *`ORACLE_HOME`*`/oracle_common/upgrade/bin` directory, and enter the following command:

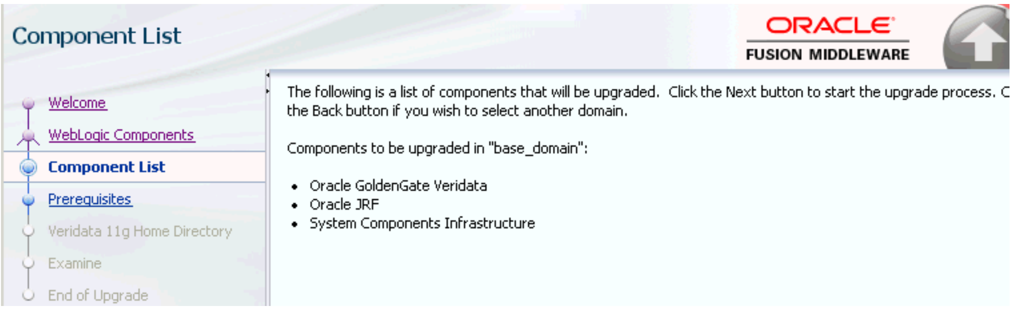
   On UNIX: `./ua`

   On Windows: `ua.bat`

2. Select **WebLogic Component Configurations** on the Upgrade Operation page.

3. Select the Veridata domain to be upgraded. See Section 2.4, "Creating the Veridata Domain using the Upgraded Schema".

4. The Component List page displays a list of components that will be upgraded. Review the list and click Next to start the upgrade process.



The following is a list of components that will be upgraded. Click the Next button to start the upgrade process. C the Back button if you wish to select another domain.

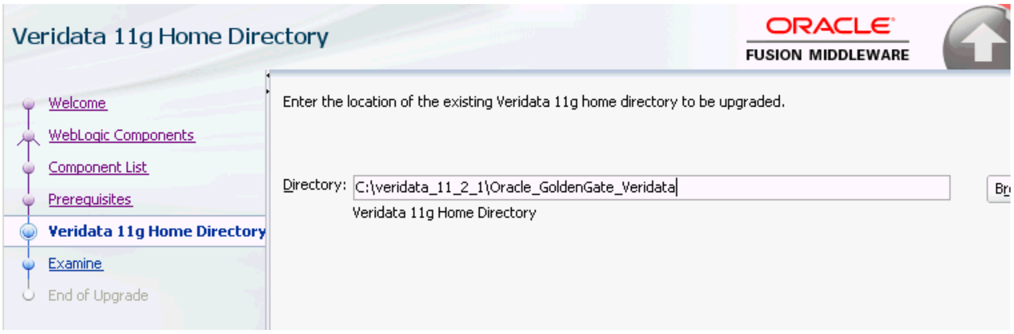
Components to be upgraded in "base_domain":

- Oracle GoldenGate Veridata
- Oracle JRF
- System Components Infrastructure

5. The Prerequisites screen lists items that you must check and verify before you can proceed.

   You must check the boxes before you can continue. The Upgrade Assistant will not verify that the prerequisites have been met.

6. On the Veridata 11g Home Directory page, specify the location of the Veridata 11.2 installation.



Enter the location of the existing Veridata 11g home directory to be upgraded.

Directory: C:\veridata_11_2_1\Oracle_GoldenGate_Veridata

Veridata 11g Home Directory

7. On the Examine screen, the Upgrade Assistant performs a series of validations before upgrading the selected components. Ensure that all validations have succeeded. Click Next to start the upgrade process.

8. Click Upgrade on the Upgrade Summary screen to begin the upgrade. The Upgrade Progress screens shows information about the progress of the upgrade, and the Upgrade Success screen summarizes the upgrade.

3 Performing Post-Upgrade Tasks

This chapter describes the post-upgrade tasks that are required for some upgrade scenarios.

This chapter includes the following sections:

- Section 3.1, "Deploying and Configuring Veridata Agent"
- Section 3.2, "Verifying the Schema Upgrade"
- Section 3.3, "Verifying the Veridata Configuration Upgrade"
- Section 3.4, "Creating Veridata Users"
- Section 3.5, "Verifying Veridata Report Files"
- Section 3.6, "Troubleshooting your Upgrade"

3.1 Deploying and Configuring Veridata Agent

After upgrading to 12c (12.1.3), you must deploy and configure the Oracle GoldenGate Veridata Agent at a non-Oracle Home location. For more information, see "Deploying and Configuring Oracle GoldenGate Veridata Agent" in *Installing and Configuring Oracle GoldenGate Veridata*.

To configure SSL communication between the Veridata Agent and the Server, set the SSL properties for the agent and follow the procedure as described in "Configuring Oracle GoldenGate Veridata Security" chapter in *Administering Oracle GoldenGate Veridata*.

3.2 Verifying the Schema Upgrade

To verify that the schema has been upgraded, run the following SQL query on the `schema_version_registry` table from the database host by an `admin` or `root` user:

Oracle:

```
SELECT COMP_ID,COMP_NAME,MRC_NAME,OWNER,VERSION,STATUS,UPGRADED FROM schema_
version_registry where MRC_NAME='Prefix Specified in RCU';
```

Example 3–1 Sample output of the query on Oracle database

```
SQL> SELECT COMP_ID,COMP_NAME,MRC_NAME,OWNER,VERSION,STATUS,UPGRADED FROM schema_
version_registry where MRC_NAME='DEV100';
```

COMP_ID	COMP_NAME	MRC_ NAME	OWNER	VERSION	STATUS	UPGRADED
IAU	Audit Service	DEV100	DEV100_ IAU	12.1.3.0.0	VALID	N
IAU_ APPEND	Audit Service Append	DEV100	DEV100_ IAU_ APPEND	12.1.3.0.0	VALID	N
IAU_ VIEWER	Audit Service Viewer	DEV100	DEV100_ IAU_ VIEWER	12.1.3.0.0	VALID	N
OPSS	Oracle Platform Security Services	DEV100	DEV100_ OPSS	12.1.3.0.0	VALID	N
STB	Service Table	DEV100	DEV100_ STB	12.1.3.0.0	VALID	N
VERIDATA	Veridata Repository	DEV100	VERIDATAONE	12.1.3.0.0	VALID	Y

MySQL:

```
SELECT COMP_ID,COMP_NAME,MRC_NAME,OWNER,VERSION,STATUS,UPGRADED FROM DB_
NAME.schema_version_registry where MRC_NAME='Prefix Specified in RCU';
```

DB_NAME is the name of the database you specified while running the RCU.

Example 3–2 Sample output of the query on MySQL database

```
SELECT COMP_ID,COMP_NAME,MRC_NAME,OWNER,VERSION,STATUS,UPGRADED FROM veridata_
test.schema_version_registry where MRC_NAME='DEV';
```

COMP_ID	COMP_NAME	MRC_ NAME	OWNER	VERSION	STATUS	UPGRADED
STB	Service Table	DEV	DEV_STB	12.1.3.0.0	VALID	N
VERIDATA	Veridata Repository	DEV100	VERIDATAONE	12.1.3.0.0	VALID	Y

SQL Server:

```
SELECT COMP_ID,COMP_NAME,MRC_NAME,OWNER,VERSION,STATUS,UPGRADED FROM DB_
NAME.dbo.schema_version_registry where MRC_NAME='Prefix Specified in RCU';
```

DB_NAME is the name of the database you specified while running the RCU.

Example 3–3 Sample output of the query on SQL Server database

```
SELECT COMP_ID, comp_name, mrc_name, owner, version, status, upgraded FROM
veridata12.dbo.schema_version_registry where mrc_name='DEV';
```

COMP_ID	COMP_NAME	MRC_ NAME	OWNER	VERSION	STATUS	UPGRADED
STB	Service Table	DEV	DEV_STB	12.1.3.0.0	VALID	N
VERIDATA	Veridata Repository	DEV100	VERIDATAONE	12.1.3.0.0	VALID	Y

3.3 Verifying the Veridata Configuration Upgrade

After upgrading to Oracle GoldenGate Veridata 12c (12.1.3), you can verify the following and confirm whether the upgrade was successful:

- **Verify the configuration properties**: After upgrade, the configuration properties are copied to the `veridata.cfg` file located in the `VERIDATA_DOMAIN_HOME/config/veridata` directory. If any of the default values were changed in the 11g installation before upgrade, verify the values are copied to the `veridata.cfg` file after upgrade.

> **Note:** The `veridata.cfg` file contains only the properties which are valid in 12c. Deprecated properties are not copied after upgrade.

- **Verify the connections**: After upgrade, Veridata users can login to the Veridata application and verify the existing connections. If the password fields are not empty and if the Test Connection succeeds, then the upgrade is successful.

3.4 Creating Veridata Users

After upgrading to Oracle GoldenGate Veridata 12c (12.1.3), the Veridata 11g users cannot login to the Veridata application. You must create Veridata users and assign privileges by using the Oracle WebLogic Administration Console.

For more information, see "Securing Access to Oracle GoldenGate Veridata by Defining User Roles" in *Administering Oracle GoldenGate Veridata*.

3.5 Verifying Veridata Report Files

The Veridata report file location is defined by the `server.veridata_data` property in the `veridata.cfg` file. In 12c (12.1.3), the default location is *`VERIDATA_DOMAIN_HOME`*`/veridata/reports`. For more information about the `server.veridata_data` property see "Server Parameters" section in *Administering Oracle GoldenGate Veridata*.

Before upgrading to 12c, if the report files were stored in the default report file location of the Veridata 11g installation (`shared/data` directory), the default location will be updated as *`VERIDATA_DOMAIN_HOME`*`/veridata/reports` after upgrade.

If you have specified another location for the report files in the Veridata 11g installation then the same location will be used in Veridata 12c after upgrade. To access the old report files after upgrade, you should retain the directory structure of the report files.

Note that the report location of 11g reports are not updated after upgrade.

3.6 Troubleshooting your Upgrade

If the upgrade process fails, you must close the Upgrade Assistant, correct the issue, and then restart the Upgrade Assistant.

Reviewing Log Files

By default, upgrade log files are created and saved in the *`ORACLE_HOME`*`/oracle_common/upgrade/logs/` directory. You can review these log files to diagnose problems that might have occurred during the upgrade process.

The log file location is displayed on the console where you run the Upgrade Assistant.

For Veridata schema upgrade, only one log file is created, but when you upgrade Veridata configuration, two log files are created. One is the default log file and the other is an output log file. The output log file is created at the same location but with an `.out` extension. The file name remains the same

For more troubleshooting information, see "Troubleshooting Your Upgrade" in *Upgrading with the Upgrade Assistant*.