

# Oracle® Fusion Middleware

## Installing and Upgrading Oracle GoldenGate for Big Data



21c (21.1.0.0.0)

F29113-01

June 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Installing and Upgrading Oracle GoldenGate for Big Data, 21c (21.1.0.0.0)

F29113-01

Copyright © 2015, 2021, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs 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" or "commercial computer software documentation" 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 and Java 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

---

Audience	vi
Documentation Accessibility	vi
Conventions	vi
Related Information	vii

## 1 Overview

---

## 2 Installing Oracle GoldenGate Classic for Big Data

---

2.1 What's Supported in Oracle GoldenGate for Big Data?	2-1
2.1.1 Verifying Certification and System Requirements	2-1
2.1.2 Understanding Handler Compatibility	2-2
2.1.3 What are the Additional Support Considerations?	2-2
2.2 Preparing for Installation	2-4
2.2.1 Downloading Oracle GoldenGate for Big Data	2-4
2.2.2 Installation Overview	2-5
2.2.2.1 Contents of the Installation ZIP File	2-5
2.2.2.2 Using the Generic Build of Oracle GoldenGate	2-5
2.2.2.3 Considerations for Using a Custom Build for a Big Data Instance of Oracle GoldenGate	2-6
2.2.2.4 Installing to a Non-Generic Instance of Oracle GoldenGate	2-6
2.2.3 Directory Structure	2-7
2.2.4 Setting up Environmental Variables	2-8
2.2.4.1 Java on Linux/UNIX	2-9
2.2.4.2 Java on Windows	2-9
2.3 Installation Steps	2-9
2.4 Getting Started with Oracle GoldenGate (Classic) for Big Data	2-10
2.4.1 Setting Up the Java Runtime Environment	2-11
2.4.2 About Oracle GoldenGate Properties Files	2-11
2.4.3 Grouping Transactions	2-11
2.5 Configuring Oracle GoldenGate for Big Data	2-11

2.5.1	Running with Replicat	2-12
2.5.1.1	Configuring Replicat	2-12
2.5.1.2	Adding the Replicat Process	2-13
2.5.1.3	Replicat Grouping	2-13
2.5.1.4	About Replicat Checkpointing	2-13
2.5.1.5	About Initial Load Support	2-13
2.5.1.6	About the Unsupported Replicat Features	2-13
2.5.1.7	How the Mapping Functionality Works	2-14
2.5.2	Overview of Logging	2-14
2.5.2.1	About Replicat Process Logging	2-14
2.5.2.2	About Java Layer Logging	2-14
2.5.3	About Schema Evolution and Metadata Change Events	2-15
2.5.4	About Configuration Property CDATA[] Wrapping	2-16
2.5.5	Using Regular Expression Search and Replace	2-16
2.5.5.1	Using Schema Data Replace	2-17
2.5.5.2	Using Content Data Replace	2-17
2.5.6	Scaling Oracle GoldenGate for Big Data Delivery	2-18
2.5.7	Using Identities in Oracle GoldenGate Credential Store	2-21
2.5.7.1	Creating a Credential Store	2-21
2.5.7.2	Adding Users to a Credential Store	2-22
2.5.7.3	Configuring Properties to Access the Credential Store	2-22

## 3 Installing Oracle GoldenGate Microservices for Big Data

---

3.1	Installing Oracle GoldenGate MA for Big Data Using the UI	3-1
3.2	Silent Installation	3-2
3.3	Setting Up Secure or Non-Secure Deployments	3-3
3.3.1	How to Add Secure or Non-Secure Deployments	3-3
3.3.2	How to Remove a Deployment	3-9
3.3.2.1	How to Remove a Deployment: GUI	3-9
3.3.2.2	How to Remove a Deployment: Silent Mode	3-10

## 4 Upgrading Oracle GoldenGate Classic for Big Data

---

4.1	Upgrading by Overwriting the Existing Installation	4-1
4.2	Upgrading by Installing into a New Directory	4-2
4.3	Switching Existing Extract Processes to Replicat Processes	4-3

## 5 Upgrading Oracle GoldenGate Microservices for Big Data

---

5.1	Obtaining the Oracle GoldenGate Distribution	5-1
5.2	Scope of Upgrade	5-2

5.2.1	Replicat Upgrade Considerations	5-2
5.3	Upgrading Oracle GoldenGate Microservices – GUI Based	5-2
5.4	Upgrading Oracle GoldenGate Microservices Using REST APIs	5-3

## 6 Setting Up Oracle GoldenGate for Big Data in a High Availability Environment

---

6.1	Running Oracle GoldenGate for Big Data from a Single Instance	6-1
6.2	Running Oracle GoldenGate for Big Data on a Cluster of Servers	6-1
6.3	Shared Storage	6-2

## 7 Dependency Downloader

---

7.1	Dependency Downloader Setup	7-1
7.2	Running the Dependency Downloader Scripts	7-2
7.3	Dependency Downloader Scripts	7-3

# Preface

Understanding Oracle GoldenGate21c (21.1.0.0.0) describes data replication concepts, Oracle GoldenGate Classic Architecture, Oracle GoldenGate Microservices, and the architecture components. Using these concepts you can implement a data replication solution using Oracle GoldenGate.

- [Audience](#)
- [Documentation Accessibility](#)
- [Conventions](#)
- [Related Information](#)

## Audience

This guide is intended for system administrators who are configuring and running Oracle GoldenGate for Big Data.

## Documentation Accessibility

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

### Accessible Access to Oracle Support

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

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	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.

## Related Information

The Oracle GoldenGate Product Documentation Libraries are found at:

[Oracle GoldenGate Documentation](#)

Oracle GoldenGate for Big Data Documentation:

<https://docs.oracle.com/en/middleware/goldengate/big-data/index.html>

For additional information on Oracle GoldenGate, refer to:

<https://www.oracle.com/middleware/technologies/goldengate.html>

[Oracle Database High Availability](#)

# 1

## Overview

Oracle GoldenGate for Big Data supports specific components - the handlers, which are compatible with clearly defined software versions.

For more information about what is supported for Oracle GoldenGate Classic, see *Understanding What's Supported* in the *Oracle GoldenGate for Big Data Installation and Upgrading Guide*. For more information about configuring Oracle GoldenGate for Big Data, see *Configuring Oracle GoldenGate for Big Data* in the *Oracle GoldenGate for Big Data Installation and Upgrading Guide*. For more information about what is supported for Oracle GoldenGate Microservices, see [What is Required?](#) in the *Installing Oracle GoldenGate* guide.

### Topics:



# 2

## Installing Oracle GoldenGate Classic for Big Data

This chapter describes how to install a new instance of Oracle GoldenGate for Big Data.

### Topics:

- [What's Supported in Oracle GoldenGate for Big Data?](#)
- [Preparing for Installation](#)
- [Installation Steps](#)
- [Getting Started with Oracle GoldenGate \(Classic\) for Big Data](#)
- [Configuring Oracle GoldenGate for Big Data](#)

### 2.1 What's Supported in Oracle GoldenGate for Big Data?

#### Topics:

- [Verifying Certification and System Requirements](#)  
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.
- [Understanding Handler Compatibility](#)
- [What are the Additional Support Considerations?](#)

#### 2.1.1 Verifying Certification and System Requirements

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.

##### 1. Verifying that your environment meets certification requirements:

Make sure that you install your product on a supported hardware and software configuration. See the certification document for your release on the [Oracle Fusion Middleware Supported System Configuration](#) page.

Oracle has tested and verified the performance of your product on all certified systems and environments. Whenever new certifications are released, they are added to the certification document right away. New certifications can be released at any time. Therefore, the certification documents are kept outside the documentation libraries and are available on Oracle Technology Network.

##### 2. Using the system requirements document to verify certification:

Oracle recommends that you use the [Oracle Fusion Middleware Supported System Configuration](#) document to verify that the certification requirements are met. For example, if the certification document indicates that your product is certified for installation on 64-Bit Oracle Linux 6.5, use this document to verify that your system meets the required

minimum specifications. These include disk space, available memory, specific platform packages and patches, and other operating system-specific requirements. System requirements can change in the future. Therefore, the system requirement documents are kept outside of the documentation libraries and are available on Oracle Technology Network.

### 3. Verifying interoperability among multiple products:

To learn how to install and run multiple Fusion Middleware products from the same release or mixed releases with each other, see [Oracle Fusion Middleware Supported System Configuration](#) in *Oracle Fusion Middleware Understanding Interoperability and Compatibility*.

The compatibility of the Oracle GoldenGate for Big Data Handlers with the various data collections, including distributions, database releases, and drivers is included in the certification document.

## 2.1.2 Understanding Handler Compatibility

For more information, see the Certification Matrix.

## 2.1.3 What are the Additional Support Considerations?

This section describes additional Oracle GoldenGate for Big Data Handlers additional support considerations.

### Pluggable Formatters—Support

The handlers support the Pluggable Formatters as follows:

- The HDFS Handler supports all of the pluggable formatters.
- Pluggable formatters are not applicable to the HBase Handler. Data is streamed to HBase using the proprietary HBase client interface.
- The Kafka Handler supports all of the pluggable formatters.
- The Kafka Connect Handler does *not* support pluggable formatters. You can convert data to JSON or Avro using Kafka Connect data converters.
- The Kinesis Streams Handler supports all of the pluggable formatters described in the Using the Pluggable Formatters topic in the *Oracle GoldenGate for Big Data User Guide*.
- The Cassandra, MongoDB, and JDBC Handlers do *not* use a pluggable formatter.

### Java Delivery Using Extract

Java Delivery using Extract is not supported. Support for Java Delivery is only supported using the Replicat process. Replicat provides better performance, better support for checkpointing, and better control of transaction grouping.

### MongoDB Handler—Support

- The handler can only replicate unique rows from source table. If a source table has no primary key defined and has duplicate rows, replicating the duplicate rows to the MongoDB target results in a duplicate key error and the Replicat process abends.
- Missed updates and deletes are undetected so are ignored.

- Untested with sharded collections.
- Only supports date and time data types with millisecond precision. These values from a trail with microseconds or nanoseconds precision are truncated to millisecond precision.
- The `datetime` data type with `timezone` in the trail is not supported.
- A maximum BSON document size of 16 MB. If the trail record size exceeds this limit, the handler cannot replicate the record.
- No DDL propagation.
- No truncate operation.

### JDBC Handler—Support

- The JDBC handler uses the generic JDBC API, which means any target database with a JDBC driver implementation should be able to use this handler. There are a myriad of different databases that support the JDBC API and Oracle cannot certify the JDBC Handler for all targets. Oracle has certified the JDBC Handler for the following RDBMS targets:

- Oracle
- MySQL
- Netezza
- Redshift

- The handler supports Replicat using the `REPERROR` and `HANDLECOLLISIONS` parameters, see *Reference for Oracle GoldenGate*.
- The database metadata retrieved through the Redshift JDBC driver has known constraints, see *Release Notes for Oracle GoldenGate for Big Data*.

Redshift target table names in the Replicat parameter file must be in lower case and double quoted. For example:

```
MAP SourceSchema.SourceTable, target "public"."targetable";
```

- DDL operations are ignored by default and are logged with a `WARN` level.
- Coordinated Replicat is a multithreaded process that applies transactions in parallel instead of serially. Each thread handles all of the filtering, mapping, conversion, SQL construction, and error handling for its assigned workload. A coordinator thread coordinates transactions across threads to account for dependencies. It ensures that DML is applied in a synchronized manner preventing certain DMLs from occurring on the same object at the same time due to row locking, block locking, or table locking issues based on database specific rules. If there are database locking issue, then Coordinated Replicat performance can be extremely slow or pauses.

### DDL Event Handling

Only the `TRUNCATE TABLE` DDL statement is supported. All other DDL statements, such as `CREATE TABLE`, `CREATE INDEX`, and `DROP TABLE` are ignored.

You can use the `TRUNCATE` statements one of these ways:

- In a DDL statement, `TRUNCATE TABLE`, `ALTER TABLE TRUNCATE PARTITION`, and other DDL `TRUNCATE` statements. This uses the `DDL` parameter.

- Standalone TRUNCATE support, which just has TRUNCATE TABLE. This uses the GETTRUNCATES parameter.

## 2.2 Preparing for Installation

Prepare your Java environment by ensuring that you have the correct version of Java installed, and that the environmental variables have been set up and configured correctly.

- [Downloading Oracle GoldenGate for Big Data](#)
- [Installation Overview](#)
- [Directory Structure](#)
- [Setting up Environmental Variables](#)

### 2.2.1 Downloading Oracle GoldenGate for Big Data

Oracle GoldenGate (both Classic and Microservices) for Big Data are available for Windows, Linux, and UNIX. To download, first visit the Oracle support site to see if there is a patch available for your operating system and architecture.

#### Note:

If you are not planning to use the generic build included in the installation, ensure that the major release of the Oracle GoldenGate for Big Data build you download matches (or is known to be compatible with) the major release of the Oracle GoldenGate instance that will be used with it.

1. Navigate to <http://support.oracle.com>.
2. Sign in with your Oracle ID and password.
3. Select the **Patches and Upgrades** tab.
4. On the **Search** tab, click **Product or Family**.
5. In the **Product** field, type **Oracle GoldenGate for Big Data**.
6. From the **Release** drop-down list, select the release version that you want to download.
7. Make sure Platform is displayed as the default in the next field, and then select the platform from the drop-down list.
8. Leave the last field blank.
9. Click **Search**.
10. In the **Advanced Patch Search Results** list, select the available builds that satisfy the criteria that you supplied.
11. In the file **Download** dialog box, click the ZIP file to begin the download.

If patches are not available on the support site, go to the Oracle delivery site for the release download.

1. Navigate to <http://edelivery.oracle.com>.

2. Sign in with your Oracle ID and password.
3. On the Terms and Restrictions page:
  - Accept the **Trial License Agreement** (even if you have a permanent license).
  - Accept the **Export Restrictions**.
  - Click **Continue**.
4. On the **Media Pack Search** page:
  - Select the Oracle Fusion Middleware Product Pack.
  - Select the platform on which you will be installing the software.
  - Click **Go**.
5. In the **Results** list:
  - Select the **Oracle GoldenGate Applications Big Data Media Pack** that you want.
  - Click **Continue**.
6. On the **Download** page:
  - View the *Readme* file.
  - Click **Download** for each component that you want. Follow the automatic download process to transfer the zip file to your system.

## 2.2.2 Installation Overview

This section provides an overview of the installation contents and the Oracle GoldenGate instances used with the Oracle GoldenGate for Big Data.

- [Contents of the Installation ZIP File](#)
- [Using the Generic Build of Oracle GoldenGate](#)
- [Considerations for Using a Custom Build for a Big Data Instance of Oracle GoldenGate](#)
- [Installing to a Non-Generic Instance of Oracle GoldenGate](#)

### 2.2.2.1 Contents of the Installation ZIP File

The Oracle GoldenGate for Big Data installation ZIP file contains:

- Oracle GoldenGate Java Adapter
- A version of Oracle GoldenGate designed to stream data to Big Data targets. This version is labeled *generic* because it is not specific to any database, but it is platform dependent.

### 2.2.2.2 Using the Generic Build of Oracle GoldenGate

For JMS capture, the Java Adapter must run in the generic build of Oracle GoldenGate. However, the generic build is not required when using the adapter for delivery of trail data to a target; in this case, the Java Adapter can be used with any database version of Oracle GoldenGate.

### 2.2.2.3 Considerations for Using a Custom Build for a Big Data Instance of Oracle GoldenGate

There are both advantages and disadvantages to installing a custom build for a Big Data Oracle GoldenGate instance. Also, there are limitations in the releases of Oracle GoldenGate that are compatible with releases of the Big Data.

#### Advantages

- The non-generic instance allows you to configure Extract to login to the database for metadata. This removes the need to use a source definitions file that must be synchronized your the source database DDL.
- There is no need to manage two separate versions of Oracle GoldenGate when doing database capture and JMS delivery on the same server.

#### Disadvantages

- If you need to patch Oracle GoldenGate core instance, you must also copy the Big Data into the new patched installation of Oracle GoldenGate.
- The Oracle GoldenGate for Big Data are only tested and certified with the generic version of Oracle GoldenGate core. New patches of the core can trigger incompatibilities.

#### Limitations

- The Replicat module to write to Big Data targets is only available in the Generic Oracle GoldenGate distribution.
- The generic build must be used with JMS capture, as this is the only version of Extract that is capable of loading the VAM.
- A `DEFGEN` utility is not included with the Big Data. To generate source definitions, you will need a version of Oracle GoldenGate that is built specifically for your database type.

### 2.2.2.4 Installing to a Non-Generic Instance of Oracle GoldenGate

If you decide to install the Java user exit to a non-generic instance of Oracle GoldenGate, unzip to a temporary location first and then copy the adapter files to your Oracle GoldenGate installation location.

To install the Java user exit to a non-generic instance of Oracle GoldenGate:

1. Navigate to the Oracle GoldenGate installation directory, for example `C:/ggs`.
2. Create a temporary directory, and extract the Java user exit ZIP file into this into sub directory within it, for example `ggjava`.
3. Copy or move the files from the `ggjava` sub directory and shared libraries into the Oracle GoldenGate installation directory (`C:/ggs`).

 **Note:**

You need not copy the shared library `ggjava_vam` because, it only works with the generic build.

## 2.2.3 Directory Structure

The following table is a sample that includes the subdirectories and files that result from unzipping the installation file and creating the subdirectories. The following conventions have been used:

- Subdirectories are enclosed in square brackets []
- Levels are indicated by a pipe and hyphen |-
- The `Internal` notation indicates a read-only directory that should not be modified
- Text files (\*.txt) are not included in the list
- Oracle GoldenGate utilities, such as Defgen, Logdump, and Keygen, are not included in the list

**Table 2-1 Sample installation directory structure**

Directory	Explanation
[gg_install_dir]	Oracle GoldenGate installation directory, such as <code>C:/ggs</code> on Windows or <code>/home/user/ggs</code> on UNIX.
-ggsci	Command line interface used to start, stop, and manage processes.
-mgr	Manager process.
-extract	Extract process that will start the Java application.
-replicat	Replicat process that will start the Java application.
-[UserExitExamples]	Sample C programming language user exit code examples.
-[dirprm]	Subdirectory that holds all the parameter and property files created by the user, for example:  <code>javaue.prm</code> <code>javaue.properties</code> <code>jmsvam.prm</code> <code>jmsvam.properties</code> <code>ffwriter.prm</code>
-[dirdef]	Subdirectory that holds source definitions files (*.def) defining the metadata of the trail: <ul style="list-style-type: none"> <li>• Created by the core utility for the user exit trail data.</li> </ul>
-[dirdat]	Subdirectory that holds the trail files produced by the VAM Extract or read by the user exit Extract.

**Table 2-1 (Cont.) Sample installation directory structure**

Directory	Explanation
-[dirrpt]	Subdirectory that holds log and report files.
-[dirchk]	Internal Subdirectory that holds checkpoint files.
-[dirpcs]	Internal Subdirectory that holds process status files.
-[dirjar]	Internal Subdirectory that holds Oracle GoldenGate Monitor jar files.
-[ggjava]	Internal Installation directory for Java jars. Read-only; do not modify.
-[ggjava.jar]	The main Java application jar that defines the class path and dependencies.
-[resources]	Subdirectory that contains all ggjava.jar dependencies. Includes subdirectories for: <ul style="list-style-type: none"> <li>[class] - properties and resources</li> <li>[lib] - application jars required by ggjava.jar</li> </ul>
-[ggjava_vam.dll]	The VAM shared library. This is libggjava_vam.so on UNIX.
-[ggjava.dll]	Used by the Replicat based delivery process. This is libggjava.so on UNIX.
-. . .	Other subdirectories and files included in the installation or created later.

## 2.2.4 Setting up Environmental Variables

To configure your Java environment for Oracle GoldenGate for Java:

- The `PATH` environmental variable should be configured to find your Java Runtime
- The shared (dynamically linked) Java virtual machine (JVM) library must also be found.

On Windows, these environmental variables should be set as system variables; on Linux/UNIX, they should be set globally or for the user running the Oracle GoldenGate processes. Examples of setting these environmental variables for Windows, UNIX, and Linux are in the following sections.

### Note:

There may be two versions of the `JAVA_HOME/.../client`, and another in `JAVA_HOME/.../server`. For improved performance, use the server version, if it is available. On Windows, only the client JVM may be there if only the JRE was installed (and not the JDK).



- [Java on Linux/UNIX](#)
- [Java on Windows](#)

### 2.2.4.1 Java on Linux/UNIX

Configure the environment to find the JRE in the `PATH`, and the JVM shared library, using the appropriate environmental variable for your system. For example, on Linux (and Solaris), set `LD_LIBRARY_PATH` to include the directory containing the JVM shared library as follows (for `sh/ksh/bash`):

**Note:**

On AIX platforms, you set `LIBPATH=`. On HP-UX IA64, you set `SHLIB_PATH=`.

#### Example 2-1 Configuring path for Java on Linux

```
export JAVA_HOME=/opt/jdk1.8
export PATH=$JAVA_HOME/bin:$PATH
export LD_LIBRARY_PATH=$JAVA_HOME/jre/lib/amd64/server:$LD_LIBRARY_PATH
```

In this example, the directory `$JAVA_HOME/jre/lib/i386/server` should contain the `libjvm.so` and `libjsig.so` files. The actual directory containing the JVM library depends on the operating system and if the 64-bit JVM is being used.

Verify the environment settings by opening a command prompt and checking the Java version as in this example:

```
$ java -version
java version "1.8.0_92"
Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
```

### 2.2.4.2 Java on Windows

After Java is installed, configure the `PATH` to find the JRE and JVM DLL (`jvm.dll`):

#### Example 2-2 Configuring Path for Java on Windows

```
set JAVA_HOME=C:\Program Files\Java\jdk1.8.0
set PATH=%JAVA_HOME%\bin;%PATH%
set PATH=%JAVA_HOME%\jre\bin\server;%PATH%
```

In the example above, the directory `%JAVA_HOME%\jre\bin\server` should contain the file `jvm.dll`.

Verify the environment settings by opening a command prompt and checking the Java version as in this example:

```
C:\> java -version
java version "1.8.0_92" Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
```

## 2.3 Installation Steps

Perform the following steps to install the Oracle GoldenGate for Big Data:

1. Create an installation directory that has no spaces in its name. Then extract the ZIP file into this new installation directory. For example:

```
Shell> mkdir installation_directory
Shell> cp path/to/installation_zip installation_directory
Shell> cd installation_directory
Shell> unzip installation_zip
```

If you are on Linux or UNIX, run:

```
Shell> tar -xf installation_tar
```

This downloads the files into several of the subdirectories [Directory Structure](#) .

2. Stay on the installation directory and bring up GGSCI to create the remaining subdirectories in the installation location.

```
Shell> ggsci
GGSCI> CREATE SUBDIRS
```

3. Create a Manager parameter file:

```
GGSCI> EDIT PARAM MGR
```

4. Specify a port for the Manager to listen on by using the editor to add a line to the Manager parameter file. For example:

```
PORT 7801
```

5. If you are on Windows and running Manager as a service, set the system variable PATH to include `jvm.dll`, then delete the Manager service and re-add it.

6. Go to GGSCI, start the Manager, and check to see that it is running:

```
GGSCI>START MGR
GGSCI>INFO MGR
```

#### Note:

To check for environmental variable problems locating the JVM at runtime:

- Add the parameter `GETENV(PATH)` for Windows or `GETENV(LD_LIBRARY_PATH)` for UNIX to the Replicat parameter file.
- Start the Replicat process
- Check the output for the report using the GGSCI command: `SEND REPLICAT group_name REPORT`

## 2.4 Getting Started with Oracle GoldenGate (Classic) for Big Data

This topic lists the various tasks that you need to preform to set up Oracle GoldenGate (Classic) for Big Data integrations with Big Data targets.

- [Setting Up the Java Runtime Environment](#)
- [About Oracle GoldenGate Properties Files](#)

- [Grouping Transactions](#)

## 2.4.1 Setting Up the Java Runtime Environment

The Oracle GoldenGate for Big Data integrations create an instance of the Java virtual machine at runtime. Oracle GoldenGate for Big Data requires that you install Oracle Java 8 Java Runtime Environment (JRE) at a minimum.

Oracle recommends that you set the `JAVA_HOME` environment variable to point to Java 8 installation directory. Additionally, the Java Delivery process needs to load the `libjvm.so` and `libjsig.so` Java shared libraries. These libraries are installed as part of the JRE. The location of these shared libraries need to be resolved and the appropriate environmental variable set to resolve the dynamic libraries needs to be set so the libraries can be loaded at runtime (that is, `LD_LIBRARY_PATH`, `PATH`, or `LIBPATH`).

## 2.4.2 About Oracle GoldenGate Properties Files

There are two Oracle GoldenGate properties files required to run the Oracle GoldenGate Java Deliver user exit (alternatively called the Oracle GoldenGate Java Adapter). It is the Oracle GoldenGate Java Delivery that hosts Java integrations including the Big Data integrations. A Replicat properties file is required in order to run either process. The required naming convention for the Replicat file name is the `process_name.prm`. The exit syntax in the Replicat properties file provides the name and location of the Java Adapter properties file. It is the Java Adapter properties file that contains the configuration properties for the Java adapter include GoldenGate for Big Data integrations. The Replicat and Java Adapters properties files are required to run Oracle GoldenGate for Big Data integrations.

Alternatively the Java Adapters properties can be resolved using the default syntax, `process_name.properties`. If you use the default naming for the Java Adapter properties file then the name of the Java Adapter properties file can be omitted from the Replicat properties file.

Samples of the properties files for Oracle GoldenGate for Big Data integrations can be found in the subdirectories of the following directory:

```
GoldenGate_install_dir/AdapterExamples/big-data
```

## 2.4.3 Grouping Transactions

The principal way to improve performance in Oracle GoldenGate for Big Data integrations is using transaction grouping. In transaction grouping, the operations of multiple transactions are grouped together in a single larger transaction. The application of a larger grouped transaction is typically much more efficient than the application of individual smaller transactions. Transaction grouping is possible with the Replicat process discussed in [Running with Replicat](#).

## 2.5 Configuring Oracle GoldenGate for Big Data

This topic describes how to configure Oracle GoldenGate for Big Data Handlers.

- [Running with Replicat](#)  
You need to run the Java Adapter with the Oracle GoldenGate Replicat process to begin configuring Oracle GoldenGate for Big Data.

- [Overview of Logging](#)  
Logging is essential to troubleshooting Oracle GoldenGate for Big Data integrations with Big Data targets.
- [About Schema Evolution and Metadata Change Events](#)
- [About Configuration Property CDATA\[\] Wrapping](#)
- [Using Regular Expression Search and Replace](#)
- [Scaling Oracle GoldenGate for Big Data Delivery](#)
- [Using Identities in Oracle GoldenGate Credential Store](#)  
The Oracle GoldenGate credential store manages user IDs and their encrypted passwords (together known as credentials) that are used by Oracle GoldenGate processes to interact with the local database. The credential store eliminates the need to specify user names and clear-text passwords in the Oracle GoldenGate parameter files.

## 2.5.1 Running with Replicat

You need to run the Java Adapter with the Oracle GoldenGate Replicat process to begin configuring Oracle GoldenGate for Big Data.

This topic explains how to run the Java Adapter with the Oracle GoldenGate Replicat process.

- [Configuring Replicat](#)
- [Adding the Replicat Process](#)
- [Replicat Grouping](#)
- [About Replicat Checkpointing](#)
- [About Initial Load Support](#)
- [About the Unsupported Replicat Features](#)
- [How the Mapping Functionality Works](#)

### 2.5.1.1 Configuring Replicat

The following is an example of how you can configure a Replicat process properties file for use with the Java Adapter:

```
REPLICAT hdfs
TARGETDB LIBFILE libggjava.so SET property=dirprm/hdfs.properties
--SOURCEDEFS ./dirdef/dbo.def
DDL INCLUDE ALL
GROUPTRANSOPS 1000
MAPEXCLUDE dbo.excludetable
MAP dbo.*, TARGET dbo.*;
```

The following is explanation of these Replicat configuration entries:

REPLICAT hdfs - The name of the Replicat process.

TARGETDB LIBFILE libggjava.so SET property=dirprm/hdfs.properties - Sets the target database as you exit to libggjava.so and sets the Java Adapters property file to dirprm/hdfs.properties.

--SOURCEDEFS ./dirdef/dbo.def - Sets a source database definitions file. It is commented out because Oracle GoldenGate trail files provide metadata in trail.

GROUPTRANSOPS 1000 - Groups 1000 transactions from the source trail files into a single target transaction. This is the default and improves the performance of Big Data integrations.

MAPEXCLUDE dbo.excludetable - Sets the tables to exclude.

MAP dbo.\*, TARGET dbo.\*; - Sets the mapping of input to output tables.

## 2.5.1.2 Adding the Replicat Process

The command to add and start the Replicat process in `ggsci` is the following:

```
ADD REPLICAT hdfs, EXTTRAIL ./dirdat/gg
START hdfs
```

## 2.5.1.3 Replicat Grouping

The Replicat process provides the Replicat configuration property, `GROUPTRANSOPS`, to control transaction grouping. By default, the Replicat process implements transaction grouping of 1000 source transactions into a single target transaction. If you want to turn off transaction grouping then the `GROUPTRANSOPS` Replicat property should be set to 1.

## 2.5.1.4 About Replicat Checkpointing

In addition to the Replicat checkpoint file `./cpr`, an additional checkpoint file, `dirchk/group.cpj`, is created that contains information similar to `CHECKPOINTTABLE` in Replicat for the database.

## 2.5.1.5 About Initial Load Support

Replicat can already read trail files that come from both the online capture and initial load processes that write to a set of trail files. In addition, Replicat can also be configured to support the delivery of the special run initial load process using `RMTTASK` specification in the Extract parameter file. For more details about configuring the direct load, see [Loading Data with an Oracle GoldenGate Direct Load](#).



### Note:

The `SOURCEDB` or `DBLOGIN` parameter specifications vary depending on your source database.

## 2.5.1.6 About the Unsupported Replicat Features

The following Replicat features are not supported in this release:

- BATCHSQL
- SQLEXEC
- Stored procedure
- Conflict resolution and detection (CDR)

## 2.5.1.7 How the Mapping Functionality Works

The Oracle GoldenGate Replicat process supports mapping functionality to custom target schemas. You must use the Metadata Provider functionality to define a target schema or schemas, and then use the standard Replicat mapping syntax in the Replicat configuration file to define the mapping. For more information about the Replicat mapping syntax in the Replication configuration file, see Mapping and Manipulating Data.

## 2.5.2 Overview of Logging

Logging is essential to troubleshooting Oracle GoldenGate for Big Data integrations with Big Data targets.

This topic details how Oracle GoldenGate for Big Data integration log and the best practices for logging.

- [About Replicat Process Logging](#)
- [About Java Layer Logging](#)

### 2.5.2.1 About Replicat Process Logging

Oracle GoldenGate for Big Data integrations leverage the Java Delivery functionality described in the Delivering Java Messages. In this setup, either a Oracle GoldenGate Replicat process loads a user exit shared library. This shared library then loads a Java virtual machine to thereby interface with targets providing a Java interface. So the flow of data is as follows:

Replicat Process → User Exit → Java Layer

It is important that all layers log correctly so that users can review the logs to troubleshoot new installations and integrations. Additionally, if you have a problem that requires contacting Oracle Support, the log files are a key piece of information to be provided to Oracle Support so that the problem can be efficiently resolved.

A running Replicat process creates or appends log files into the *GoldenGate\_Home/ dirrpt* directory that adheres to the following naming convention: *process\_name.rpt*. If a problem is encountered when deploying a new Oracle GoldenGate process, this is likely the first log file to examine for problems. The Java layer is critical for integrations with Big Data applications.

### 2.5.2.2 About Java Layer Logging

The Oracle GoldenGate for Big Data product provides flexibility for logging from the Java layer. The recommended best practice is to use Log4j logging to log from the Java layer. Enabling simple Log4j logging requires the setting of two configuration values in the Java Adapters configuration file.

```
gg.log=log4j
gg.log.level=INFO
```

These *gg.log* settings will result in a Log4j file to be created in the *GoldenGate\_Home/ dirrpt* directory that adheres to this naming convention, *{GROUPNAME}.log*. The supported Log4j log levels are in the following list in order of increasing logging granularity.

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- TRACE

Selection of a logging level will include all of the coarser logging levels as well (that is, selection of `WARN` means that log messages of `FATAL`, `ERROR` and `WARN` will be written to the log file). The Log4j logging can additionally be controlled by separate Log4j properties files. These separate Log4j properties files can be enabled by editing the `bootoptions` property in the Java Adapter Properties file. These three example Log4j properties files are included with the installation and are included in the classpath:

```
log4j-default.properties
log4j-debug.properties
log4j-trace.properties
```

You can modify the `bootoptions` in any of the files as follows:

```
javawriter.bootoptions=-Xmx512m -Xms64m -Djava.class.path=.:ggjava/ggjava.jar -
Dlog4j.configurationFile=samplelog4j.properties
```

You can use your own customized Log4j properties file to control logging. The customized Log4j properties file must be available in the Java classpath so that it can be located and loaded by the JVM. The contents of a sample custom Log4j properties file is the following:

```
# Root logger option
log4j.rootLogger=INFO, file

# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=sample.log
log4j.appender.file.MaxFileSize=1GB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L -
%m%n
```

There are two important requirements when you use a custom Log4j properties file. First, the path to the custom Log4j properties file must be included in the `javawriter.bootoptions` property. Logging initializes immediately when the JVM is initialized while the contents of the `gg.classpath` property is actually appended to the classloader after the logging is initialized. Second, the classpath to correctly load a properties file must be the directory containing the properties file without wildcards appended.

### 2.5.3 About Schema Evolution and Metadata Change Events

The Metadata in trail is a feature that allows seamless runtime handling of metadata change events by Oracle GoldenGate for Big Data, including schema evolution and schema propagation to Big Data target applications. The `NO_OBJECTDEFS` is a sub-parameter of the `Extract` and `Replicat` `EXTTRAIL` and `RMTTRAIL` parameters that lets you suppress the important metadata in trail feature and revert to using a static metadata definition.

The Oracle GoldenGate for Big Data Handlers and Formatters provide functionality to take action when a metadata change event is encountered. The ability to take action in the case of metadata change events depends on the metadata change events being available in the source trail file. Oracle GoldenGate supports metadata in trail and the propagation of DDL data from a source Oracle Database. If the source trail file does not have metadata in trail and DDL data (metadata change events) then it is not possible for Oracle GoldenGate for Big Data to provide and metadata change event handling.

## 2.5.4 About Configuration Property CDATA[] Wrapping

The GoldenGate for Big Data Handlers and Formatters support the configuration of many parameters in the Java properties file, the value of which may be interpreted as white space. The configuration handling of the Java Adapter trims white space from configuration values from the Java configuration file. This behavior of trimming whitespace may be desirable for some configuration values and undesirable for other configuration values. Alternatively, you can wrap white space values inside of special syntax to preserve the white space for selected configuration variables. GoldenGate for Big Data borrows the XML syntax of CDATA[] to preserve white space. Values that would be considered to be white space can be wrapped inside of CDATA[].

The following is an example attempting to set a new-line delimiter for the Delimited Text Formatter:

```
gg.handler.{name}.format.lineDelimiter=\n
```

This configuration will not be successful. The new-line character is interpreted as white space and will be trimmed from the configuration value. Therefore the `gg.handler` setting effectively results in the line delimiter being set to an empty string.

In order to preserve the configuration of the new-line character simply wrap the character in the CDATA[] wrapper as follows:

```
gg.handler.{name}.format.lineDelimiter=CDATA[\n]
```

Configuring the property with the CDATA[] wrapping preserves the white space and the line delimiter will then be a new-line character.

## 2.5.5 Using Regular Expression Search and Replace

You can perform more powerful search and replace operations of both schema data (catalog names, schema names, table names, and column names) and column value data, which are separately configured. Regular expressions (`regex`) are characters that customize a search string through pattern matching. You can match a string against a pattern or extract parts of the match. Oracle GoldenGate for Big Data uses the standard Oracle Java regular expressions package, `java.util.regex`, see "Regular Expressions" in [The Single UNIX Specification, Version 4](#).

### Topics:

- [Using Schema Data Replace](#)
- [Using Content Data Replace](#)



### 2.5.5.1 Using Schema Data Replace

You can replace schema data using the `gg.schemareplaceregex` and `gg.schemareplacestring` properties. Use `gg.schemareplaceregex` to set a regular expression, and then use it to search catalog names, schema names, table names, and column names for corresponding matches. Matches are then replaced with the content of the `gg.schemareplacestring` value. The default value of `gg.schemareplacestring` is an empty string or "".

For example, some system table names start with a dollar sign like `$mytable`. You may want to replicate these tables even though most Big Data targets do not allow dollar signs in table names. To remove the dollar sign, you could configure the following replace strings:

```
gg.schemareplaceregex=[$]
gg.schemareplacestring=
```

The resulting example of searched and replaced table name is `mytable`. These properties also support `CDATA[ ]` wrapping to preserve whitespace in the value of configuration values. So the equivalent of the preceding example using `CDATA[ ]` wrapping use is:

```
gg.schemareplaceregex=CDATA[[ $]]
gg.schemareplacestring=CDATA[ ]
```

The schema search and replace functionality supports using multiple search regular expressions and replacements strings using the following configuration syntax:

```
gg.schemareplaceregex=some_regex
gg.schemareplacestring=some_value
gg.schemareplaceregex1=some_regex
gg.schemareplacestring1=some_value
gg.schemareplaceregex2=some_regex
gg.schemareplacestring2=some_value
```

### 2.5.5.2 Using Content Data Replace

You can replace content data using the `gg.contentreplaceregex` and `gg.contentreplacestring` properties to search the column values using the configured regular expression and replace matches with the replacement string. For example, this is useful to replace line feed characters in column values. If the delimited text formatter is used then line feeds occurring in the data will be incorrectly interpreted as line delimiters by analytic tools.

You can configure *n* number of content replacement regex search values. The regex search and replacements are done in the order of configuration. Configured values must follow a given order as follows:

```
gg.contentreplaceregex=some_regex
gg.contentreplacestring=some_value
gg.contentreplaceregex1=some_regex
gg.contentreplacestring1=some_value
```

```
gg.contentreplaceregex2=some_regex  
gg.contentreplacestring2=some_value
```

Configuring a subscript of 3 without a subscript of 2 would cause the subscript 3 configuration to be ignored.

#### **NOT\_SUPPORTED:**

Regular express searches and replacements require computer processing and can reduce the performance of the Oracle GoldenGate for Big Data process.

To replace line feeds with a blank character you could use the following property configurations:

```
gg.contentreplaceregex=[\n]  
gg.contentreplacestring=CDATA[ ]
```

This changes the column value from:

```
this is  
me
```

to :

```
this is me
```

Both values support CDATA wrapping. The second value must be wrapped in a CDATA[ ] wrapper because a single blank space will be interpreted as whitespace and trimmed by the Oracle GoldenGate for Big Data configuration layer. In addition, you can configure multiple search a replace strings. For example, you may also want to trim leading and trailing white space out of column values in addition to trimming line feeds from:

```
^\s+|\s+$
```

```
gg.contentreplaceregex1=^\s+|\s+$  
gg.contentreplacestring1=CDATA[ ]
```

## 2.5.6 Scaling Oracle GoldenGate for Big Data Delivery

Oracle GoldenGate for Big Data supports breaking down the source trail files into either multiple Replicat processes or by using Coordinated Delivery to instantiate multiple Java Adapter instances inside a single Replicat process to improve throughput.. This allows you to scale Oracle GoldenGate for Big Data delivery.

There are some cases where the throughput to Oracle GoldenGate for Big Data integration targets is not sufficient to meet your service level agreements even after

you have tuned your Handler for maximum performance. When this occurs, you can configure parallel processing and delivery to your targets using one of the following methods:

- Multiple Replicat processes can be configured to read data from the same source trail files. Each of these Replicat processes are configured to process a subset of the data in the source trail files so that all of the processes collectively process the source trail files in their entirety. There is no coordination between the separate Replicat processes using this solution.
- Oracle GoldenGate Coordinated Delivery can be used to parallelize processing the data from the source trail files within a single Replicat process. This solution involves breaking the trail files down into logical subsets for which each configured subset is processed by a different delivery thread. For more information about Coordinated Delivery, see [https://blogs.oracle.com/dataintegration/entry/goldengate\\_12c\\_coordinated\\_replicat](https://blogs.oracle.com/dataintegration/entry/goldengate_12c_coordinated_replicat).

With either method, you can split the data into parallel processing for improved throughput. Oracle recommends breaking the data down in one of the following two ways:

- Splitting Source Data By Source Table –Data is divided into subsections by source table. For example, Replicat process 1 might handle source tables table1 and table2, while Replicat process 2 might handle data for source tables table3 and table2. Data is split for source table and the individual table data is not subdivided.
- Splitting Source Table Data into Sub Streams – Data from source tables is split. For example, Replicat process 1 might handle half of the range of data from source table1, while Replicat process 2 might handler the other half of the data from source table1.

Additional limitations:

- Parallel apply is *not* supported.
- The BATCHSQL parameter not supported.

### Example 2-3 Scaling Support for the Oracle GoldenGate for Big Data Handlers

Handler Name	Splitting Source Data By Source Table	Splitting Source Table Data into Sub Streams
Cassandra	Supported	Supported when: <ul style="list-style-type: none"> <li>• Required target tables in Cassandra are pre-created.</li> <li>• Metadata change events do not occur.</li> </ul>
Elastic Search	Supported	Supported

Handler Name	Splitting Source Data By Source Table	Splitting Source Table Data into Sub Streams
HBase	Supported when all required HBase namespaces are pre-created in HBase.	Supported when: <ul style="list-style-type: none"> <li>All required HBase namespaces are pre-created in HBase.</li> <li>All required HBase target tables are pre-created in HBase. Schema evolution is not an issue because HBase tables have no schema definitions so a source metadata change does not require any schema change in HBase.</li> <li>The source data does not contain any truncate operations.</li> </ul>
HDFS	Supported	Supported with some restrictions. <ul style="list-style-type: none"> <li>You must select a naming convention for generated HDFS files where the file names do not collide. Colliding HDFS file names results in a Replicat abend. When using coordinated apply it is suggested that you configure <code>groupname</code> as part of the configuration for the <code>gg.handler.name.fileNameMappingTemplate</code> property. The <code>groupname</code> template resolves to the Replicat name concatenated with the Replicat thread number, which provides unique naming per Replicat thread.</li> <li>Schema propagation to HDFS and Hive integration is <i>not</i> currently supported.</li> </ul>
JDBC	Supported	Supported
Kafka	Supported	Supported for formats that support schema propagation, such as Avro. This is less desirable due to multiple instances feeding the same schema information to the target.
Kafka Connect	Supported	Supported
Kinesis Streams	Supported	Supported
MongoDB	Supported	Supported

Handler Name	Splitting Source Data By Source Table	Splitting Source Table Data into Sub Streams
Java File Writer	Supported	Supported with the following restrictions: You must select a naming convention for generated files where the file names do not collide. Colliding file names may results in a Replicat abend and/or polluted data. When using coordinated apply it is suggested that you configure <code>\${groupName}</code> as part of the configuration for the <code>gg.handler.name.fileNameMappingTemplate</code> property. The <code>\${groupName}</code> template resolves to the Replicat name concatenated with the Replicat thread number, which provides unique naming per Replicat thread.

## 2.5.7 Using Identities in Oracle GoldenGate Credential Store

The Oracle GoldenGate credential store manages user IDs and their encrypted passwords (together known as credentials) that are used by Oracle GoldenGate processes to interact with the local database. The credential store eliminates the need to specify user names and clear-text passwords in the Oracle GoldenGate parameter files.

An optional alias can be used in the parameter file instead of the user ID to map to a userid and password pair in the credential store. The credential store is implemented as an auto login wallet within the Oracle Credential Store Framework (CSF). The use of an LDAP directory is not supported for the Oracle GoldenGate credential store. The auto login wallet supports automated restarts of Oracle GoldenGate processes without requiring human intervention to supply the necessary passwords.

In Oracle GoldenGate for Big Data, you specify the alias and domain in the property file not the actual user ID or password. User credentials are maintained in secure wallet storage.

- [Creating a Credential Store](#)
- [Adding Users to a Credential Store](#)
- [Configuring Properties to Access the Credential Store](#)

### 2.5.7.1 Creating a Credential Store

You can create a credential store for your Big Data environment.

Run the `GGSCI ADD CREDENTIALSTORE` command to create a file called `cwallet.sso` in the `dircrd/` subdirectory of your Oracle GoldenGate installation directory (the default).

You can the location of the credential store (`cwallet.sso` file by specifying the desired location with the `CREDENTIALSTORELOCATION` parameter in the `GLOBALS` file.

For more information about credential store commands, see *Reference for Oracle GoldenGate*.

**Note:**

Only one credential store can be used for each Oracle GoldenGate instance.

## 2.5.7.2 Adding Users to a Credential Store

After you create a credential store for your Big Data environment, you can add users to the store.

Run the GGSCI `ALTER CREDENTIALSTORE ADD USER userid PASSWORD password [ALIAS alias] [DOMAIN domain]` command to create each user, where:

- *userid* is the user name. Only one instance of a user name can exist in the credential store unless the `ALIAS` or `DOMAIN` option is used.
- *password* is the user's password. The password is echoed (not obfuscated) when this option is used. If this option is omitted, the command prompts for the password, which is obfuscated as it is typed (recommended because it is more secure).
- *alias* is an alias for the user name. The alias substitutes for the credential in parameters and commands where a login credential is required. If the `ALIAS` option is omitted, the alias defaults to the user name.

For example:

```
ALTER CREDENTIALSTORE ADD USER scott PASSWORD tiger ALIAS scsm2 domain
ggadapters
```

For more information about credential store commands, see *Reference for Oracle GoldenGate*.

## 2.5.7.3 Configuring Properties to Access the Credential Store

The Oracle GoldenGate Java Adapter properties file requires specific syntax to resolve user name and password entries in the Credential Store at runtime. For resolving a user name the syntax is the following:

```
ORACLEWALLETUSERNAME[alias domain_name]
```

For resolving a password the syntax required is the following:

```
ORACLEWALLETPASSWORD[alias domain_name]
```

The following example illustrates how to configure a Credential Store entry with an alias of `myalias` and a domain of `mydomain`.



**Note:**

With HDFS Hive JDBC the user name and password is encrypted.

Oracle Wallet integration only works for configuration properties which contain the string username or password. For example:

```
gg.handler.hdfs.hiveJdbcUsername=ORACLEWALLETUSERNAME[myalias mydomain]
gg.handler.hdfs.hiveJdbcPassword=ORACLEWALLETPASSWORD[myalias mydomain]
```

ORACLEWALLETUSERNAME and ORACLEWALLETPASSWORD can be used in the Extract (similar to Replicat) in JMS handler as well. For example:

```
gg.handler.<name>.user=ORACLEWALLETUSERNAME[JMS_USR JMS_PWD]
gg.handler.<name>.password=ORACLEWALLETPASSWORD[JMS_USR JMS_PWD]
```

Consider the user name and password entries as accessible values in the Credential Store. Any configuration property resolved in the Java Adapter layer (not accessed in the C user exit layer) can be resolved from the Credential Store. This allows you more flexibility to be creative in how you protect sensitive configuration entries.

# 3

## Installing Oracle GoldenGate Microservices for Big Data

The Oracle GoldenGate Microservices Architecture (MA) for Big Data is installed using OUI. You can also use a command line silent installation using OUI.

This chapter describes how to install a new instance of Oracle GoldenGate Microservices for Big Data. The installation is a three-step process:

- Install the Oracle GoldenGate MA.
- Set the necessary environment variables.
- Deploy an Oracle GoldenGate instance using the configuration assistant.

The installer registers the Oracle GoldenGate home directory with the central inventory that is associated with the selected database. The inventory stores information about all Oracle software products installed on a host if the product was installed using OUI.

Disk space is also required for the Oracle GoldenGate Bounded Recovery feature. Bounded Recovery is a component of the general Extract checkpointing facility. It caches long-running open transactions to disk at specific intervals to enable fast recovery upon a restart of Extract. At each bounded recovery interval (controlled by the `BRINTERVAL` option of the `BR` parameter) the disk required is as follows: for each transaction with cached data, the disk space required is usually 64k plus the size of the cached data rounded up to 64k. Not every long-running transaction is persisted to disk.

### Topics:

- [Installing Oracle GoldenGate MA for Big Data Using the UI](#)  
Interactive installation provides a graphical user interface that prompts for the required installation information. These instructions apply to new installations and upgrades.
- [Silent Installation](#)  
Silent installation from the command line interface can be performed if your system does not have an X-Windows or graphical interface or you want to perform the installation in an automated way. Silent installations ensure that multiple users in your organization use the same installation options when installing Oracle products.
- [Setting Up Secure or Non-Secure Deployments](#)  
You can choose to set up a secure or non-secure deployment.

### 3.1 Installing Oracle GoldenGate MA for Big Data Using the UI

Interactive installation provides a graphical user interface that prompts for the required installation information. These instructions apply to new installations and upgrades.

To install Oracle GoldenGate for Big Data using the UI:

1. Create a temporary staging directory into which you will install Oracle GoldenGate. For example, `mkdir /u01/stage/oggsc`.



2. Extract the installation ZIP file into the temporary staging directory. For example:  
`unzip .ggs_Linux_x64_BigData_64bit_services.zip -d ./temp directory`
  3. From the expanded directory, run the `ggs_Linux_x64_BigData_64bit_services/Disk1/runInstaller` program on UNIX or Linux to display the **Installation Wizard**.
  4. On the **Select Installation Option** page, select the Oracle Database version for your environment, then click **Next**.
  5. If you are on Windows and running Manager as a service, set the system variable `PATH` to include `jvm.dll`, then delete the Manager service and re-add it.
  6. On the **Specify Installation Details** page, ensure that the following environment variable is set:
    - `OGG_HOME`
  7. Click **Next** to display the **Summary** page.
  8. Confirm that there is enough space for the installation and that the installation selections are correct.
    - (Optional) Click **Save Response File** to save the installation information to a response file. You can run the installer from the command line with this file as input to duplicate the results of a successful installation on other systems. You can edit this file or create a new one from a template.
    - Click **Install** to begin the installation or **Back** to go back and change any input specifications. When upgrading an existing Oracle GoldenGate installation, OUI notifies you that the software location has files or directories. Click **Yes** to continue.
    - If you created a central inventory directory, then you are prompted to run the `INVENTORY_LOCATION/orainstRoot.sh` script. This script must be executed as the root operating system user. This script establishes the inventory data and creates subdirectories for each installed Oracle product (in this case, Oracle GoldenGate).
- You are notified when the installation is finished.
9. Click **Close** to complete the installation.

## 3.2 Silent Installation

Silent installation from the command line interface can be performed if your system does not have an X-Windows or graphical interface or you want to perform the installation in an automated way. Silent installations ensure that multiple users in your organization use the same installation options when installing Oracle products.

Silent installations are driven by using a response file. Response files can be saved by selecting the Save Response File option during an interactive Oracle Universal Installer session or by editing the `oggcore.rsp` template located in the response directory after unzipping the binaries.

The Oracle GoldenGate response file contains a standard set of Oracle configuration parameters in addition to parameters that are specific to Oracle GoldenGate. These

parameters correspond to the fields in the interactive session. The response file location is

```
unzipped_directory/ggs_Linux_x64_BigData_64bit_services/Disk1/response
```

To perform the installation using a response file, issue the following command:

```
unzipped_directory/ggs_Linux_x64_BigData_64bit_services.zip/Disk1/runInstaller -silent  
-nowait -responseFile absolute_path_to_response_file
```

## 3.3 Setting Up Secure or Non-Secure Deployments

You can choose to set up a secure or non-secure deployment.

A secure deployment involves making RESTful API calls and conveying trail data between the Distribution Server and Receiver Server, over SSL/TLS. You can use your own existing business certificate from your Certificate Authority (CA) or you might create your own certificates. When first creating the SSL/TLS security certificates, you must ensure that the SSL/TLS security environment variables.

For a non-secure deployment, the RESTful API calls occur over plain-text HTTP and conveyance between Distribution Server and Receiver Server is performed using the wss, ogg, and ws protocols.

This section describes the steps to configure a non-secure deployment and prerequisites and tasks to configure a secure deployment.

- [How to Add Secure or Non-Secure Deployments](#)  
Adding deployments is the first task in the process of setting up a data replication platform. Deployments are managed from the Service Manager.
- [How to Remove a Deployment](#)  
You can remove a deployment using OGGCA or in silent mode.

### 3.3.1 How to Add Secure or Non-Secure Deployments

Adding deployments is the first task in the process of setting up a data replication platform. Deployments are managed from the Service Manager.

After completing the Oracle GoldenGate Microservices installation, you can add initial and subsequent deployments using the Configuration Assistant (OGGCA) wizard.



#### Note:

Oracle recommends that you have a single Service Manager per host, to avoid redundant upgrade and maintenance tasks with Oracle GoldenGate releases.

Use OGGCA to add multiple deployments to a Service Manager. This allows you to upgrade the same Service Manager with new releases or patches. The source and target deployments serve as endpoints for setting up the distribution path for data replication.

1. From the OGG\_HOME directory, run the `$OGG_HOME/bin/oggca.sh` program on UNIX or Linux.

The Oracle GoldenGate Configuration Assistant (oggca) is started. Run this program, each time you want to add a deployment.

2. In the **Select Service Manager Options** step:
  - a. Select whether you want to use an existing Service Manager or create a new one. In most configurations, you only have one Service Manager that is responsible for multiple deployments.
  - b. For a new Service Manager, enter or browse to the directory that you want to use for your deployment. Oracle recommends that you create a `ServiceManager` directory within the deployment sub-directory structure to store the Service Manager files.
  - c. Enter the hostname or IP Address of the server.
  - d. Enter a unique port number that the Service Manager will listen on, or choose the port already in use if selecting an existing Service Manager.
  - e. (Optional) You can register the Service Manager to run as a service so as to avoid manually starting and stopping it.

You can choose to run *one* Service Manager as a service (daemon). If there is an existing Service Manager registered as a service and you select a new Service Manager to register as a service, an alert is displayed indicating that you cannot register the new one as a service. All other Service Managers are started and stopped using scripts installed in the `bin` directory of the deployment. You cannot register an existing Service Manager as a service.

3. In the **Configuration Options** step, you can add or remove deployments. You can only add or remove one deployment for one Service Manager at a time.

 **Note:**

Ensure that your Service Manager is up and running prior to launching OGGCA.

4. In the **Deployment Details** step:
  - a. Enter the deployment name using these conventions:
    - Must begin with a letter.
    - Can be a standard ASCII alphanumeric string not exceeding 32 characters.
    - Cannot include extended ASCII characters.
    - Special characters that are allowed include underscore ('\_'), hyphen ('/'), dash ('-'), period ('.'). The name before the / symbol should be "slash" or "forward slash".
    - Cannot be "ServiceManager".
  - b. Enter or select the Oracle GoldenGate installation directory. If you have set the `$OGG_HOME` environment variable, the directory is automatically populated. Otherwise, the parent directory of the `oggca.sh` (Linux) or `oggca.bat` (Windows) script is used.
  - c. Click **Next**.
5. On the **Select Deployment Directories** page:

- a. Enter or select a deployment directory where you want to store the deployment registry and configuration files. When you enter the deployment directory name, it is created if it doesn't exist. Oracle recommends that you do *not* locate your deployment directory inside your `$OGG_HOME` and that you create a separate directory for easier upgrades. The additional fields are automatically populated based on the specified deployment directory.

 **Note:**

The deployment directory name (user deployment directory) needs to be different than the directory name chosen in the first screen (Service Manager deployment directory).

- b. You can customize the deployment directories so that they are named and located differently from the default.
  - c. Enter or select different directories for the various deployment elements.
  - d. Click **Next**.
6. On the **Environment Variables** page:

Enter the requested values for the environment variables. Double-click in the field to edit it. You can copy and paste values in the environment variable fields. Make sure that you tab or click outside of the field after entering each value, otherwise it's not saved. If you have set any of these environment variables, the directory is automatically populated.

#### **OGG\_HOME**

The directory where you installed Oracle GoldenGate. This variable is fixed and cannot be changed.

 **Note:**

On a Windows platform, ensure that there's no space in the `OGG_HOME` directory path otherwise `OGGCA` will not run.

#### **LD\_LIBRARY\_PATH**

This variable is used to specify the path to search for libraries on UNIX and Linux. It may have a different name on some operating systems, such as `LIBPATH` on IBM AIX on POWER Systems (64-Bit), and `SHLIB_PATH` on HP-UX. This path points to the Oracle GoldenGate installation directory and the underlying instant client directory by default. It might be extended if `USER_EXITS` are in use.

You can add additional environment variables to customize your deployment or remove variables. For instance, you can enter the following variable to default to another international charset: `ENV_LC_ALL=zh_CN.UTF-8`

Click **Next**.

7. On the **Administrator Account** page:
  - a. Enter a user name and password that you want to use to sign in to the Oracle GoldenGate Microservices Service Manager and the other servers. This user is the security user for this deployment. Select the **Enable strong password policy in the**

**new deployment** checkbox to ensure setting a highly secure password for your user account. The strong password policy has the following requirements:

- At least one lowercase character [a...z]
- At least one uppercase character [A...Z]
- At least one digit [0...9]
- At least one special character [- ! @ % & \* . #]
- The length should be between 8 and 30 characters.

If you are using an existing Service Manager, you must enter the same log in credentials that were used when adding the first deployment.

- b. Select the check box that allows you to enable a strong password policy for your new deployment. If you select this option, then the password must adhere to restrictions, otherwise an error occurs, which requires you to specify a stronger password.
  - c. Click **Next**.
8. On the **Security Options** page:

- a. You can choose whether or not you want to secure your deployment. Oracle recommends that you enable SSL/TLS security. If you do not want to use security for your deployment, deselect the check box.

This operation exposes the option **This non-secure deployment will be used to send trail data to a secure deployment**. Select this check box if the non-secure target deployment is meant to communicate with a secure source deployment.

However, you must enable security if configuring for Oracle GoldenGate sharding support.

- b. Also see: [About Target-Initiated Paths](#) in *Step by Step Data Replication Using Oracle GoldenGate Microservices Architecture Guide*.
- c. (Optional) You can specify a client wallet location so that you can send trail data to a secure deployment. This option is useful when Distribution Server from the source deployment is unsecured whereas the Receiver Server on the target deployment is secured. So, the sender may be configured for public access while the Receiver Server requires authentication and authorization, which is established using PKI before the incoming data is applied. For more information, see [Creating a Self Signed Certificate](#) and [Creating a Client Certificate](#) in *Oracle GoldenGate Security Guide*.
- d. For your Server, select one of the options, and then provide the required file locations. When using an existing wallet, it must have the appropriate certificates already imported into it. If you choose to use a certificate, enter the corresponding pass phrase.  
When using a self-signed certificate, a new Oracle Wallet is created in the new deployment and these certificates are imported into it. For certificates, enter the location of the private key file and the pass phrase. The private key files must be in the PKCS#8 format.
- e. For your Client, select one of the options, and then provide the required information as you did for your server.
- f. Click **Next**.

9. (If Security is enabled) On the **Advanced Security Settings** page, the TLS 1.1 and TLS 1.2 options are available. TLS 1.2 is selected by default.

When you open the Advanced Security Settings for the first time with TLS 1.2, the following cipher suites are listed:

```
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
```

- a. Use the arrows to add or remove cipher suites.
- b. Use **Up** and **Down** to reorder how the cipher suites are applied
- c. Click **Next**.

 **Note:**

For more information on TCP/IP encryption options with RMTHOST, see RMTHOST in *Reference for Oracle GoldenGate*.

10. (If Sharding is enabled) On the **Sharding Options** page:
  - a. Locate and import your Oracle GoldenGate Sharding Certificate. Enter the distinguished name from the certificate that will be used by the database sharding code to identify itself when making REST API calls to the Oracle GoldenGate MA services.
  - b. Enter a unique name for the certificate.
  - c. Click **Next**.
11. On the **Port Settings** page:

- a. Enter the Administration Server port number, and then when you leave the field the other port numbers are populated in ascending numbers. Optionally, you can enter unique ports for each of the servers.
- b. Select **Enable Monitoring** to use the Performance Metrics Server.
- c. Click inside the Performance Metrics Server port fields to populate or enter the ports you want to use. Ensure that you choose available ports for TCP.

Select the UDP port for performance monitoring. The option to select the UDP port is displayed only with deployments on Windows and other operating systems that don't support UDS communication with Performance Metric Server. See Supported Operating Systems for UDS.

You can change the TCP port from the Service Manager console after the deployment is done. For more information on PMSRVR, see ENABLEMONITORING.

- d. Select the type of datastore that you want the Performance Metrics Server to use, the default Berkeley Database (BDB) data store or Open LDAP Lightning Memory-Mapped Database (LMDB). You can also designate the Performance Monitor as a Critical Service if integrating the Service Manager with XAG.  
For BDB information, see [Oracle Berkeley DB 12c Release 1](#). For LMDB information, see <http://www.lmdb.tech/doc/>.
- e. Select the location of your datastore. BDB and LMDB are in-memory and disk-resident databases. The Performance Metrics server uses the datastore to store all performance metrics information.
- f. Click **Next**.

 **Note:**

The `oggca` utility validates whether or not the port you entered is currently in use or not.

12. On the **Summary** page:

- a. Review the detailed configuration settings of the deployment before you continue.
- b. (Optional) You can save the configuration information to a response file. Oracle recommends that you save the response file. You can run the installer from the command line using this file as an input to duplicate the results of a successful configuration on other systems. You can edit this file or a new one from the provided template.

 **Note:**

When saving to a response file, the administrator password is not saved for security reasons. You must edit the response file and enter the password if you want to reuse the response file for use on other systems.

- c. Click **Finish** to the deployment.
- d. Click **Next**.

**13.** On the **Configure Deployment** page:

Displays the progress of the deployment creation and configuration.

- a. If the Service Manager is being registered as a service, a pop-up appears that directs you how to run the script to register the service. The Configuration Assistant verifies that these scripts have been run. If you did not run them, you are queried if you want to continue. When you click **Yes**, the configuration completes successfully. When you click **No**, a temporary failed status is set and you click **Retry** to run the scripts.

Click **Ok** after you run the script to continue.

- b. Click **Next**.

**14.** On the **Finish** page:

Click **Close** to close the Configuration Assistant.

## 3.3.2 How to Remove a Deployment

You can remove a deployment using OGGCA or in silent mode.

**Topics:**

- [How to Remove a Deployment: GUI](#)  
You can remove a deployment using the Oracle GoldenGate Configuration Assistant wizard.
- [How to Remove a Deployment: Silent Mode](#)  
You can remove a deployment silently using the Oracle GoldenGate Configuration Assistant (oggca) from the Oracle GoldenGate Home bin directory.

### 3.3.2.1 How to Remove a Deployment: GUI

You can remove a deployment using the Oracle GoldenGate Configuration Assistant wizard.

**To remove a deployment:** **Note:**

When you remove a deployment or uninstall Oracle GoldenGate Microservices, the system does not automatically stop processes. As a result, you may have to stop processes associated with the deployment and you must clean files manually.

1. Run the Oracle GoldenGate Configuration Assistant wizard:  
`$OGG_HOME/bin`
2. Select **Existing Service Manager** from the **Select Service Manager Options** screen. Click **Next**
3. Select **Remove Existing Oracle GoldenGate Deployment** from the Configuration Options screen.
4. Select the deployment you need to remove from the **Deployment Name** list box. Also select the **Delete Deployment Files from Disk** check box if you want to remove all the deployment files (including configuration files) from the host.
5. Enter the Administration account user name and password and click **Next**.



6. See the list of settings that are deleted with the deployment and click **Finish**.

**To remove a Service Manager:**

1. Run Oracle GoldenGate Configuration Assistant wizard:  
`$OGG_HOME/bin`
2. Select **Existing Service Manager** from the **Select Service Manager Options** screen. Click **Next**.
3. If there are no other deployments to remove, then the option to remove the Service Manager is available in the drop down. Select **Remove Service Manager Deployment** from the Configuration Options screen.
4. Click **Finish**.


**Files to be Removed Manually After Removing Deployment**

It's mandatory to delete some files manually only in case there's a Service Manager registered but you have to unregister it and register a new one. To remove files manually, you must have `root` or `sudo` privileges. The files to be deleted include:

Operating System	Files to be Removed Manually to Unregister an Existing Service Manager
Linux 6	<ul style="list-style-type: none"> <li>• /etc/init.d/OracleGoldenGate</li> <li>• /etc/rc.d/*OracleGoldenGate</li> <li>• /etc/rc*.d/*OracleGoldenGate</li> <li>• /etc/oggInst.loc</li> </ul>
Linux 7	/etc/systemd/system/OracleGoldenGate.service

The following commands are executed to stop the Service Manager:

```
systemctl stop OracleGoldenGate
systemctl disable OracleGoldenGate *
```

 **Note:**

If the Service Manager is not registered as a service (with or without the integration with XAG), OGGCA stops the Service Manager deployment, otherwise, a script called `unregisterServiceManager` is created, and when executed by the user, it runs the `systemctl` commands and deletes the mentioned files.

### 3.3.2.2 How to Remove a Deployment: Silent Mode

You can remove a deployment silently using the Oracle GoldenGate Configuration Assistant (`oggca`) from the Oracle GoldenGate Home bin directory.

By removing a deployment, you can delete various components of the deployment, including, Extracts, Replicats, paths, and configuration files. However, the Service Manager is not deleted.

**To remove a deployment silently:****Note:**

If the Service Manager is registered as a system service, removing a deployment silently will not unregister the service.

1. Ensure that you have a deployment response file. To get the deployment response file, run the OGGCA and save the response file.
2. Update the following lines within the deployment response file:

```
CONFIGURATION_OPTION=REMOVE  
ADMINISTRATOR_PASSWORD=*****  
CREATE_NEW_SERVICEMANAGER=false  
DEPLOYMENT_NAME=deployment_name  
REMOVE_DEPLOYMENT_FROM_DISK=true
```

In case of multiple deployments, you must specify the deployment name using the `DEPLOYMENT_NAME` field. You can use the `REMOVE_DEPLOYMENT_FROM_DISK` option to remove physical files and folders associated with deployment.

3. Run the OGGCA program from the following location using the `-silent` and `-responseFile` options. Providing the exact path to the deployment response is needed.

```
$OGG_HOME/bin/oggca.sh -silent -responseFile  
path_to_response_file/response_file.rsp
```

**Example:**

```
$OGG_HOME/bin/oggca.sh -silent -responseFile  
/home/oracle/software/ogg_deployment.rsp
```

# 4

## Upgrading Oracle GoldenGate Classic for Big Data

This chapter describes how to upgrade to Oracle GoldenGate for Big Data 21c (21.1.0.0.0) by downloading the product as described in [Downloading Oracle GoldenGate for Big Data](#), and then choosing the upgrade paths that suits your environment.

After upgrading, you must then convert your Extract processes to Replicat processes.

### Note:

There is no supported upgrade path from release 12.1.2 to 12.2.0.x.

There is no supported upgrade path from release 12.2.0.1.0 to 12.2.0.1.1.

### Topics:

- [Upgrading by Overwriting the Existing Installation](#)
- [Upgrading by Installing into a New Directory](#)
- [Switching Existing Extract Processes to Replicat Processes](#)

### 4.1 Upgrading by Overwriting the Existing Installation

The most straightforward upgrade path is to copy the Oracle GoldenGate for Big Data 21c (21.1.0.0.0) files into the existing 12c (19.1.0.0.x) installation directory. Overwriting the product files is possible because there is neither structural nor package name changes in the current release. In most cases the 19.1.0.0.x handler and formatter configurations are also compatible with the 21.1.0.0.0 release. However, in some cases configuration updates may be required.

1. (Source systems) Back up the current Oracle GoldenGate for Big Data installation directory on the source system, and any working directories that you have installed on a shared drive in a cluster (if applicable).
2. (Source system) Stop user activity on objects in the Oracle GoldenGate configuration.
3. Create a new installation directory for the 21.1.0.0.0 installation with no spaces in its name.
4. Extract the ZIP file into this new installation directory, which divides the files into several subdirectories.
5. Copy the Oracle GoldenGate for Big Data 21c (21.1.0.0.x) files into the existing 19c (19.1.0.0.x) installation directory.

6. Start the Replicat processes and verify that they are running.

```
GGSCI> START MANAGER
GGSCI> START REPLICAT group_name
GGSCI> INFO REPLICAT group_name
GGSCI> VIEW REPORT group_name
```

Certain 19.1.0.0.x JAR files are still present after the overwriting process though the new 21.1.0.0.0 JAR files are used.

## 4.2 Upgrading by Installing into a New Directory

Use the following steps to install the Oracle GoldenGate for Big Data 19c (19.1.0.0.x) files into a new installation directory.

1. (Source systems) Back up the current Oracle GoldenGate for Big Data installation directory on the source systems, and any working directories that you have installed on a shared drive in a cluster (if applicable).
2. (Source system) Stop user activity on objects in the Oracle GoldenGate configuration.
3. Create a new installation directory for the 19.1.0.0 installation with no spaces in its name.
4. Extract the ZIP file into this new installation directory, which divides the files into several subdirectories.
5. Start GGSCI to create the remaining subdirectories in the installation location.

```
$ ggsci
GGSCI> CREATE SUBDIRS
```

6. Copy all of the `dirprm` files from your existing installation into the `dirprm` directory in the new installation location.

### Note:

All of your configuration files must be in the `dirprm` directory. If you have property files, Velocity templates, or other configuration files in a location other than `dirprm` in your old installation, then you must copy them to the `dirprm` directory in the new installation.

7. Copy all of the `dirdef` files from your existing installation into the `dirdef` directory in the new installation location.
8. If you have data files stored in the 12.3.1.x or 12.2.0.x installation `dirdat` directory, then copy or move the existing trail files to the `dirdat` directory of the new installation.
9. If you have additional JAR files or other custom files in your 12.3.1.x or 12.2.0.x installation, then copy them to the new installation directory.

10. Configure the Replicat processes in the new installation directory by starting GGSCI and adding the Replicat and naming the trails.

```
GGSCI> ADD REPLICAT group_name, EXTTRAIL trail_name, ...
GGSCI> ALTER group_name EXTSEQNO seqno EXTRBA rba
```

Optionally, you could alter the starting position of Replicat processing as needed.

11. Start the Replicat processes and verify that they are running.

```
GGSCI> START MANAGER
GGSCI> START REPLICAT group_name
GGSCI> INFO REPLICAT group_name
GGSCI> VIEW REPORT group_name
```

12. Modify the source system to write to the new Oracle GoldenGate for Big Data installation directory:
  - a. (Optional) Upgrade the source database Oracle GoldenGate capture following the upgrade procedure for your database platform.
  - b. Configure the source database capture to write to the new Oracle GoldenGate for Big Data 12.3.1.x installation `dirdat` directory.
  - c. When the old Oracle GoldenGate for Big Data installation has processed all its data, switch over to the process that will send data to the new location.

## 4.3 Switching Existing Extract Processes to Replicat Processes

In previous releases, you could use an Extract and pump process to write to your Big Data targets. In this release, this solution is deprecated so you must use Replicat.

A typical Extract configuration is similar to:

```
EXTRACT mygroup
SOURCEDEFS path/to/source/def/file
CUSEREXIT libggjava_ue.so CUSEREXIT PASSTHRU INCLUDEUPDATEBEFORES
GETUPDATEBEFORES
TABLE *.*;
```

With Replicat the preceding configuration would be:

```
REPLICAT mygroup
SOURCEDEFS path/to/source/def/file
TARGETDB LIBFILE libggjava.so SET property=/path/to/properties/file
MAP *.* TARGET *.*;
```

The same properties file works in the Replicat configuration so you do not need to change the properties file.

To complete this process, you must add the Replicat group, start the process, and verify that it is running:

```
GGSCI> ADD REPLICAT group_name, EXTTRAIL trail_name, ...
GGSCI> ALTER group_name EXTSEQNO seqno EXTRBA rbaGGSCI> START REPLICAT
```

```
group_name  
GGSCI> INFO REPLICAT group_name  
GGSCI> VIEW REPORT group_name
```

# 5

## Upgrading Oracle GoldenGate Microservices for Big Data

For Microservices, the earliest version that can be upgraded from is Oracle GoldenGate 21c (21.1.0.0). As a best practice, perform a minimal upgrade first, so that you can troubleshoot more easily in case of any issue. After the environment is upgraded successfully, you can implement the new functionality.

This chapter describes how to upgrade Oracle GoldenGate Microservices Architecture for Big Data from the previous releases of Oracle GoldenGate Microservices Architecture for BigData to the current release.

The pre-upgrade requirements are as follows:

- Stop all Oracle GoldenGate processes.
- Start Oracle GoldenGate.

### Topics:

- [Obtaining the Oracle GoldenGate Distribution](#)
- [Scope of Upgrade](#)  
Even though you may only upgrade the source or target, rather than both, all processes are involved in upgrade. All processes must be stopped in the correct order for the upgrade, regardless of which component you upgrade, and the trails must be processed until empty.
- [Upgrading Oracle GoldenGate Microservices – GUI Based](#)
- [Upgrading Oracle GoldenGate Microservices Using REST APIs](#)

## 5.1 Obtaining the Oracle GoldenGate Distribution

To obtain Oracle GoldenGate:

1. Go to [edelivery.oracle.com](https://edelivery.oracle.com). For more information, see [My Oracle Support Banner Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches \(Doc ID 1557031.1\)](#). To access Oracle Technology Network (OTN), go to <https://www.oracle.com/integration/goldengate/>
2. Find the Oracle GoldenGate 21c (21.1.0) release and download the ZIP file onto your system.

For more information about locating and downloading Oracle Fusion Middleware products, see the [Oracle® Fusion Middleware Download, Installation, and Configuration Readme Files](#) on OTN.

## 5.2 Scope of Upgrade

Even though you may only upgrade the source or target, rather than both, all processes are involved in upgrade. All processes must be stopped in the correct order for the upgrade, regardless of which component you upgrade, and the trails must be processed until empty.

Before you start the upgrade, review the information about upgrading Extract and Replicat.

Oracle recommends that you begin your upgrade with the target rather than the source to avoid the necessity of adjusting the trail file format.

- **Replicat Upgrade Considerations**  
All Replicat installations should be upgraded at the same time. It is critical to ensure that all trails leading to all Replicat groups on all target systems are processed until empty, according to the upgrade instructions.

### 5.2.1 Replicat Upgrade Considerations

All Replicat installations should be upgraded at the same time. It is critical to ensure that all trails leading to all Replicat groups on all target systems are processed until empty, according to the upgrade instructions.

Before you start the upgrade, review the information about upgrading Extract and Replicat.

Oracle recommends that you begin your upgrade with the target rather than the source to avoid the necessity of adjusting the trail file format.

## 5.3 Upgrading Oracle GoldenGate Microservices – GUI Based

To obtain the Oracle GoldenGate installation software and set up the directories for upgrade:

1. Download the latest Oracle GoldenGate Microservices 21c software from the Oracle Technology Network or eDelivery.
2. Upload the Oracle GoldenGate Microservices 21c software to a staging location on the server where a previous release of Oracle GoldenGate Microservices exists.
3. Unzip Oracle GoldenGate Microservices 21c software in the staging location.

```
$ cd /tmp $ unzip  
./fbo_ggs_Linux_x64_services_shiphome.zip
```

4. Untar the tar file that gets created after the unzip command: `tar -xvf ggs_Linux_x64_Oracle_64bit.tar`

5. Move into the unzipped files and execute the `runInstaller` command.

```
$ cd ./fbo_ggs_Linux_x64_services_shiphome/Disk1  
$ ./runInstaller
```



6. For Software Location, specify where the new Oracle GoldenGate home is located. This is not the same location as the current Oracle GoldenGate home. Click **Next**.
7. Click **Install** to begin installing the new Oracle GoldenGate MA. When the installation is done, click **Close**.

At this point, you should have two Oracle GoldenGate MA home directories: one for your old home (21c) and a new home (21.x.x).

8. Verify the current version of Oracle GoldenGate Home through Service Manager.
  - a. Login to the Service Manager:  

```
http://host:servicemanager_port
```
  - b. Review the deployment section for your current Oracle GoldenGate home location.
9. Update the Service Manager and the deployments with the location of the new Oracle GoldenGate home.
  - a. Click **Service Manager**, then **Deployment name**.
  - b. Next to the deployment details, click the pencil icon to display the dialog box to edit the Oracle GoldenGate home.
  - c. Update the Oracle GoldenGate home with the complete path to the new Oracle GoldenGate home. Also update the following, if required:

```
LD_LIBRARY_PATH
```

- d. Click **Apply**.
- e. Confirm that the Oracle GoldenGate home has been updated.
- f. Stop all Extracts, Replicats, and Distribution paths.
- g. Use the action button to restart Service Manager or Deployment.

 **Note:**

You can confirm that the Oracle GoldenGate home was updated by looking at the process from the operating system for Service Manager. The Service Manager process should be running from the new Oracle GoldenGate home.

10. To upgrade the associated deployments, follow the same steps for Service Manager after ensuring that all the Extract and Replicat processes in that deployment have been stopped.

## 5.4 Upgrading Oracle GoldenGate Microservices Using REST APIs

After you have completed the upgrade tasks detailed in [Upgrading Oracle GoldenGate Microservices – GUI Based](#), you are ready to update the Oracle GoldenGate MA home (OGG\_HOME) for the Service Manager or deployments using REST API.

To upgrade the Service Manager, the following cURL command can be used to update the Oracle GoldenGate home:

```
curl -X PATCH \  
  https://<hostname>:<port>/services/v2/deployments/ServiceManager \  
  -H 'cache-control: no-cache' \  
  -d '{"oggHome":"/opt/oracle/product/21.1.0/oggcore_1", "status":"restart"}'
```

To upgrade a deployment:

1. Stop all Extract and Replicats within the Administration Service.
2. Stop all Distribution Paths within the Distribution Service.
3. Run this cURL command to update the Oracle GoldenGate home:

```
curl -X PATCH \  
  https:// ://<hostname>:<port>/services/v2/deployments/<deployment name> \  
  -H 'cache-control: no-cache' \  
  -d '{"oggHome":"/opt/app/oracle/product/21.1.0/oggcore_1",  
      "status":"restart"}'
```

 **Note:**

You can confirm that the Oracle GoldenGate home was updated by looking at the process from the operating system for Service Manager. The Service Manager process should be running from the new Oracle GoldenGate home.

4. Start all distribution paths within the Distribution Server.
5. Start all Extracts and Replicats within the Administration Server.

After the Service Manager or Deployment restarts, you are upgraded to the next version.

# 6

## Setting Up Oracle GoldenGate for Big Data in a High Availability Environment

This topic describes the best practices of achieving high availability of Oracle GoldenGate for Big Data processes.

### Topics:

- [Running Oracle GoldenGate for Big Data from a Single Instance](#)
- [Running Oracle GoldenGate for Big Data on a Cluster of Servers](#)
- [Shared Storage](#)

Most shared storage solutions, including general purpose cluster file systems, can be used to install Oracle GoldenGate or to store the files that Oracle GoldenGate needs to recover.

### 6.1 Running Oracle GoldenGate for Big Data from a Single Instance

To configure the single server high availability, you need to configure the manager process with `AUTOSTART` and `AUTORESTART` parameters. These parameters ensure that the manager process always gets the extract or replicat group to be started back up from an inactive state.

### 6.2 Running Oracle GoldenGate for Big Data on a Cluster of Servers

Depending on which cluster manager software that is being used, you need to configure it to ensure the following:

- There is **exactly one active node** that is running Oracle GoldenGate for Big Data. It is assumed that the cluster manager can detect that a compute node is down and subsequently spawn another node to be the active.
- Install Oracle GoldenGate for Big Data in shared file system and have that shared file system mounted in the same location for all the nodes participating in the High Availability (HA) configuration. For more information about installing Oracle GoldenGate for Big Data, see *Installing Oracle GoldenGate for Big Data* in the *Installing and Upgrading Oracle GoldenGate for Big Data* guide. Most of the state files, including the Input and Output Trail files, Configuration files, and Checkpoint files described in the next point are stored in sub-directories of the Oracle GoldenGate for Big Data install. The Oracle GoldenGate for Big Data installation directory is the same across all managed nodes. This helps the administrator to leverage the exact content of entry point script to bring up Oracle GoldenGate for Big Data as part of its workflow to spawn a new active node. An example of the content of the entry point script is a command to start the Oracle GoldenGate manager process.

- Oracle GoldenGate artifacts are stored in one or more **shared file systems or volumes accessible from all nodes**. For more information about these files, see Directory Structure in the *Installing and Upgrading Oracle GoldenGate for Big Data* guide:
  - **Input and Output Trail files:** Typically these files are located in the `gg_install_dir/dirdat` directory, where `gg_install_dir` is the Oracle GoldenGate installation directory, such as `C:/ggs` on Windows or `/home/user/ggs` on UNIX. These files are configurable.
  - **Configuration files:** The configuration files are located in the `gg_install_dir/dirprm` directory.
  - **Checkpoint files:** These files are stored in an internal subdirectory, such as the `gg_install_dir/dirchk` directory.
  - When using File Writer features, for example, File Writer handler, ADW, or Redshift integration, the file writer output files and the state files must be on shared volumes.

For more information about configuring cluster high availability for handlers, see Configuring Cluster High Availability in *Using Oracle GoldenGate for Big Data* guide.

## 6.3 Shared Storage

Most shared storage solutions, including general purpose cluster file systems, can be used to install Oracle GoldenGate or to store the files that Oracle GoldenGate needs to recover.

The following options are available from Oracle:

- **Oracle Cluster File System (OCFS2) –available only on Linux:** OCFS2 can also be used for Oracle Database storage, although Oracle recommends the use of Oracle Automatic Storage Management (ASM) starting with Oracle Database 10g. For more information, see <http://oss.oracle.com/projects/ocfs2/>.
- **Oracle Automatic Storage Management (ASM) Cluster File System (ACFS):** For more information about the Oracle Database 11g Release 2 ACFS, see Oracle Database Automatic Storage Administrator's Guide as part of the Oracle Database 11g Release 2 documentation set: [https://docs.oracle.com/cd/E11882\\_01/server.112/e18951/asmfs\\_util001.htm#OSTMG91000](https://docs.oracle.com/cd/E11882_01/server.112/e18951/asmfs_util001.htm#OSTMG91000).
- **Oracle Database File System (DBFS):** For more information about DBFS, its restrictions as well as how to configure a DBFS, see Oracle Database Secure File and Large Objects Developer's Guide from the Oracle Database 11g Release 2 documentation set: [https://docs.oracle.com/cd/E11882\\_01/appdev.112/e18294/adlob\\_fs.htm#BABDHGGJ](https://docs.oracle.com/cd/E11882_01/appdev.112/e18294/adlob_fs.htm#BABDHGGJ).
- **Oracle ACFS with Oracle Database 11g Release 2**

# 7

## Dependency Downloader

Utility scripts are located in the `{OGGBD install}/DependencyDownloader` directory to download client dependency jars for the various supported Oracle GoldenGate for Big Data integrations.

These scripts use Java and Apache Maven to download the dependency jars from the Maven Central Repository and other publicly available repositories (Hortonworks, Cloudera, Confluent).

### Topics:

- [Dependency Downloader Setup](#)
- [Running the Dependency Downloader Scripts](#)
- [Dependency Downloader Scripts](#)

### 7.1 Dependency Downloader Setup

To complete the Dependency Downloader setup:

1. To verify that Java is installed, execute the following from the command line: `java -version`.

#### Note:

The Dependency Downloader utility scripts require Java to run. Ensure that Oracle Java is downloaded and is available in the PATH on the machine where the scripts are installed.

2. Configure the proxy settings in the following script: `{OGGBD install}/DependencyDownloader/config_proxy.sh`. Following are the 2 entries in this file:

- `#export PROXY_SERVER_HOST=www-proxy-hqdc.us.oracle.com`
- `#export PROXY_SERVER_PORT=80`

To configure the proxy settings:

- a. Uncomment the configuration settings. (remove the # at beginning of the lines).
- b. Change the host name and port number to your correct proxy server settings.

 **Note:**

Most companies maintain a private network which in turn has a network firewall to shield it from the public Internet. Additionally, most companies maintain a forwarding proxy server which serves as a gateway between the customer's private network and the public Internet. The Dependency Downloader utilities must access Maven repositories, which are available on the Internet. Therefore, you need to supply configuration for HTTP proxy settings in order to download dependency libraries. Proxy servers are identified by host name and port. If you do not know whether your company employs a proxy server or the settings, then contact your IT or network administrators.

The Dependency Downloader uses Bash scripts in order to invoke Maven and download dependencies. The Bash shell is not supported natively from the Windows Command Prompt. You can run the Dependency Downloader scripts on Windows, but it requires the installation of a Unix emulator. A Unix emulator provides a Unix style command line on Windows and supports various flavors of the Unix shells including Bash. An option for Unix emulators is Cygwin, which is available free of charge. After Cygwin is installed, the setup process is the same. Setup and running of the scripts should be done through the Cygwin64 Terminal. See <https://www.cygwin.com/>.

## 7.2 Running the Dependency Downloader Scripts

To run the dependency downloader scripts:

1. Use a Unix terminal interface navigate to the following directory: `{OGGBD install}/DependencyDownloader`.
2. Execute the following to run the scripts: `./{the dependency script} {version of the dependencies to download}`

For example: `./aws.sh 1.11.893`

Dependency libraries get downloaded to the following directory:

```
{OGGBD install}/DependencyDownloader/dependencies/{the dependency name}_{the_dependency_version}.
```

For example: `{OGGBD install}/DependencyDownloader/dependencies/aws_sdk_1.11.893.`

Ensure that the version string exactly matches the version string of the dependency which is being downloaded. If a dependency version doesn't exist in the public Maven repository, then it is not possible to download the dependency and running the script results in an error. Most public Maven repositories support a web-based GUI whereby you can browse the supported versions of various dependencies. The exception is the Confluent Maven repository does not support a web-based GUI. This makes downloading dependencies challenging, because the version string is not independently verifiable through a web interface.

After the dependencies are successfully downloaded, you must configure the `gg.classpath` variable in the Java Adapter properties file to include the dependencies for the corresponding replicat process.

**Note:****Best Practices**

1. Whenever possible, use the exact version of the client libraries to the server/application integration to which you are connecting.
2. Prior to running the Dependency Downloader scripts, independently verify that the version string exists in the repository through the web GUI.

## 7.3 Dependency Downloader Scripts

**Table 7-1** Dependency Downloader Scripts

Client	Script	Description	Relevant Handlers	Versions Supported	Dependency Link
Amazon Web Services SDK	aws.sh	This script downloads the Amazon Web Services (AWS) SDK, which provides client libraries for connectivity to the AWS cloud.	Kinesis Handler S3 Event Handler	1.11.x	<a href="https://search.maven.org/artifact/com.amazonaws/aws-java-sdk">https://search.maven.org/artifact/com.amazonaws/aws-java-sdk</a>
Google BigQuery	bigquery.sh	This script downloads the required client libraries for Google BigQuery.	BigQuery Handler	1.x	<a href="https://search.maven.org/artifact/com.google.cloud/google-cloud-bigquery">https://search.maven.org/artifact/com.google.cloud/google-cloud-bigquery</a>
Cassandra DSE (Datastax Enterprise) Client	cassandra_dse.sh	This script downloads the Cassandra DSE client. Cassandra DSE is the for-purchase version of Cassandra available from Datastax.	Cassandra Handler	2.0.0 and higher	<a href="https://search.maven.org/artifact/com.datastax.dse/dse-java-driver-core">https://search.maven.org/artifact/com.datastax.dse/dse-java-driver-core</a>

Table 7-1 (Cont.) Dependency Downloader Scripts

Client	Script	Description	Relevant Handlers	Versions Supported	Dependency Link
Apache Cassandra Client	<code>cassandra.sh</code>	This script downloads the Apache Cassandra client.	Cassandra Handler	4.0.0 and higher	<a href="https://search.maven.org/artifact/com.datastax.oss/java-driver-core">https://search.maven.org/artifact/com.datastax.oss/java-driver-core</a>
Elasticsearch REST client	<code>elasticsearch_rest.sh</code>	This script downloads the Elasticsearch High Level Rest Client.	Elasticsearch Handler	7.x versions are currently supported	<a href="https://search.maven.org/artifact/org.elasticsearch.client/elasticsearch-rest-high-level-client">https://search.maven.org/artifact/org.elasticsearch.client/elasticsearch-rest-high-level-client</a>
Elasticsearch Transport Client	<code>elasticsearch_transport.sh</code>	This script downloads the Elasticsearch Transport Client.	Elasticsearch Handler	5.x, 6.x, and 7.x	<a href="https://search.maven.org/artifact/org.elasticsearch.client/transport">https://search.maven.org/artifact/org.elasticsearch.client/transport</a>
Hadoop Azure Client from Cloudera	<code>hadoop_azure_cloudera.sh</code>	This script downloads the Hadoop Azure client libraries provided by Cloudera. The Hadoop Azure client libraries cannot be loaded along with the Hadoop client because in Cloudera, the version numbers between the two components do not line up perfectly.	<ul style="list-style-type: none"> <li>• HDFS Handler</li> <li>• HDFS Event Handler</li> <li>• ORC Event Handler</li> <li>• Parquet Event Handler</li> </ul>	2.x and 3.x	<a href="https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/hadoop/hadoop-azure/">https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/hadoop/hadoop-azure/</a>



Table 7-1 (Cont.) Dependency Downloader Scripts

Client	Script	Description	Relevant Handlers	Versions Supported	Dependency Link
Hadoop Client from Cloudera	<code>hadoop_cloudera.sh</code>	This script downloads the Hadoop client libraries provided by Cloudera.	<ul style="list-style-type: none"> <li>• HDFS Handler</li> <li>• HDFS Event Handler</li> <li>• ORC Event Handler</li> <li>• Parquet Event Handler</li> </ul>	2.x and 3.x	<a href="https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/hadoop/hadoop-client/">https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/hadoop/hadoop-client/</a>
Hadoop Client from Hortonworks	<code>hadoop_hortonworks.sh</code>	The Hadoop client including the libraries for connectivity to Azure Data Lake available from Hortonworks.	<ul style="list-style-type: none"> <li>• HDFS Handler</li> <li>• HDFS Event Handler</li> <li>• ORC Event Handler</li> <li>• Parquet Event Handler</li> </ul>	2.x and 3.x	<a href="https://repo.hortonworks.com/content/groups/public/org/apache/hadoop/hadoop-azure/">https://repo.hortonworks.com/content/groups/public/org/apache/hadoop/hadoop-azure/</a>
Apache Hadoop Client Plus Required Libraries for Azure Connectivity	<code>hadoop.sh</code>	The Hadoop client including the libraries for connectivity to Azure Data Lake.	<ul style="list-style-type: none"> <li>• HDFS Handler</li> <li>• HDFS Event Handler</li> <li>• ORC Event Handler</li> <li>• Parquet Event Handler</li> </ul>	2.7.x and higher, and 3.x	<a href="https://search.maven.org/artifact/org.apache.hadoop/hadoop-azure">https://search.maven.org/artifact/org.apache.hadoop/hadoop-azure</a>
HBase Client Provided by Cloudera	<code>hbase_cloudera.sh</code>	The HBase client libraries provided by Cloudera.	HBase Handler	1.x and 2.x	<a href="https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/hbase/hbase-client/">https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/hbase/hbase-client/</a>
HBase Client Provided by Hortonworks	<code>hbase_hortonworks.sh</code>	The HBase client libraries provided by Hortonworks.	HBase Handler	1.x and 2.x	<a href="https://repo.hortonworks.com/content/groups/public/org/apache/hbase/hbase-client/">https://repo.hortonworks.com/content/groups/public/org/apache/hbase/hbase-client/</a>

Table 7-1 (Cont.) Dependency Downloader Scripts

Client	Script	Description	Relevant Handlers	Versions Supported	Dependency Link
Apache HBase Client	hbase.sh	The HBase client.	HBase Handler	1.x and 2.x	<a href="https://search.maven.org/artifact/org.apache.hbase/hbase-client">https://search.maven.org/artifact/org.apache.hbase/hbase-client</a>
Apache Kafka Client plus Kafka Connect Framework and JSON Converter from Cloudera	kafka_cloudera.sh	The Kafka Client plus libraries for the Kafka Connect framework and the Kafka Connect JSON Converter provided by Cloudera.	<ul style="list-style-type: none"> <li>• Kafka Handler</li> <li>• Kafka Connect Handler</li> <li>• Kafka Capture</li> </ul>	0.9.x to current	<a href="https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/kafka/kafka-clients/">https://repository.cloudera.com/artifactory/cloudera-repos/org/apache/kafka/kafka-clients/</a>
Apache Kafka Client plus Kafka Connect Framework and JSON Converter from Hortonworks	kafka_hortonworks.sh	The Kafka Client plus libraries for the Kafka Connect framework and the Kafka Connect JSON Converter provided by Hortonworks	<ul style="list-style-type: none"> <li>• Kafka Handler</li> <li>• Kafka Connect Handler</li> <li>• Kafka Capture</li> </ul>	0.9.x to current	<a href="https://repo.hortonworks.com/content/groups/public/org/apache/kafka/kafka-clients/">https://repo.hortonworks.com/content/groups/public/org/apache/kafka/kafka-clients/</a>
Apache Kafka Client plus Kafka Connect Framework and JSON Converter	kafka.sh	The Kafka Client plus libraries for the Kafka Connect framework and the Kafka Connect JSON Converter.	<ul style="list-style-type: none"> <li>• Kafka Handler</li> <li>• Kafka Connect Handler</li> <li>• Kafka Capture</li> </ul>	0.9.x to current	<a href="https://search.maven.org/artifact/org.apache.kafka/kafka-clients">https://search.maven.org/artifact/org.apache.kafka/kafka-clients</a>

Table 7-1 (Cont.) Dependency Downloader Scripts

Client	Script	Description	Relevant Handlers	Versions Supported	Dependency Link
Confluent Kafka Client plus Kafka Connect Framework and JSON and Avro Converters	kafka_conf_luent.sh	The Kafka Client plus libraries for the Kafka Connect framework and the Kafka Connect JSON Converter and the Kafka Connect Avro Converter available from Confluent.	<ul style="list-style-type: none"> <li>Kafka Handler</li> <li>Kafka Connect Handler</li> <li>Kafka Capture</li> </ul>	Confluent platform 4.1.0 and higher.	The Confluent Maven repository does not provide a web GUI interface.
MongoDB Client	mongodb.sh	The MongoDB client libraries.	MongoDB Handler	3.x	<a href="https://search.maven.org/artifact/org.mongodb/mongo-java-driver">https://search.maven.org/artifact/org.mongodb/mongo-java-driver</a>
Oracle NoSQL Client	oracle_nosql.sh	The Oracle NoSQL client libraries.	Oracle NoSQL Handler	3.x, 4.x, and 18.x	<a href="https://search.maven.org/artifact/com.oracle.kv/oracle-nosql-client">https://search.maven.org/artifact/com.oracle.kv/oracle-nosql-client</a>
Oracle OCI Client	oracle_oci.sh	The Oracle OCI client libraries.	Oracle OCI Event Handler	1.x	<a href="https://search.maven.org/artifact/com.oracle.oci.sdk/oci-java-sdk-full">https://search.maven.org/artifact/com.oracle.oci.sdk/oci-java-sdk-full</a>

Table 7-1 (Cont.) Dependency Downloader Scripts

Client	Script	Description	Relevant Handlers	Versions Supported	Dependency Link
Apache ORC (Optimized Row Columnar) Client	<code>orc.sh</code>	The Apache ORC client libraries. ORC is built on top of the Hadoop client so the ORC Event Handler needs the Hadoop client in order to run. The Hadoop client needs to be downloaded separately.	ORC Event Handler	1.x	<a href="https://search.maven.org/artifact/org.apache.orc/orc-core">https://search.maven.org/artifact/org.apache.orc/orc-core</a>
Apache Parquet Client	<code>parquet.sh</code>	The Apache Parquet client libraries. Parquet is built on top of the Hadoop client, so the Parquet Event Handler needs the Hadoop client in order to run. The Hadoop client needs to be downloaded separately.	Parquet Event Handler	1.x	<a href="https://search.maven.org/artifact/org.apache.parquet/parquet-hadoop">https://search.maven.org/artifact/org.apache.parquet/parquet-hadoop</a>