

A person's head and shoulders are visible from a side-on perspective. They are wearing a VR headset with a screen showing a vibrant, abstract pattern of orange, yellow, blue, and white. The background behind them is dark.

ORACLE



Oracle OpenWorld 2019

SAN FRANCISCO

Safe Harbor

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.



TRN6112

Oracle GoldenGate for Big Data

Overview

Randall Richeson

Senior Principal Instructor
Oracle University

What's Ahead

Today

Introduction to GoldenGate for Big Data

Configuring and Using the Apache Kafka Handler



Our Goal

Explain GoldenGate for Big Data Architecture

Show how to configure GoldenGate to
replicate changes from Oracle Database redo
logs to a Kafka topic.

Introduction to GoldenGate for Big Data

- Oracle GoldenGate for Big Data: Product Overview
- Describing the Java Adapter and Oracle GoldenGate for Big Data
- Configuring and Using the Java Adapter
- Configuring Message Capture
- Message Parsing
- Configuring Message Delivery

Product Overview

Oracle GoldenGate for Big Data is a module that adds **pluggable functionality** to the Oracle GoldenGate Java Adapters framework.

Traditionally, the **Java Adapters** framework read the trail files produced by core GoldenGate instances and helped integration with **JMS queues and/or files**.

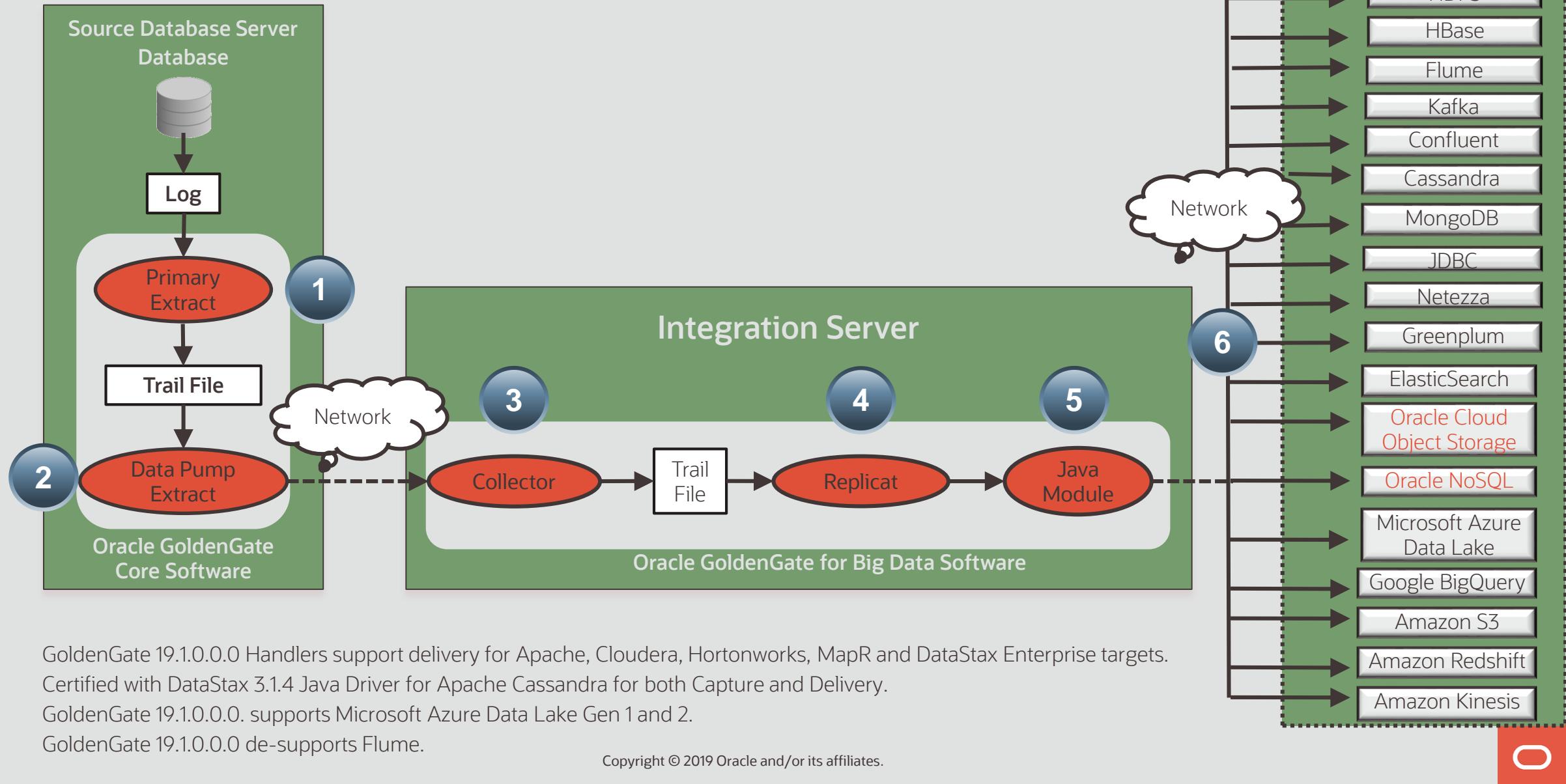
Oracle **GoldenGate for Big Data** extends the Java Adapters framework, ensuring interoperability with the Hadoop ecosystem, Apache and DataStax Enterprise Cassandra, MongoDB, Generic JDBC, and other targets.



Integration with Oracle GoldenGate Core Instances

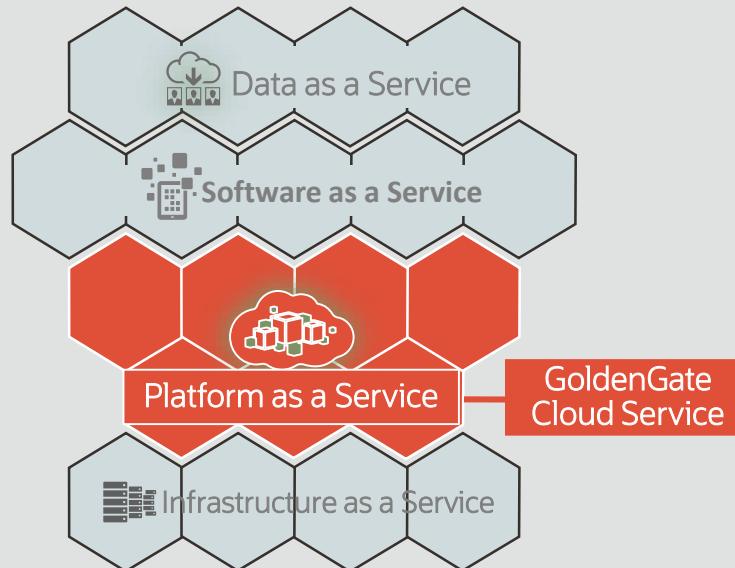
- The Oracle GoldenGate core software enables capturing transactional changes from a source database and sending those changes as a set of database-independent files called trail files.
- Altering the source data by using mapping parameters, functions, and filtering is an option.
- The Oracle GoldenGate Java Adapters framework and the Big Data plug-in can read the trail files and deliver the transactional content to Hadoop, NoSQL databases, JDBC and other big data targets.

Oracle GoldenGate for Big Data: Typical Data Flow



Oracle GoldenGate Cloud Service

PaaS Solution



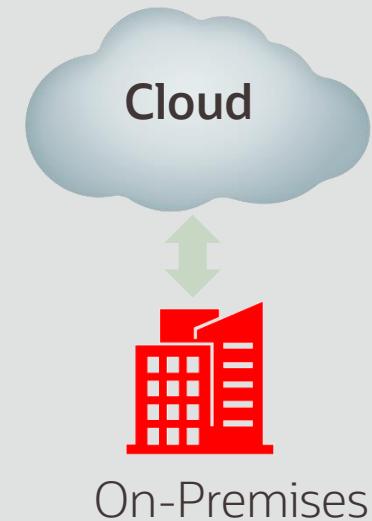
Deep Integration
In a Broader Solution

Built on GoldenGate



High Level Standard
Industry Proven Solution
Secure, Robust, and Scalable

Hybrid Deployment



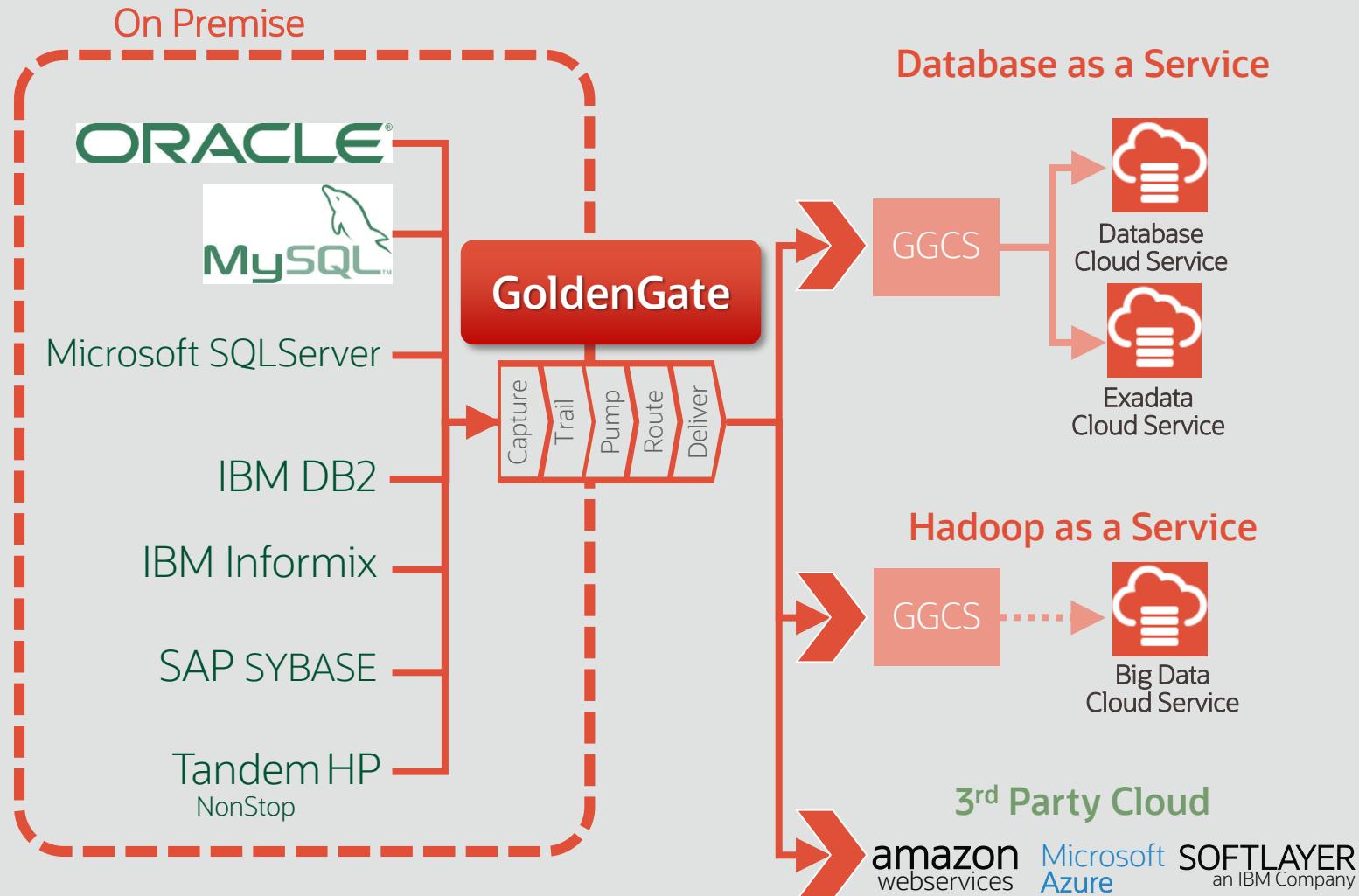
On-Premises

Consistent Standard
Architecture and Products

Oracle GoldenGate Cloud Service

- Real-Time Data Replication for Cloud
- Subscription-based **real-time data replication service** for Cloud
- Non-intrusive **log-based change data capture**
- High throughput encrypted and compressed **data transport via secure TCP/IP** between on-premises and cloud
- High performance **parallel data delivery**
- Easy to use **Web UI for provisioning** and managing sandbox environments
- **Prebuild integration** within Oracle Database Cloud Service

Oracle GoldenGate Cloud Service



GoldenGate Cloud Service

- ✓ Available in the Public Cloud via Subscription or Hourly basis

Key Benefits

- ✓ Oracle Database Cloud Service delivery
- ✓ Exadata Cloud Service delivery
- ✓ Big Data Cloud Service delivery to Hadoop and NoSQL

3rd Party Cloud

- ✓ More Choices run as BYOL on other Clouds for delivery to any supported Database.

Integration with Oracle Database Cloud Service Example

Database Configuration

* DB Name ORCL
* PDB Name PDB1
* Administration Password ?
* Confirm Password ?
* Usable Database Storage (GB) 50 ?
Total Data File Storage (GB) 260 ?
* Compute Shape VM.Standard2.1 - 1.0 OCPU, 15. ?
* SSH Public Key ssh-rsa AAAAB3NzaC1yc2EAA Edit ?

Advanced Settings

* Listener Port 1521 ?
* Timezone (UTC) Coordinated Universal Time ?
* Character Set AL32UTF8 - Unicode Universal ?
* National Character Set AL16UTF16 - Unicode UTF-16 L ?
Enable Oracle GoldenGate ?
Include "Demos" PDB ?

Backup and Recovery Configuration

* Backup Destination None ?

Initialize Data From Backup

* Create Instance from Existing Backup No ?

SQL> select NAME, SUPPLEMENTAL_LOG_DATA_MIN, CDB, FORCE_LOGGING from v\$database;

NAME	SUPPLEMENTAL CDB FORCE_LOGGING
ORCL	YES YES

SQL> sho parameter enable_gold

NAME	TYPE	VALUE
enable_goldengate_replication	boolean	TRUE

SQL> sho pdbs

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ WRITE	NO

SQL>

GoldenGate Cloud Service Console Example

The screenshot shows the Oracle Cloud GoldenGate Cloud Service console interface. At the top, there is a navigation bar with links like 'Most Visited', 'Classroom Support L...', 'Global Education', 'Oracle Online Evalu...', and 'OUKC'. Below the navigation bar is a blue header bar with the 'GoldenGate Cloud Service' logo and 'QuickStarts Welcome!' text. The main content area has two tabs: 'Instances' (which is selected) and 'Activity'. A timestamp 'As of Aug 22, 2019 3:19:47 PM UTC' is shown with a refresh icon. Below the tabs, the word 'Instances' is displayed. On the right side of this section is a 'Create Instance' button. A large text block says: 'You don't have any instances. After meeting the [prerequisites](#), use this button to create an instance.' A blue curved arrow points from this text towards the 'Create Instance' button. Below this, there is a section titled 'Need help creating the instance?' with two options: 'Watch a video' and 'Step through a tutorial'. The bottom of the page includes a copyright notice 'Copyright © 2019 Oracle and/or its affiliates.' and the Oracle logo.

GoldenGate Cloud Service supports Classic and Microservices Architectures

Create Instance

[Cancel](#)



[Next >](#)

Instance

Provide basic instance information

* Instance Name



Service Description



Notification Email



* Region



IP Network



Tags



GoldenGate Components selection

Replication only

Select the provisioning Choice of GoldenGate Version

GoldenGate Version

12.2

18.1 Classic

18.1 Microservices

GoldenGate Cloud Ser 18.1 Classic

Subscribe to a new GoldenGate Cloud Service software license and the GoldenGate Cloud Service.

...

...

GoldenGate Cloud Service Compute Shape Options

Create Instance

Previous Cancel Details Confirm Next >

Service Details

Some settings are dependent on current region, uscom-east-1. Go back to select a different region.

Selection Summary

Backup and Recovery Configuration

Backup Destination: None

Replication Node Configuration

Node Description: OC3 - 1.0 OCPU, 7.5GB RAM
OC4 - 2.0 OCPU, 15.0GB RAM
OC5 - 4.0 OCPU, 30.0GB RAM
OC6 - 8.0 OCPU, 60.0GB RAM
OC7 - 16.0 OCPU, 120.0GB RAM

* SSH Public Key

* Compute Shape: OC3 - 1.0 OCPU, 7.5GB RAM

Reserved IPs: Assign Automatically



GoldenGate Cloud Service Confirmation

Create Instance

[Previous](#) [Cancel](#)



[Create](#)

Confirmation

Confirm your responses and create this GoldenGate Service instance.



This instance is being created using [BYOL terms](#).

Service

Instance Name:	amer
Service Description:	amer
Bring Your Own License:	BYOL
Service Level:	GoldenGate Cloud Service
SSH Public Key:	ssh-rsa AAAAB3NzaC1yc2EAAA...
Region:	uscom-east-1
IP Network:	No Preference
Tags:	training ...

GoldenGate Version Details

GoldenGate Version: 18.1 Classic

Replication Node Configuration

Compute Shape: OC5 - 4.0 OCPU, 30.0GB RAM

GoldenGate Cloud Service Summary

GoldenGate Cloud Service

QuickStarts Welcome!

Instances Activity

As of Aug 22, 2019 4:35:32 PM UTC [↻](#)

Summary	1 Instances	4 OCPUs	30 GB Memory	408 GB Storage	1 Public IPs
---------	-------------	---------	--------------	----------------	--------------

Instances

Instance Name [▼](#) Search by instance name or tags [🔍](#) [?](#) [Create Instance](#)

 amer	Version: 19.1.3.0.0-190128 Edition: Enterprise Edition Tags: training ...	Created On: Aug 22, 2019 3:35:25 PM UTC	OCPUs: 4 Memory: 30 GB Storage: 408 GB
---	--	--	---



Introduction to GoldenGate for Big Data

- Oracle GoldenGate for Big Data: Product Overview
- Describing the Java Adapter and Oracle GoldenGate for Big Data
- Configuring and Using the Java Adapter
- Configuring Message Capture
- Message Parsing
- Configuring Message Delivery

Oracle GoldenGate Adapter Architecture

- Two products are based on the Oracle GoldenGate Adapter.

The [Oracle GoldenGate Java Adapter](#), which is the overall framework, helps you to implement custom code to handle trail records according to their specific requirements.

[Oracle GoldenGate for Big Data](#), which contains built-in support to write operation data from Oracle GoldenGate trail records into various Big Data targets and NoSQL databases.

- You do not need to write any custom code to integrate GoldenGate with Big Data and NoSQL applications.

Crossover Functionality

Feature	OGG Java Adapter	OGG for Big Data
Read JMS messages and deliver them as an Oracle GoldenGate trail.	✓	✓
Read an Oracle GoldenGate trail and deliver transactions to a JMS provider or other messaging system or custom application.	✓	✓
Read an Oracle GoldenGate trail and write transactions to a file that can be used by other applications.	✓	✗
Read an Oracle GoldenGate trail and write transactions to Big Data targets.	✗	✓

Oracle GoldenGate for Big Data 12.3.2.1 there is a new Flat file writer designed to load to a local file system and then load completed files to another location like HDFS.

Introduction to GoldenGate for Big Data

- Oracle GoldenGate for Big Data: Product Overview
- Describing the Java Adapter and Oracle GoldenGate for Big Data
- **Configuring and Using the Java Adapter**
- Configuring Message Capture
- Message Parsing
- Configuring Message Delivery

Oracle GoldenGate Java Adapter: Message Capture and Delivery

Oracle GoldenGate message **capture** reads messages from a JMS queue and communicates with an Oracle GoldenGate Extract process to generate a set of trail files that contain the processed data.

The message **capture** processing is implemented as a Vendor Access Module (VAM) plug-in to a generic Extract process.

Oracle GoldenGate Java Adapter **delivery** applies transactional changes to targets other than a relational database that can be used for ETL tools (DataStage, Ab Initio, Informatica), Integration (JMS messaging), Analytics (Big Data), or custom APIs.

The **delivery** occurs through a dynamically linked or shared library and a set of Java libraries used to communicate with the Replicat process using the Java Native Interface (JNI).



Oracle GoldenGate Java Adapter: No Custom Code Required !

- All Oracle GoldenGate Java Adapters, Big Data Handlers, and formatters are configured through **predefined** properties, which are stored in property files located in the standard Oracle GoldenGate directory (dirprm).
- Writing to Big Data and NoSQL targets in various formats is configured by using a set of properties and requires **no custom programming**.
- Replicating relational tables into JMS queues is also accomplished through **predefined** properties.
- You can use **gendef** to create an Oracle GoldenGate source definitions file to parse the resulting trail written by the Extract process.

Introduction to GoldenGate for Big Data

- Oracle GoldenGate for Big Data: Product Overview
- Describing the Java Adapter and Oracle GoldenGate for Big Data
- Configuring and Using the Java Adapter
- **Configuring Message Capture**
- Message Parsing
- Configuring Message Delivery

Configuring the Vendor Access Module (VAM) Extract

- JMS capture works with the Oracle GoldenGate Extract process.
- The prerequisites to run the Java message capture process include:

Oracle GoldenGate for Java Adapter

Extract process

Extract parameter file configured for message capture

Source definitions file created with [gendef](#) or a COBOL copybook showing incoming format

[Java 8](#) or higher installed on the host machine

Adding the VAM Extract

- The GGSCI utility is used to add the message capture **VAM**.

```
ADD EXTRACT extvam, VAM
```

```
ADD EXTTRAIL dirdat/id, EXTRACT extvam, MEGABYTES 100
```

- The process name (EXTVAM in the example) can be replaced with any name that is no more than **eight** characters.
- The trail identifier (id in the example) can be any **two** characters.

VAM Extract Parameters

Parameter	Explanation
EXTRACT <Extract Name>	Is the name of the Extract process
VAM libgjava_vam.so PARAMS (dirprm/<ext name>.properties)	Specifies the name of the VAM library and the location of the properties file
TranLogOptions VAMCOMPATIBILITY1	Specifies that the original (1) implementation of the VAM is to be used
TranLogOptions GETMETADATAFROMVAM	Specifies that metadata will be sent by the VAM
ExtTrail dirdat/<xx>	Specifies the identifier of the target trail that the Extract process creates
Table <schema>.<table>;	Specifies metadata information about tables

VAM Properties File

- To process JMS messages:

Configure the connection to the JMS interface

Retrieve and parse the messages in a transaction

Write each message to a trail

Commit the transaction

Remove its messages from the queue

- You configure message capture using the VAM properties file.

This file is identified by the **PARAMS** option of the Extract **VAM** parameter.

It determines logging characteristics, parser mappings, and JMS connection settings.

VAM Properties File: Example

```
...Omitted lines...
### JMS settings
gg.jms.connectionFactory=ConnectionFactory
gg.jms.destination=dynamicQueues/testWQ
java.naming.provider.url=tcp://localhost:61616?jms.prefetchPolicy.all=1000
java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQFactory
### native JNI library log configuration (output directory,
###                                     filename,prefix)
goldengate.log.logname=dir rpt/extvam_jni
goldengate.log.level=INFO
goldengate.log.tostdout=false
goldengate.log.tofile=true
### XML parsing properties
parser.type=xml
xml.sourcedefs=dir def/tc.def
xml.rules=tx_rule
...Many more omitted lines...
```



Introduction to GoldenGate for Big Data

- Oracle GoldenGate for Big Data: Product Overview
- Describing the Java Adapter and Oracle GoldenGate for Big Data
- Configuring and Using the Java Adapter
- Configuring Message Capture
- **Message Parsing**
- Configuring Message Delivery

Message Parsing: Overview

The **parser** translates JMS text message data and header properties into an appropriate set of transactions and operations to pass into the **VAM** interface.

Several **bits** of information must be made available to the message **parser**:

- Transaction Identifier
- Sequence Identifier
- Time stamp
- Table Name
- Operation Type
- Column Data

The **parser** obtains this data from JMS header properties, system generated values, static values, or in some parser-specific way.

Parser Types

- The Oracle GoldenGate message capture adapter supports three types of parsers:
 - Fixed: Messages contain data presented as fixed width fields in contiguous text.
 - Delimited: Messages contain data delimited by field and end-of-record characters.
 - XML: Messages contain XML data accessed through XPath expressions.
- Configured properties determine how the selected **parser** gets data and how the source definitions are converted to target definitions.
- These properties are configured in the **VAM properties file**.

Parser Properties Example

...Omitted lines...

```
### delimited parser properties
parser.type=delimited
delim.schematype=sourcedefs
delim.sourcedefs=dirdef/tc.def
delim.header=schemaandtable,timestamp,optype
delim.optype=optype
delim.seqid=*seqid
delim.quote="
delim.quote.escaped=\"
delim.timestamp=timestamp
delim.txid=*txid
delim.fielddelim=,
delim.fielddelim.escaped=\
delim.linedelim=\X0a
```

...Many more omitted lines...

Agenda: Introduction to GoldenGate for Big Data

- Oracle GoldenGate for Big Data: Product Overview
- Describing the Java Adapter and Oracle GoldenGate for Big Data
- Configuring and Using the Java Adapter
- Configuring Message Capture
- Message Parsing
- **Configuring Message Delivery**

Replicat for Java Delivery

Java Delivery is compatible with the **Replicat** process. Transaction data is read from the trail files and delivered to the Java Delivery module across the JNI interface.

The Java Delivery module can be configured to stream data into various targets (JMS, file writing, and custom integrations).

Oracle GoldenGate for Big Data helps streaming data to Big Data, NoSQL, and other targets.

The **SPECIALRUN** parameter is supported for JMS and other targets.

Adding the Java Delivery Replicat

- The **GGSCI** utility is used to add the delivery Replicat.

```
ADD REPLICAT javarep, EXTTRAIL ./dirdat/bb
```

- Replicat consumes a local trail (**dirdat/bb**) and sends the data to the Java Delivery module which then processes the data and applies it to the target.
- The Replicat process name must be **eight** characters or less, and the trail file name must be **two** characters.
- An **optional** java delivery checkpoint file is used instead of a checkpoint table.

Replicat Parameters

Parameter	Explanation
REPLICAT <Replicat Name>	All Replicat parameter files start with the Replicat name.
SOURCEDEFS ./dirdef/xxx.def	If the input trail files do not contain the metadata records, the Replicat process requires metadata describing the trail data (optional parameter).
TARGETDB LIBFILE libggjava.so SET property=dirprm/<file>	The TARGETDB LIBFILE libggjava.so parameter serves as a trigger to initialize the Java module. The SET subcommand specifies the properties file for the Java Adapter.
MAP schema.* , TARGET *.*;	This specifies the tables to pass to the Java module; tables not included will be skipped.
GROUPTRANSOPS 1000	This groups source transactions into a single larger target transaction for improved performance.



Replicat Parameter File: Example

```
REPLICAT repkfk
TARGETDB LIBFILE libggjava.so SET property=dirprm/kafka.properties
REPORTCOUNT EVERY 1 MINUTES, RATE
GROUPTRANSOPS 10000
MAP QASOURCE.* , TARGET QASOURCE.*;
```



Oracle GoldenGate Java Handlers

Java handlers provide the functionality to **push** data to integration targets such as JMS or Big Data/NoSQL applications.

Multiple handlers may be separated by commas, and used for debugging or replicating the same information concurrently to multiple targets.

The Java Adapter **properties file** is used to configure Java Delivery and Java handlers.

Custom handlers can be implemented, in which case the type would be the fully qualified name of the **Java** class for the handler.



Replicat Java Handler: Kafka Example

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.mode=tx
gg.handler.kafka.topicMappingTemplate=oggtopic
gg.handler.kafka.format=avro_op
gg.handler.kafka.SchemaTopicName=mySchemaTopic
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
...Many more omitted lines...
```

Demos: Overview

1-1: Starting and Administering the Hadoop Single-Node Cluster

Verify Big Data single node cluster with Kafka, HBase, and HDFS.

1-2: Installing Oracle GoldenGate for Big Data

Install the Oracle GoldenGate for Big Data from the distribution kit.



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 1

Configure password-less connection to machine using ssh-keygen.

```
[oracle@edvmr1p0 ~]$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/home/oracle/.ssh'.
Your identification has been saved in /home/oracle/.ssh/id_rsa.
Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
fb:f2:fa:2f:22:f6:3b:9c:58:cf:52:15:5c:bd:3c:a5 oracle@edvmr1p0
The key's randomart image is:
++-[ RSA 2048]---+
 . . .
 o  o
 . .o
 . E+
 S . .
 ...
 +.=
 + B.+
 . o+X+o.
+-----+
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ chmod 0600 ~/.ssh/authorized_keys
[oracle@edvmr1p0 ~]$ █
```



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 2

Test password-less connection to machine using ssh.

```
[oracle@edvmr1p0 ~]$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is b6:f7:35:01:50:43:8b:97:73:73:c6:cc:c6:ee:d5:2d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
[oracle@edvmr1p0 ~]$ exit
logout
Connection to localhost closed.
[oracle@edvmr1p0 ~]$ ssh 0.0.0.0
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
RSA key fingerprint is b6:f7:35:01:50:43:8b:97:73:73:c6:cc:c6:ee:d5:2d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '0.0.0.0' (RSA) to the list of known hosts.
Last login: Mon Aug 19 18:14:45 2019 from localhost.localdomain
[oracle@edvmr1p0 ~]$ exit
logout
Connection to 0.0.0.0 closed.
[oracle@edvmr1p0 ~]$
```



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 3

Start the Apache Hadoop cluster and HBase RegionServer.

```
[oracle@edvmr1p0 ~]$ start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /opt/hadoop/logs/hadoop-oracle-namenode-edvmr1p0.out
localhost: starting datanode, logging to /opt/hadoop/logs/hadoop-oracle-datanode-edvmr1p0.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-oracle-secondarynamenode-edvmr1p0.out
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop/logs/yarn-oracle-resourcemanager-edvmr1p0.out
localhost: starting nodemanager, logging to /opt/hadoop/logs/yarn-oracle-nodemanager-edvmr1p0.out
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ start-hbase.sh
localhost: starting zookeeper, logging to /opt/hbase/logs/hbase-oracle-zookeeper-edvmr1p0.out
starting master, logging to /opt/hbase/logs/hbase-oracle-master-edvmr1p0.out
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
starting regionserver, logging to /opt/hbase/logs/hbase-oracle-1-regionserver-edvmr1p0.out
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ $JAVA_HOME/bin/jps
9136 SecondaryNameNode
8816 NameNode
9297 ResourceManager
9410 NodeManager
10184 HRegionServer
10056 HMaster
9930 HQuorumPeer
8955 DataNode
10523 Jps
[oracle@edvmr1p0 ~]$
```



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 3

Start the Apache Hadoop cluster and HBase RegionServer.

```
[oracle@edvmr1p0 ~]$ start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /opt/hadoop/logs/hadoop-oracle-namenode-edvmr1p0.out
localhost: starting datanode, logging to /opt/hadoop/logs/hadoop-oracle-datanode-edvmr1p0.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-oracle-secondarynamenode-edvmr1p0.out
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop/logs/yarn-oracle-resourcemanager-edvmr1p0.out
localhost: starting nodemanager, logging to /opt/hadoop/logs/yarn-oracle-nodemanager-edvmr1p0.out
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ start-hbase.sh
localhost: starting zookeeper, logging to /opt/hbase/logs/hbase-oracle-zookeeper-edvmr1p0.out
starting master, logging to /opt/hbase/logs/hbase-oracle-master-edvmr1p0.out
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
starting regionserver, logging to /opt/hbase/logs/hbase-oracle-1-regionserver-edvmr1p0.out
[oracle@edvmr1p0 ~]$
[oracle@edvmr1p0 ~]$ $JAVA_HOME/bin/jps
9136 SecondaryNameNode
8816 NameNode
9297 ResourceManager
9410 NodeManager
10184 HRegionServer
10056 HMaster
9930 HQuorumPeer
8955 DataNode
10523 Jps
[oracle@edvmr1p0 ~]$
```



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 4

Connect to the Apache Hadoop Resource Manager console.

The screenshot shows the Apache Hadoop Resource Manager (YARN) interface. The URL in the browser is `localhost:8088/cluster`. The page title is "All Applications". On the left, there's a sidebar with a yellow elephant icon and the word "hadoop". The sidebar has sections for "Cluster Metrics" and "Scheduler Metrics". Under "Cluster Metrics", there's a table with columns: Apps Submitted (0), Apps Pending (0), Apps Running (0), Apps Completed (0), Containers Running (0 B), Memory Used (8 GB), Memory Total (0 B), Memory Reserved (0), Vcores Used (8), Vcores Total (0), Vcores Reserved (0), Active Nodes (1), Decommissioned Nodes (0), Lost Nodes (0), Unhealthy Nodes (0), and Rebooted Nodes (0). Under "Scheduler Metrics", it shows "Capacity Scheduler" as the Scheduler Type and "[MEMORY]" as the Scheduling Resource Type. The Minimum Allocation is <memory:1024, vCores:1> and the Maximum Allocation is <memory:8192, vCores:32>. A table at the bottom shows "No data available in table". At the bottom of the sidebar, there's a "Tools" section with a "Scheduler" link. The footer says "Showing 0 to 0 of 0 entries" and has links for "First", "Previous", "Next", and "Last".



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 5

Create and verify directories in the HDFS file system.

```
[oracle@edvmr1p0 ~]$ hdfs dfs -mkdir /user
[oracle@edvmr1p0 ~]$ hdfs dfs -mkdir /user/oracle
[oracle@edvmr1p0 ~]$ hdfs dfs -ls /
Found 2 items
drwxr-xr-x  - oracle supergroup          0 2019-08-20 15:54 /hbase
drwxr-xr-x  - oracle supergroup          0 2019-08-20 16:31 /user
[oracle@edvmr1p0 ~]$ █
```

Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 6

Use the browser to access the Hadoop Name Node component.

localhost:50070/dfshealth.html#tab-overview

Most Visited ▾ Classroom Support ... Global Education Oracle Online Evalu... OUKC

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Browse the file system
Logs

Overview 'localhost:9000' (active)

Started:	Tue Aug 20 15:53:07 GMT 2019
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-2cb7c1b2-4f57-4bed-9610-d09d179c7098
Block Pool ID:	BP-1817849106-10.237.16.207-1485653398498

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	oracle	supergroup	0 B	Tue 20 Aug 2019 03:54:52 PM GMT	0	0 B	hbase
drwxr-xr-x	oracle	supergroup	0 B	Tue 20 Aug 2019 04:31:25 PM GMT	0	0 B	user

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	oracle	supergroup	0 B	Tue 20 Aug 2019 04:31:25 PM GMT	0	0 B	oracle



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 7

Start the Apache Kafka server.

```
oracle@edvmr1p0 ~]$ kafka-server-start.sh $KAFKA_HOME/config/server.properties
2019-08-20 17:13:19,850 INFO KafkaConfig values:
advertised.host.name = null
metric.reporters = []
quota.producer.default = 9223372036854775807
offsets.topic.num.partitions = 50
log.flush.interval.messages = 9223372036854775807
auto.create.topics.enable = true ← This is the default. The GoldenGate Kafka
controller.socket.timeout.ms = 30000 Handler can dynamically create topics !
log.flush.interval.ms = null
principal.builder.class = class org.apache.kafka.common.security.auth.DefaultPrincipalBuilder
replica.socket.receive.buffer.bytes = 65536
min.insync.replicas = 1
replica.fetch.wait.max.ms = 500
num.recovery.threads.per.data.dir = 1
ssl.keystore.type = JKS
sasl.mechanism.inter.broker.protocol = GSSAPI
default.replication.factor = 1
ssl.truststore.password = null
log.preallocate = false
sasl.kerberos.principal.to.local.rules = [DEFAULT]
fetch.purgatory.purge.interval.requests = 1000
ssl.endpoint.identification.algorithm = null
replica.socket.timeout.ms = 30000
message.max.bytes = 1000012
num.io.threads = 8
offsets.commit.required.acks = -1
log.flush.offset.checkpoint.interval.ms = 60000
There are many more lines to check for
default or user defined settings !!!
```



Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 8

Create a topic.

```
[oracle@edvmr1p0 ~]$ kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic ogg19cBigData
Created topic "ogg19cBigData".
[oracle@edvmr1p0 ~]$ ls -l /tmp/kafka-logs/ogg19cBigData-0/
total 0
-rw-r--r-- 1 oracle oinstall 10485760 Aug 20 17:33 00000000000000000000000000000000.index
-rw-r--r-- 1 oracle oinstall 0 Aug 20 17:33 00000000000000000000000000000000.log
[oracle@edvmr1p0 ~]$ █
```

The topic can also be pre-configured by creating it manually, which is an Oracle best practice !!!

Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 9

Send messages to Apache Kafka cluster.

```
[oracle@edvmr1p0 ~]$ kafka-console-producer.sh --broker-list localhost:9092 --topic ogg19cBigData
Testing Kafka in the context of OGG 19c for Big Data
Second message sent to the Kafka cluster for Oracle OpenWorld 2019
```

Demo 1-1: Starting and Administering the Hadoop Single-Node Cluster, Step 10

Receive the messages sent by the producer.

```
[oracle@edvmrlp0 ~]$ kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic ogg19cBigData --from-beginning --zookeeper localhost:2181
Testing Kafka in the context of OGG 19c for Big Data
Second message sent to the Kafka cluster for Oracle OpenWorld 2019
```

Configuring and Using the Oracle GoldenGate Apache Kafka Handler

- Describe the Oracle GoldenGate for Big Data Kafka Handler features
- Identify the Kafka-specific features and parameters
- Configure the Kafka Handler by using the relevant parameters in the property file
- Configure the transactional and operational modes that are available in the Kafka Handler
- Configure the Kafka Handler in Blocking or Non-Blocking mode

What is Apache Kafka ?

- Apache Kafka is a distributed, partitioned, and replicated **guaranteed** messaging service.
- Kafka can be run as a single instance or as a **cluster** on multiple servers.
- Each Kafka server instance is called a **broker**.
- A Kafka **topic** is a category or feed name to which messages are published by the producers and retrieved by consumers.

Oracle GoldenGate Kafka Handler Overview

- The **Kafka Handler** implements a Kafka producer
- The producer writes **serialized** change data capture from multiple source tables to either of the following:
 - A single configured topic
 - Different Kafka topics when the topic name corresponds to the fully qualified source table name

Kafka Core APIs

1. The **Producer API** helps an application to publish a stream of records to one or more Kafka topics.
2. The **Consumer API** helps an application to subscribe to one or more topics and process the stream of records produced to them.
3. The **Streams API** helps an application to act as a stream processor, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input stream to an output stream.
4. The **Connector API** helps for building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems.

Kafka Connect API

5. The Kafka Connect API is a functional layer on top of standard Kafka Producer and Consumer APIs.

Confluent is primary adopter of Kafka Connect.

Starting with [GoldenGate for Big Data 12.3.1.1](#), the [GoldenGate Kafka Connect Handler](#) can create schemas and messages [in memory](#) and pass them to the Kafka Connect converter to convert bytes to send to Kafka. It supports [JSON](#) and [AVRO](#) converters.

Kafka REST proxy API

6. The Kafka REST proxy API provides a RESTful interface to a Kafka cluster.

The Kafka REST Proxy Handler helps Kafka messages to be streamed using the **HTTPS** protocol and can be used to stream Kafka messages from an **Oracle GoldenGate On Premises** installation to cloud or alternately from cloud to cloud.

Kafka REST Proxy is part of the **Confluent Open Source** and **Confluent Enterprise** distributions and **not** available in the **Apache Kafka** distribution.

Oracle GoldenGate Kafka Handler Modes

- **Transaction Mode** serializes data for **every operation** in a **transaction** from the source Oracle GoldenGate trail files and **concatenates** them into a single value of the Kafka **ProducerRecord** object.
- **Operation Mode** serializes data for each **operation** is placed into an **individual** **ProducerRecord** object as the value. The **ProducerRecord** is immediately sent by using the Kafka Producer API.
- **Blocking Mode** delivers messages to Kafka on a **synchronous** basis.
- **Non-Blocking** Mode delivers messages to Kafka on an **asynchronous** basis.

Oracle GoldenGate Kafka Handler Classpath Configuration

- The `gg.classpath` configuration variable for the Kafka Handler must include:
 1. The location of the Kafka Producer properties file
 2. The location of the Kafka client JARs
- The recommended storage location for the Kafka Producer properties file is the Oracle GoldenGate `dirprm` directory.
- The default location of the Kafka client JARs is `$KAFKA_HOME/libs/*`.

Relevant Oracle GoldenGate Kafka Handler Configuration Parameters: 1

Parameter	Explanation
gg.handlerlist=kafka gg.handler.kafka.type=kafka	Selects the Kafka Handler for use with Replicat
gg.handler.kafka.KafkaProducerConfigFile	Specifies the file name in the classpath that holds the Apache Kafka properties to configure the Apache Kafka producer
gg.handler.kafka.topicMappingTemplate	Specifies the name of the Kafka topic where payload records will be sent. Templates allow you to configure static values and keywords.
gg.handler.kafka.Format	Specifies the formatter to use to format payload; can be one of xml, delimitedtext, json, json_row, avro_row, or avro_op
gg.handler.kafka.SchemaTopicName	Specifies the topic name where schema data will be delivered. Schemas will be propagated only for Avro formatters.



Relevant Oracle GoldenGate Kafka Handler Configuration Parameters: 2

Parameter	Explanation
gg.handler.kafka.BlockingSend	Is set to True or False. If this property is set to true, delivery to Kafka is made to work in a completely synchronous model.
gg.handler.kafka.mode	Is set to tx or op. With Kafka Handler operation mode, each change capture data record payload will be represented as a Kafka Producer Record and will be flushed one at a time.
gg.handler.kafka.topicMappingTemplate	A template string value to resolve the Kafka topic name at runtime.

Oracle GoldenGate Kafka Handler Properties File: Example

...Omitted lines...

```
gg.handlerlist=kafka
gg.handler.kafka.Type=kafka
gg.handler.kafka.KafkaProducerConfigFile=custom_kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggtopic
gg.handler.kafka.Format=avro_op
gg.handler.kafka.SchemaTopicName=oggSchemaTopic
gg.handler.kafka.Mode=tx
gg.handler.kafka.BlockingSend=true
...More omitted lines...
```

Oracle GoldenGate Kafka Handler Producer Configuration File

The Kafka **Handler** must access a Kafka producer configuration file to publish messages to Kafka ([KafkaProducerConfigFile](#) parameter).

- The Kafka Handler locates the Kafka producer configuration file using the Java classpath.
- The Java classpath must include the directory containing the Kafka producer configuration file.
- The Kafka **Handler** uses the properties specified in the Kafka producer configuration file to:
 - Resolve the host and port of the Kafka **brokers**
 - Control the behavior of the interaction between the [Kafka producer client](#) and the Kafka brokers

Resolve the host and port of the Kafka **brokers**

Control the behavior of the interaction between the [Kafka producer client](#) and the Kafka brokers

Oracle GoldenGate Kafka Handler Schema Propagation

- The Kafka Handler can publish schemas to a schema topic .
- **Avro** Row and Operation formatters are the **only** formatters enabled for schema publishing.

Oracle GoldenGate Kafka Handler Schema Propagation continued

When the Kafka Handler `schemaTopicName` property is set:

- The Avro schema for a specific table will be published the first time an operation for that table is encountered
- If the Kafka Handler receives a metadata change event, the schema is flushed
- If the Avro wrapping functionality is enabled, the generic wrapper Avro schema is published the first time any operation is encountered

Oracle GoldenGate Kafka Handler Best Practices

- Oracle recommends avoiding setting the `linger.ms` parameter in the Kafka producer configuration file when `gg.handler.kafka.BlockingSend` is set to `true`.
- When the value is true, each send blocks for at least `linger.ms` minutes leading to major performance issues. This is set in milliseconds.
- Oracle recommends preconfigured topics by manually creating them.

Oracle GoldenGate Kafka Handler Best Practices

For best performance, you should set:

- gg.handler.kafka.mode=op
- gg.handler.kafka.BlockingSend=false
- Replicat GROUPTRANSOPS to 10,000.

Oracle GoldenGate Kafka Handler Advanced Configuration for Security

You can secure the Kafka Handler using:

- Transport Layer Security (TLS)
- Secured Socket Layer (SSL)
- Kerberos

Oracle GoldenGate Kafka Handler Advanced Configuration for Metadata Change Events

- If a **schema topic** is configured and the formatter used supports schema propagation (**Avro row** and **Avro Operation** formatters), the next time an operation is encountered for a table for which the schema has changed, the updated schema is **published** to the schema topic.
- To support metadata change events, the Oracle GoldenGate process that is **capturing** changes in the source database must support the Oracle GoldenGate metadata in trail feature (**Oracle GoldenGate core release 12.2 or higher**).

Kafka Handler Advanced Configuration for Data Compression

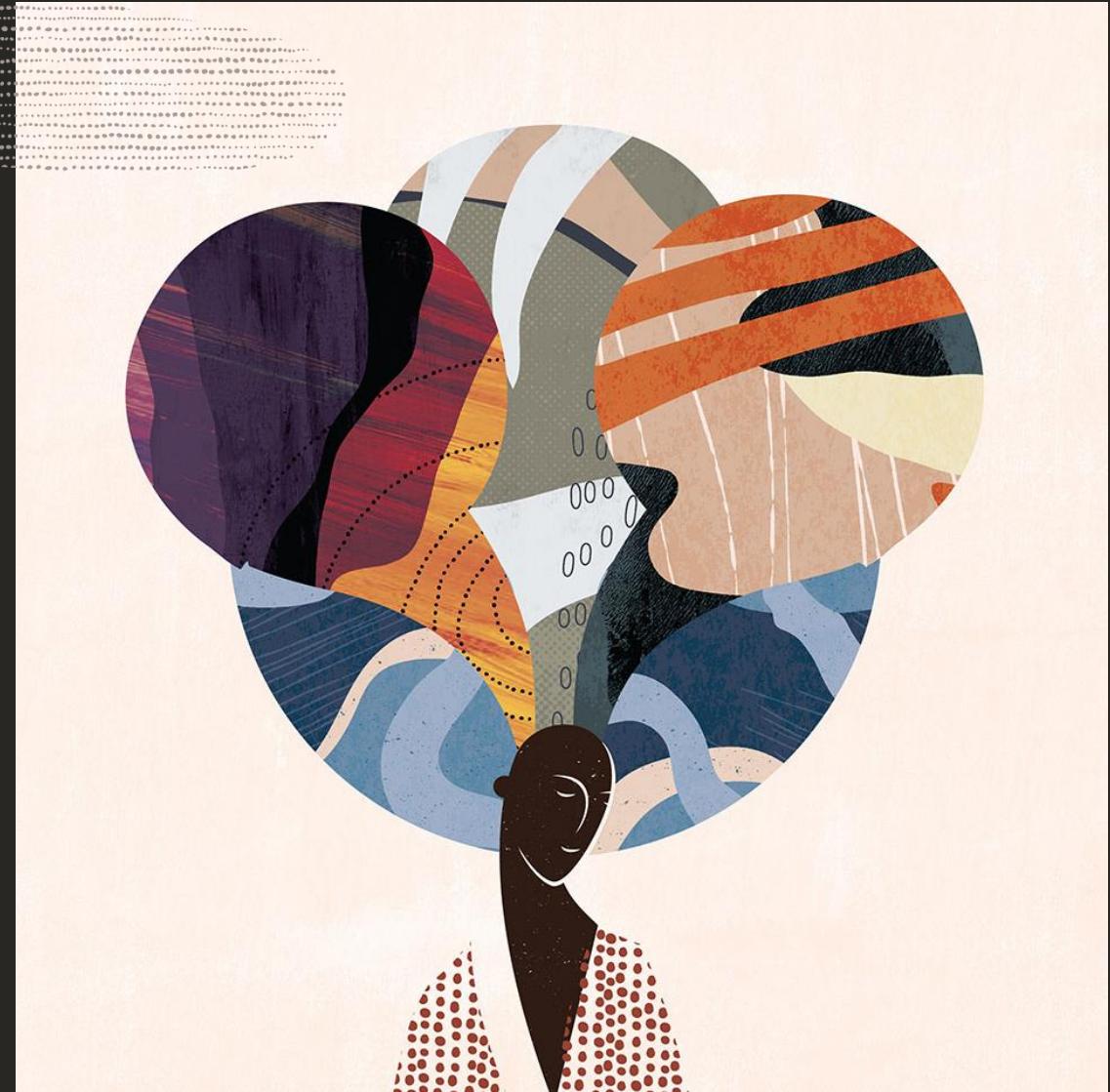
The Kafka producer configuration file supports the following compression methods:

- gzip
- Snappy ([Oracle recommends](#) this on Linux for performance reasons.)

Demos: Overview

2-1: Replicating Customer Data
from an Oracle RDBMS to
Apache Kafka

2-2: Performing and Verifying Updates



Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 1

Create the Java Adapter kafka.properties file

```
[oracle@edvmr1p0 oggtrg]$ pwd  
/u01/app/oggtrg  
[oracle@edvmr1p0 oggtrg]$ cd ./dirprm  
[oracle@edvmr1p0 dirprm]$ vi kafka.properties
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 2

Configure the parameters in the kafka.properties file.

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode =tx
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm/:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```



Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 3

Review the handler parameters in the kafka.properties file.

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode =tx
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```



Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 4

Review the handler parameters in the kafka.properties file.

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode =tx
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```



Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 5

Review the handler parameters in the kafka.properties file.

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode =tx
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm/:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 6

Review the gg.classpath parameter in the kafka.properties file.

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode =tx
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm/:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```



Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 7

Review the jvm.bootoptions parameter in the kafka.properties file.

```
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode =tx
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 8

Configure the parameters in the kafka_producer.properties file.

```
[oracle@edvmr1p0 dirprm]$ vi kafka_producer.properties
[oracle@edvmr1p0 dirprm]$ more kafka_producer.properties
bootstrap.servers=localhost:9092
acks=1
compression.type=gzip
reconnect.backoff.ms=1000
value.serializer=org.apache.kafka.common.serialization.ByteArraySerializer
key.serializer=org.apache.kafka.common.serialization.ByteArraySerializer
batch.size=102400
linger.ms=10000
max.request.size = 5024000
send.buffer.bytes = 5024000
```

This is used to connect to the Kafka broker.

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 9

Launch ggsci to assess the target environment.

```
[oracle@edvmr1p0 oggtrg]$ ggsci

Oracle GoldenGate for Big Data
Version 19.1.0.0.0 (Build 007)

Oracle GoldenGate Command Interpreter
Version 19.1.0.0.0 OGGCORE_OGGADP.19.1.0.0.0 PLATFORMS 190618.1416
Linux, x64, 64bit (optimized), Generic on Jun 18 2019 15:37:50
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.
```

```
GGSCI (edvmr1p0) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING

GGSCI (edvmr1p0) 2>
```



Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 10

Create the parameter file for the Replicat.

```
GGSCI (edvmr1p0) 2> edit param rkafka
```

```
GGSCI (edvmr1p0) 3> view param rkafka
```

```
REPLICAT rkafka
```

```
TARGETDB LIBFILE libggjava.so SET property=dirprm/kafka.properties
```

```
REPORTCOUNT EVERY 1 MINUTES, RATE
```

```
GROUPTRANSOPS 10000
```

```
MAP OGGSRC.* , TARGET OGGTRG.*;
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 11

Add the Replicat group.

```
GGSCI (edvmr1p0) 4> add replicat rkafka, exttrail ./dirdat/kf  
REPLICAT added.
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 12

Start the Replicat and check its status.

```
GGSCI (edvmr1p0) 1> start rkafka
```

```
Sending START request to MANAGER ...
REPLICAT RKAFKA starting
```

```
GGSCI (edvmr1p0) 2> info rkafka
```

```
REPLICAT RKAFKA      Last Started 2019-08-22 03:37      Status RUNNING
Checkpoint Lag        00:00:00 (updated 00:00:00 ago)
Process ID            24304
Log Read Checkpoint  File /u01/app/oggtrg/dirdat/kf000000000
                      First Record RBA 0
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 13

Confirm the source Manager and Extracts are running.

```
GGSCI (edvmr1p0) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING    DPUMP      00:00:00      00:00:07
EXTRACT      RUNNING    PRIEX      00:00:10      00:00:00
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 14

Generate database activity on source tables.

```
SQL> insert into customer_prod select * from customer where customer_id < 101;  
100 rows created.  
  
SQL> commit;  
  
Commit complete.
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 15

Verify the Extract captured the changes.

```
GGSCI (edvmr1p0) 1> send priex, stats total table oggsrc.customer_prod
Sending stats request to EXTRACT PRIEX ...
Start of Statistics at 2019-08-22 04:05:04.
DDL replication statistics (for all trails):
*** Total statistics since extract started      ***
    Operations                      6.00
Output to ./dirdat/in:
Extracting from OGGSRC.CUSTOMER_PROD to OGGSRC.CUSTOMER_PROD:
*** Total statistics since 2019-08-22 04:00:33 ***
    Total inserts                  100.00
    Total updates                   0.00
    Total deletes                   0.00
    Total discards                  0.00
    Total operations                100.00
End of Statistics.
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 16

Verify the Replicat delivered the changes.

```
GGSCI (edvmrlp0) 1> send rkafka stats total table oggsrc.customer_prod
```

```
Sending STATS request to REPLICAT RKAFKA ...
```

```
Start of Statistics at 2019-08-22 04:09:18.
```

```
Replicating from OGGSRC.CUSTOMER_PROD to OGGTRG.CUSTOMER_PROD:
```

```
*** Total statistics since 2019-08-22 04:00:35 ***
```

Total inserts	100.00
Total updates	0.00
Total deletes	0.00
Total upserts	0.00
Total discards	0.00
Total operations	100.00

```
End of Statistics.
```

Demo 4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 17

Verify the changes were delivered into the topic.

```
[oracle@edvmr1p0 oggtrg]$ kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic oggbgtopic --from-beginning --zookeeper localhost:2181
{"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.940000","pos":"00000000000000002372","after": {"CUSTOMER_ID":1,"FIRST_NAME":"Jaden","LAST_NAME":"Poole","EMAIL_ADDRESS":"Suspendisse@Donecconsectetuer.org","SSN":"16060407-7404","ADDRESS":"3640 Ac St.","CITY":"San Juan de Dios","ZIP_CODE":"31014","CUSTOMER_TYPE":1}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.942000","pos":"0000000000000002658","after": {"CUSTOMER_ID":2,"FIRST_NAME":"Jack","LAST_NAME":"Cox","EMAIL_ADDRESS":"dapibus.gravida@semollisdui.org","SSN":"16040918-3654","ADDRESS":"Ap #305-8153 Libero Ave","CITY":"Catas","ZIP_CODE":"40813","CUSTOMER_TYPE":4}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.942001","pos":"0000000000000002912","after": {"CUSTOMER_ID":3,"FIRST_NAME":"Anastasia","LAST_NAME":"Stewart","EMAIL_ADDRESS":"nibh.Quisque@duinecurna.org","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibas)","ZIP_CODE":"50409","CUSTOMER_TYPE":7}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.942002","pos":"0000000000000003198","after": {"CUSTOMER_ID":4,"FIRST_NAME":"Rowan","LAST_NAME":"Fleming","EMAIL_ADDRESS":"vitae.erat.Vivamus@eu.edu","SSN":"16700223-3166","ADDRESS":"7594 Adipiscing. Ave","CITY":"San Vicente","ZIP_CODE":"40102","CUSTOMER_TYPE":2}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.943000","pos":"0000000000000003448","after": {"CUSTOMER_ID":5,"FIRST_NAME":"Adena","LAST_NAME":"Williamson","EMAIL_ADDRESS":"Nunc quis@In.co.uk","SSN":"16420904-9727","ADDRESS":"711-5217 Mauris Av.","CITY":"Quesada","ZIP_CODE":"20508","CUSTOMER_TYPE":1}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.943000","pos":"0000000000000003448","after": {"CUSTOMER_ID":6,"FIRST_NAME":"Liam","LAST_NAME":"Hanson","EMAIL_ADDRESS":"ut.sed@Quisque.com","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibas)","ZIP_CODE":"50409","CUSTOMER_TYPE":7}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.943000","pos":"0000000000000003448","after": {"CUSTOMER_ID":7,"FIRST_NAME":"Audrey","LAST_NAME":"Perez","EMAIL_ADDRESS":"ut.sed@Quisque.com","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibas)","ZIP_CODE":"50409","CUSTOMER_TYPE":7}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.943000","pos":"0000000000000003448","after": {"CUSTOMER_ID":8,"FIRST_NAME":"Audrey","LAST_NAME":"Perez","EMAIL_ADDRESS":"ut.sed@Quisque.com","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibas)","ZIP_CODE":"50409","CUSTOMER_TYPE":7}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.943000","pos":"0000000000000003448","after": {"CUSTOMER_ID":9,"FIRST_NAME":"Audrey","LAST_NAME":"Perez","EMAIL_ADDRESS":"ut.sed@Quisque.com","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibas)","ZIP_CODE":"50409","CUSTOMER_TYPE":7}} {"table":"OGGTRG.CUSTOMER_PROD","op_type":"I","op_ts":"2019-08-22 04:00:31.000250","current_ts":"2019-08-22T04:00:36.943000","pos":"0000000000000003448","after": {"CUSTOMER_ID":10,"FIRST_NAME":"Audrey","LAST_NAME":"Perez","EMAIL_ADDRESS":"ut.sed@Quisque.com","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibas)","ZIP_CODE":"50409","CUSTOMER_TYPE":7}}
```

Inserts

After Images

There should be a single KafkaProducerRecord object containing 100 concatenated operations from the source transaction.

Demos: Overview

4-1: Replicating Customer Data from an Oracle RDBMS to Apache Kafka

4-2: Performing and Verifying Updates



Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 1

Verify the extracts are running.

```
GGSCI (edvmr1p0) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING    DPUMP      00:00:00      00:00:00
EXTRACT      RUNNING    PRIEX      00:00:10      00:00:03
```

```
GGSCI (edvmr1p0) 2> view param priex
```

```
Extract priex
SETENV (ORACLE_SID='orcl')
SETENV (ORACLE_HOME='/u01/app/oracle/product/12.1.0.2/db_1')
UserIdAlias oggadmin
TranlogOptions IntegratedParams (max_sga_size 256)
Extrail ./dirdat/in
LOGALLSUPCOLS
UPDATERECORDFORMAT COMPACT
GETUPDATEBEFRES
Table oggsrc.*;
```

There is no DDL parameter included in the parameter file and it is not needed to create a topic.

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 2

Update a source column.

```
SQL> update customer_prod set city='San Rafael' where customer_id=100;  
1 row updated.
```

```
SQL> commit;
```

```
Commit complete.
```

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 3

Verify the Replicat delivered the changes.

```
GGSCI (edvmr1p0) 1> send rkafka stats total table oggsrc.customer_prod
Sending STATS request to REPLICAT RKAFKA ...
Start of Statistics at 2019-08-22 04:45:09.
Replicating from OGGSRC.CUSTOMER_PROD to OGGTRG.CUSTOMER_PROD:
*** Total statistics since 2019-08-22 04:00:35 ***
      Total inserts                      100.00
      Total updates                      1.00
      Total deletes                      0.00
      Total upserts                      0.00
      Total discards                      0.00
      Total operations                   101.00
End of Statistics.
```

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 4

Verify the changes were delivered into the topic.

```
00s", "ZIP_CODE": "20002", "CUSTOMER_TYPE": 6}]} {"table": "OGGTRG.CUSTOMER_PROD", "op_type": "I", "op_ts": "2019-08-22 04:00:31.000250", "current_ts": "2019-08-22T04:00:36.962005", "pos": "00000000000000027677", "after": {"CUSTOMER_ID": 100, "FIRST_NAME": "Orli", "LAST_NAME": "Mitchell", "EMAIL_ADDRESS": "Morbi quis@Maecenas.org", "SSN": "16970620-9575", "ADDRESS": "Ap #584-1924 Maecenas Street", "CITY": "Dallas", "ZIP_CODE": "51817", "CUSTOMER_TYPE": 3}}  
{"table": "OGGTRG.CUSTOMER_PROD", "op_type": "U", "op_ts": "2019-08-22 04:40:58.000280", "current_ts": "2019-08-22T04:41:02.302000", "pos": "00000000000000027932", "before": {"CUSTOMER_ID": 100, "FIRST_NAME": "Orli", "LAST_NAME": "Mitchell", "EMAIL_ADDRESS": "Morbi quis@Maecenas.org", "SSN": "16970620-9575", "ADDRESS": "Ap #584-1924 Maecenas Street", "CITY": "Dallas", "ZIP_CODE": "51817", "CUSTOMER_TYPE": 3}, "after": {"CUSTOMER_ID": 100, "CITY": "San Rafael"}}
```

Before Image

Update

After Image

There should be a single KafkaProducerRecord object containing 1 update operation from the source transaction with before and after images.

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 5

Modify the gg.handler.kafka.mode parameter in the kafka.properties file.

```
[oracle@edvmr1p0 dirprm]$ vi kafka.properties
[oracle@edvmr1p0 dirprm]$ cat kafka.properties
gg.handlerlist=kafka
gg.handler.kafka.type=kafka
gg.handler.kafka.KafkaProducerConfigFile=kafka_producer.properties
gg.handler.kafka.topicMappingTemplate=oggbgtopic
gg.handler.kafka.format=json
gg.handler.kafka.BlockingSend=false
gg.handler.kafka.includeTokens=false
gg.handler.kafka.mode = op
goldengate.userexit.timestamp=utc
goldengate.userexit.writers=javawriter
javawriter.stats.display=TRUE
javawriter.stats.full=TRUE
gg.log=log4j
gg.log.level=INFO
gg.report.time=30sec
gg.classpath=dirprm/:/opt/kafka/libs/*
jvm.bootoptions=-Xmx512m -Xms32m -Djava.class.path=ggjava/ggjava.jar
```

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 6

Stop and re-start the Replicat.

```
GGSCI (edvmr1p0) 1> stop rkafka
```

```
Sending STOP request to REPLICAT RKAFKA ...
Request processed.
```

```
GGSCI (edvmr1p0) 2> start rkafka
```

```
Sending START request to MANAGER ...
REPLICAT RKAFKA starting
```

```
GGSCI (edvmr1p0) 3> info rkafka
```

```
REPLICAT RKAFKA      Last Started 2019-08-22 05:04      Status RUNNING
Checkpoint Lag        00:00:00 (updated 00:00:04 ago)
Process ID            26436
Log Read Checkpoint File /u01/app/oggtrg/dirdat/kf0000000000
                      2019-08-22 04:00:33.537324  RBA 28253
```

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 7

Update 4 source records.

```
SQL> update customer_prod set city='Denver' where customer_id < 5;  
4 rows updated.
```

```
SQL> commit;
```

```
Commit complete.
```

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 8

Verify the Replicat delivered the changes.

```
GGSCI (edvmr1p0) 1> send rkafka stats total table oggsrc.customer_prod
```

```
Sending STATS request to REPLICAT RKAFKA ...
```

```
Start of Statistics at 2019-08-22 05:11:55.
```

```
Replicating from OGGSRC.CUSTOMER_PROD to OGGTRG.CUSTOMER_PROD:
```

```
*** Total statistics since 2019-08-22 05:09:46 ***
```

Total inserts	0.00
Total updates	4.00
Total deletes	0.00
Total upserts	0.00
Total discards	0.00
Total operations	4.00

```
End of Statistics.
```

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 9

Verify the first update was replicated into the topic.

```
{"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 04:40:58.000280","current_ts":"2019-08-22T04:41:02.302000","pos":"000000000000000027932","before":{"CUSTOMER_ID":100,"FIRST_NAME":"Orli","LAST_NAME":"Mitchell","EMAIL_ADDRESS":"Morbi.quis@Maecenas.org","SSN":"16970620-9575","ADDRESS":"Ap #584-1924 Maecenas Street","CITY":"Cacasas","ZIP_CODE":"51817","CUSTOMER_TYPE":3},"after":{"CUSTOMER_ID":100,"CITY":"San Rafael"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.205000","pos":"000000000000000028253","before":{"CUSTOMER_ID":1,"FIRST_NAME":"Jaden","LAST_NAME":"Poole","EMAIL_ADDRESS":"Suspendisse@Donecconsectetuer.org","SSN":"16060407-7404","ADDRESS":"3640 Ac St.","CITY":"San Juan de Dios","ZIP_CODE":"31014","CUSTOMER_TYPE":1},"after":{"CUSTOMER_ID":1,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.243000","pos":"000000000000000028567","before":{"CUSTOMER_ID":2,"FIRST_NAME":"Jack","LAST_NAME":"Cox","EMAIL_ADDRESS":"dapibus.gravida@semollisdui.org","SSN":"16040918-3654","ADDRESS":"Ap #305-8153 Libero Ave","CITY":"Cacasas","ZIP_CODE":"40813","CUSTOMER_TYPE":4},"after":{"CUSTOMER_ID":2,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.244000","pos":"000000000000000028853","before":{"CUSTOMER_ID":3,"FIRST_NAME":"Anastasia","LAST_NAME":"Stewart","EMAIL_ADDRESS":"nibh.Quisque@duinecurna.org","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tiboss)","ZIP_CODE":"50409","CUSTOMER_TYPE":7},"after":{"CUSTOMER_ID":3,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.245000","pos":"000000000000000029171","before":{"CUSTOMER_ID":4,"FIRST_NAME":"Rowan","LAST_NAME":"Fleming","EMAIL_ADDRESS":"vitae.erat.Vivamus@eu.edu","SSN":"16700223-3166","ADDRESS":"7594 Adipiscing Ave","CITY":"San Vicente","ZIP_CODE":"40102","CUSTOMER_TYPE":2},"after":{"CUSTOMER_ID":4,"CITY":"Denver"}}}
```

Before Images

After Images

There should be 4 new KafkaProducerRecord objects each containing 1 update operation from the source transaction with before and after images.



Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 10

Verify the second update was replicated into the topic.

```
{"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 04:40:58.000280","current_ts":"2019-08-22T04:41:02.302000","pos":"000000000000000027932","before":{"CUSTOMER_ID":100,"FIRST_NAME":"Orli","LAST_NAME":"Mitchell","EMAIL_ADDRESS":"Morbi.quis@Maecenas.org","SSN":"16970620-9575","ADDRESS":"Ap #584-1924 Maecenas Street","CITY":"Capas","ZIP_CODE":"51817","CUSTOMER_TYPE":3},"after":{"CUSTOMER_ID":100,"CITY":"San Rafael"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.205000","pos":"000000000000000028253","before":{"CUSTOMER_ID":1,"FIRST_NAME":"Jaden","LAST_NAME":"Poole","EMAIL_ADDRESS":"Suspendisse@Donecconsectetuer.org","SSN":"16060407-7404","ADDRESS":"3640 Ac St.","CITY":"San Juan de Dios","ZIP_CODE":"31014","CUSTOMER_TYPE":1},"after":{"CUSTOMER_ID":1,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.243000","pos":"000000000000000028567","before":{"CUSTOMER_ID":2,"FIRST_NAME":"Jack","LAST_NAME":"Cox","EMAIL_ADDRESS":"dapibus.gravida@semmollisdui.org","SSN":"16040918-3654","ADDRESS":"Ap #305-8153 Libero Ave","CITY":"Capas","ZIP_CODE":"40813","CUSTOMER_TYPE":4},"after":{"CUSTOMER_ID":2,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.244000","pos":"000000000000000028853","before":{"CUSTOMER_ID":3,"FIRST_NAME":"Anastasia","LAST_NAME":"Stewart","EMAIL_ADDRESS":"nibh.Quisque@duinecurna.org","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibos)","ZIP_CODE":"50409","CUSTOMER_TYPE":7},"after":{"CUSTOMER_ID":3,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.245000","pos":"000000000000000029171","before":{"CUSTOMER_ID":4,"FIRST_NAME":"Rowan","LAST_NAME":"Fleming","EMAIL_ADDRESS":"vitae.erat.Vivamus@eu.edu","SSN":"16700223-3166","ADDRESS":"7594 Adipiscing. Ave","CITY":"San Vicente","ZIP_CODE":"40102","CUSTOMER_TYPE":2},"after":{"CUSTOMER_ID":4,"CITY":"Denver"}}
```

Before Images

After Images

There should be 4 new KafkaProducerRecord objects each containing 1 update operation from the source transaction with before and after images.



Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 11

Verify the third update was replicated into the topic.

```
{"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 04:40:58.000280","current_ts":"2019-08-22T04:41:02.302000","pos":"000000000000000027932","before":{"CUSTOMER_ID":100,"FIRST_NAME":"Orli","LAST_NAME":"Mitchell","EMAIL_ADDRESS":"Morbi.quis@Maecenas.org","SSN":"16970620-9575","ADDRESS":"Ap #584-1924 Maecenas Street","CITY":"Cattas","ZIP_CODE":"51817","CUSTOMER_TYPE":3},"after":{"CUSTOMER_ID":100,"CITY":"San Rafael"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.205000","pos":"000000000000000028253","before":{"CUSTOMER_ID":1,"FIRST_NAME":"Jaden","LAST_NAME":"Poole","EMAIL_ADDRESS":"Suspendisse@Donecconsectetuer.org","SSN":"16060407-7404","ADDRESS":"3640 Ac St.","CITY":"San Juan de Dios","ZIP_CODE":"31014","CUSTOMER_TYPE":1},"after":{"CUSTOMER_ID":1,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.243000","pos":"000000000000000028567","before":{"CUSTOMER_ID":2,"FIRST_NAME":"Jack","LAST_NAME":"Cox","EMAIL_ADDRESS":"dapibus.gravida@semmollisdui.org","SSN":"16040918-3654","ADDRESS":"Ap #305-8153 Libero Ave","CITY":"Cattas","ZIP_CODE":"40813","CUSTOMER_TYPE":4},"after":{"CUSTOMER_ID":2,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.244000","pos":"000000000000000028853","before":{"CUSTOMER_ID":3,"FIRST_NAME":"Anastasia","LAST_NAME":"Stewart","EMAIL_ADDRESS":"nibh.Quisque@duinecurna.org","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibtts)","ZIP_CODE":"50409","CUSTOMER_TYPE":7},"after":{"CUSTOMER_ID":3,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.245000","pos":"000000000000000029171","before":{"CUSTOMER_ID":4,"FIRST_NAME":"Rowan","LAST_NAME":"Fleming","EMAIL_ADDRESS":"vitae.erat.Vivamus@eu.edu","SSN":"16700223-3166","ADDRESS":"7594 Adipiscing. Ave","CITY":"San Vicente","ZIP_CODE":"40102","CUSTOMER_TYPE":2},"after":{"CUSTOMER_ID":4,"CITY":"Denver"}}
```

Before Images

After Images

There should be 4 new KafkaProducerRecord objects each containing 1 update operation from the source transaction with before and after images.

Demo 4-2: Replicating Customer Data from an Oracle RDBMS to Apache Kafka Using the Oracle GoldenGate Kafka Handler, Step 12

Verify the fourth update was replicated into the topic.

```
{"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 04:40:58.000280","current_ts":"2019-08-22T04:41:02.302000","pos":"000000000000000027932","before":{"CUSTOMER_ID":100,"FIRST_NAME":"Orli","LAST_NAME":"Mitchell","EMAIL_ADDRESS":"Morbi.quis@Maecenas.org","SSN":"16970620-9575","ADDRESS":"Ap #584-1924 Maecenas Street","CITY":"Cattas","ZIP_CODE":"51817","CUSTOMER_TYPE":3},"after":{"CUSTOMER_ID":100,"CITY":"San Rafael"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.205000","pos":"000000000000000028253","before":{"CUSTOMER_ID":1,"FIRST_NAME":"Jaden","LAST_NAME":"Poole","EMAIL_ADDRESS":"Suspendisse@Donecconsectetuer.org","SSN":"16060407-7404","ADDRESS":"3640 Ac St.","CITY":"San Juan de Dios","ZIP_CODE":"31014","CUSTOMER_TYPE":1},"after":{"CUSTOMER_ID":1,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.243000","pos":"000000000000000028567","before":{"CUSTOMER_ID":2,"FIRST_NAME":"Jack","LAST_NAME":"Cox","EMAIL_ADDRESS":"dapibus.gravida@semmollisdui.org","SSN":"16040918-3654","ADDRESS":"Ap #305-8153 Libero Ave","CITY":"Cattas","ZIP_CODE":"40813","CUSTOMER_TYPE":4},"after":{"CUSTOMER_ID":2,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.244000","pos":"000000000000000028853","before":{"CUSTOMER_ID":3,"FIRST_NAME":"Anastasia","LAST_NAME":"Stewart","EMAIL_ADDRESS":"nibh.Quisque@duinecurna.org","SSN":"16030603-2822","ADDRESS":"282-2584 Consectetuer Street","CITY":"San Juan (San Juan de Tibtts)","ZIP_CODE":"50409","CUSTOMER_TYPE":7},"after":{"CUSTOMER_ID":3,"CITY":"Denver"}}, {"table":"OGGTRG.CUSTOMER_PROD","op_type":"U","op_ts":"2019-08-22 05:09:39.000215","current_ts":"2019-08-22T05:09:46.245000","pos":"000000000000000029171","before":{"CUSTOMER_ID":4,"FIRST_NAME":"Rowan","LAST_NAME":"Fleming","EMAIL_ADDRESS":"vitae.erat.Vivamus@eu.edu","SSN":"16700223-3166","ADDRESS":"7594 Adipiscing. Ave","CITY":"San Vicente","ZIP_CODE":"40102","CUSTOMER_TYPE":2},"after":{"CUSTOMER_ID":4,"CITY":"Denver"}}
```

Before Images

After Images

There are 4 new KafkaProducerRecord objects each containing 1 update operation from the source transaction with before and after images !!!

Where can I learn more from Oracle University ?

- Oracle GoldenGate 12c: Fundamentals for Oracle (4 days)
- Oracle GoldenGate 12c: Advanced Configuration for Oracle (4 days)
- Oracle GoldenGate 12c: Troubleshooting and Tuning (4 days)
- Oracle GoldenGate 12c: Management Pack Overview (2 days)
- Oracle GoldenGate 12c: Veridata Essentials (1 day)
- Oracle GoldenGate 12c: Integrate Big Data (3 days)

Session Survey

Help us make the content even better. Please complete the session survey in the Mobile App.



Thank You

Randall Richeson

Senior Principal Instructor
Oracle University

Safe Harbor

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.





ORACLE