

Oracle® Database

Oracle GoldenGate Microservices Architecture Solutions



26ai
G49059-01
January 2026

ORACLE®

Copyright © 2024, 2026, Oracle and/or its affiliates.

Primary Author: Preeti Shukla

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

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

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

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

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

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

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

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

Contents

1 About Oracle GoldenGate Solutions

2 Flexibility Solutions in Oracle GoldenGate Microservices

3 Replicate Between Relational Databases

Data Replication Models	1
Replication Using Local Deployment	1
Replication Using a Hub	2
Replication Using a Hybrid Model	3
Solutions for Data Replication in Different Relational Database Topologies	4
Unidirectional	5
Live Reporting in a Unidirectional Replication Environment	5
Zero-Downtime System, Database, and Application Upgrades and Migrations	6
Maintain Lower-Level Environments	6
Bidirectional	7
Comparing Active-Active and Active-Passive Configurations	7
Bidirectional Replication Using Live Standby Database	9
Bidirectional Replication Using Active-Active Configuration	10
Data Distribution	11
Cascading	16
Considerations for Cascading Topologies	17
Cascading Configuration Real-World Use Cases	18
Consolidation	19
Advanced Topologies	20

4 Replicate between Data Fabrics

Solutions for Data Replication in Different Data Mesh or Data Fabric Environments	1
Data Lake Ingestion	1
Real-time Streaming Pipeline and Analytics	1
Data Streams	1

A Automation Scripts

Reporting	A-1
Business Reporting Configuration Using REST API Commands	A-1
Setup a Data Replication Environment Using REST API Commands	A-1
Check the Statistical Reports Using REST API Commands	A-6
Delete the Replication Environment Using REST API Commands	A-9
Business Reporting Configuration Using Admin Client Commands	A-14
Set Up a Data Replication Environment Using Admin Client Commands	A-14
Check the Statistical Reports Using Admin Client Commands	A-15
Delete the Replication Environment Using Admin Client Commands	A-16
Common Files and Scripts Used with Business Reporting in a Unidirectional Topology	A-17
Extract Parameter File in Unidirectional Replication	A-17
Replicat Parameter File in Unidirectional Replication	A-17
Source DML Operations	A-18
Sample Select Statements for Source and Target PDBs	A-18
Bidirectional Replication Using Active-Active Configuration	A-19
Active-Active Configuration Using REST API Commands	A-19
Create Bidirectional Configuration Using REST API Commands	A-19
Check the Statistics in a Bidirectional Environment Using REST API Commands	A-28
Remove the Bidirectional Replication Setup Using REST API Commands	A-32
Active-Active Configuration Using Admin Client Commands	A-39
Bidirectional Configuration Using the Admin Client Commands	A-39
Check the Processes in the Bidirectional Environment Using Admin Client	A-41
Remove the Bidirectional Replication Environment Using Admin Client Commands	A-42
Common Files and Scripts Used in a Bidirectional Replication	A-43
DBNORTH DML Operations	A-43
Select Statement for DBSOUTH	A-44
Source DML Operations	A-44
Select Statement for DBSOUTH PDBs	A-45
Configure a Cascading Topology Replication	A-45
Cascading Topology Data Replication using REST API Commands	A-45
Create a Cascading Data Replication Configuration Using REST API Commands	A-45
Check the Transaction Statistics for a Cascading Configuration Using REST API Commands	A-56
Delete the Cascading Environment Set Up Using REST API Commands	A-61
Cascading Topology Data Replication using Admin Client Commands	A-67
Create a Cascading Data Replication Configuration Using Admin Client Commands	A-67
Check a Cascading Environment Replication from Source to Intermediate to Target Server	A-70
Remove the Cascading Environment Set Up Using Admin Client Commands	A-71
Common Files and Scripts Used in Cascading Data Replication	A-72
Source Database DBNORTH DML Operation	A-73

Intermediate Database DBSOUTH DML Operations	A-73
Check the Status and Statistics for Processes in a Cascaded Replication Environment	A-74
Select Statement for Source to Intermediate Transactions	A-79
Select Statement for Intermediate Database DBSOUTH	A-80
Select Statement for Target Database DBWEST	A-80
Configure a Data Distribution Topology	A-80
Data Distribution Topology Data Replication using REST API Commands	A-81
Create a Broadcast Topology Based Data Replication Using REST API Commands	A-81
Check the Statistics in a Broadcast Topology Based Data Replication Using REST API Commands	A-86
Remove Broadcast Topology Setup Using REST API Commands	A-89
Data Distribution Topology Data Replication using Admin Client Commands	A-94
Create a Broadcast Topology Based Data Replication Using Admin Client Commands	A-94
Check the Statistics in a Broadcast Topology Based Data Replication Using Admin Client Commands	A-95
Remove the Broadcast Topology Setup Using Admin Client Commands	A-96
Common Files and Scripts Used in Data Distribution Topology	A-97
DBNORTH DML Operations	A-97
Data Replication in a Data Consolidation Topology Configuration	A-98
Data Replication in a Data Consolidation Topology Using REST API	A-98
Create a Consolidation Topology Based Data Replication Using REST API Commands	A-98
Check the Statistics in a Consolidation Topology Based Data Replication Using REST API Commands	A-102
Remove the Consolidation Topology Replication Setup Using REST API Commands	A-107
Data Replication in a Data Consolidation Topology Using Admin Client	A-111
Create a Consolidation Topology Based Data Replication Using Admin Client Commands	A-111
Check Processes and Statistics in a Consolidation Topology Based Data Replication Using Admin Client Commands	A-113
Remove the Consolidation Topology Based Data Replication Using Admin Client Commands	A-113
Data Streams Script	A-113
Data Streams Sample Script	A-114

Preface

The *Oracle GoldenGate Microservices Architecture Solutions* guide offers comprehensive architectural replication blueprints tailored to meet specific business requirements. These blueprints facilitate seamless data replication across both homogeneous and heterogeneous systems, enhance real-time data accessibility, and optimize workload distribution for improved efficiency.

Audience

This guide is intended for developers, database users, and administrators who are responsible for installing, deploying, and configuring Oracle GoldenGate. It is assumed that readers are familiar with web technologies, low code development, and scripting. A general understanding of Windows and UNIX platforms is required.

Documentation Accessibility

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

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Related Information

[Oracle GoldenGate Documentation](#)

[Oracle GoldenGate for Distributed Applications and Analytics](#)

[Oracle GoldenGate Studio Documentation](#)

[OCI GoldenGate](#)

[Oracle Database High Availability](#)

[Oracle GoldenGate Veridata](#)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, such as "From the File menu, select Save ." Boldface also is used for terms defined in text or in the glossary.
<i>italic</i> <i>italic</i>	Italic type indicates placeholder variables for which you supply particular values, such as in the parameter statement: TABLE <i>table_name</i> . Italic type also is used for book titles and emphasis.

Convention	Meaning
monospace MONOSPACE	Monospace type indicates code components such as user exits and scripts; the names of files and database objects; URL paths; and input and output text that appears on the screen. Uppercase monospace type is generally used to represent the names of Oracle GoldenGate parameters, commands, and user-configurable functions, as well as SQL commands and keywords.
UPPERCASE	Uppercase in the regular text font indicates the name of a process or utility unless the name is intended to be a specific case. Keywords in upper case (ADD EXTRACT, ADD EXTTRAIL, FORMAT RELEASE).
LOWERCASE	Names of processes to be written in lower case. Examples: ADD EXTRACT exte, ADD EXTTRAIL ea.
{ }	Braces within syntax enclose a set of options that are separated by pipe symbols, one of which must be selected, for example: { <i>option1</i> <i>option2</i> <i>option3</i> }.
[]	Brackets within syntax indicate an optional element. For example in this syntax, the SAVE clause is optional: CLEANUP REPLICAT <i>group_name</i> [, SAVE <i>count</i>]. Multiple options within an optional element are separated by a pipe symbol, for example: [<i>option1</i> <i>option2</i>].
Sample Locations	Compass directions such as east, west, north, south to be used for demonstrating Extract and Replicat locations. Datacenters names to use the standard similar to dc1, dc2.
Group names	Prefixes for each process, as follows: <ul style="list-style-type: none"> • Extract: ext. Usage with location: extn, where n indicates 'north' compass direction. • Replicat: rep. Usage with location: repn, where n indicates 'north' compass direction. • Distribution Path: dp. Usage with location: dpn, where n indicates 'north' compass direction. • Checkpoint table: ggs_checkpointtable • Trail file names: e or d depending on whether the trail file is for the Extract or distribution path. Suffix derived in alphabetical order. Usage for an Extract trail file: ea, eb, ec. • Trail file subdirectory: The name will use compass directions to refer to the trail subdirectories. Example for trail subdirectory name would be / east, /west, /north, /south.

1

About Oracle GoldenGate Solutions

Oracle GoldenGate provides advanced level capabilities for unidirectional, bidirectional, and multi-directional replication, live reporting, zero downtime migration (ZDM), flexibility, real-time data warehousing, load balancing, and many other business solutions.

Oracle GoldenGate Microservices Architecture supports multiple methods to monitor and administer Oracle GoldenGate. These methods include the web interface for a graphical interaction with Oracle GoldenGate, the Admin Client for a command line interface, and REST APIs for building your own user interface or automation of Oracle GoldenGate.

You can use Oracle GoldenGate for replication between relational databases and non-relational databases. Oracle GoldenGate also provides extended capabilities to apply different data management approaches such as Centralized data management (implemented using Data Mesh) and Decentralized data management (implemented using Data Fabrics).

This guide describes the implementation of these solutions using automation scripts for REST API commands using cURL and Admin Client scripts using OBEY commands, to set up various business scenarios and test Oracle GoldenGate Microservices features.

2

Flexibility Solutions in Oracle GoldenGate Microservices

Subset of Data

By replicating only relevant subsets of data, filtering can enhance replication speed and decrease the load and optimize performance on both source and target databases. The target table can be a subset of the source table, meaning that only specific rows or columns are replicated based on certain criteria.

- **Reducing Data Duplication / Data Reduction:** Filtering and mapping can help prevent unnecessary duplication of data, maintaining cleaner datasets in the target environment. This is also minimizing storage costs and improves replication performance.
- **Data Compliance and Security:** Filtering can help ensure that sensitive or personally identifiable information is not replicated to less secure environments, helping organizations meet compliance regulations.

Schema Adaptation

Mapping allows for changes in data structure, enabling the target database to have a different schema than the source. This is particularly useful when integrating with legacy systems or when the target system requires a different format.

- **Schema Compatibility:** The target table should have a compatible schema with the source table to ensure that the data can be properly replicated. This includes data types and constraints.
- **Indexes and Constraints:** While the target table can have different indexes or constraints, it's crucial to consider performance and data integrity when designing the target schema.
- **Partitioning:** In some cases, the target may use a different partitioning scheme, which can help optimize performance for specific queries.

Data Enrichment

Transformations can enhance the data during replication by adding calculated fields or converting data types, ensuring that the data is more useful for the target application.

Consistency with Business Logic

Transformations can enforce business rules and logic, ensuring that the replicated data aligns with the operational requirements of the target system.

Improved Maintenance and Management

By tailoring the data through filtering and transformations, the target database can be optimized for specific use cases, leading to easier maintenance and management over time. Incorporating filtering, mapping, and transformations allows organizations to tailor the replication process to their specific needs, enhancing efficiency, security, and data usability.

Facilitating Analytics

Transforming data into a more analyzable format can enhance reporting and analytics capabilities in the target database, making it easier for users to derive insights.

Multi-System Integration

When replicating data to multiple systems, mapping and transformations can facilitate compatibility between different databases, allowing for smoother integration.

3

Replicate Between Relational Databases

Learn about the commonly used Oracle GoldenGate Microservices Architecture solutions when replicating between two or more relational databases.

Oracle GoldenGate supports Oracle and non-Oracle relational databases for replication. You can view a list of the supported databases for the current release from the certification matrix:

<https://www.oracle.com/integration/goldengate/certifications/>.

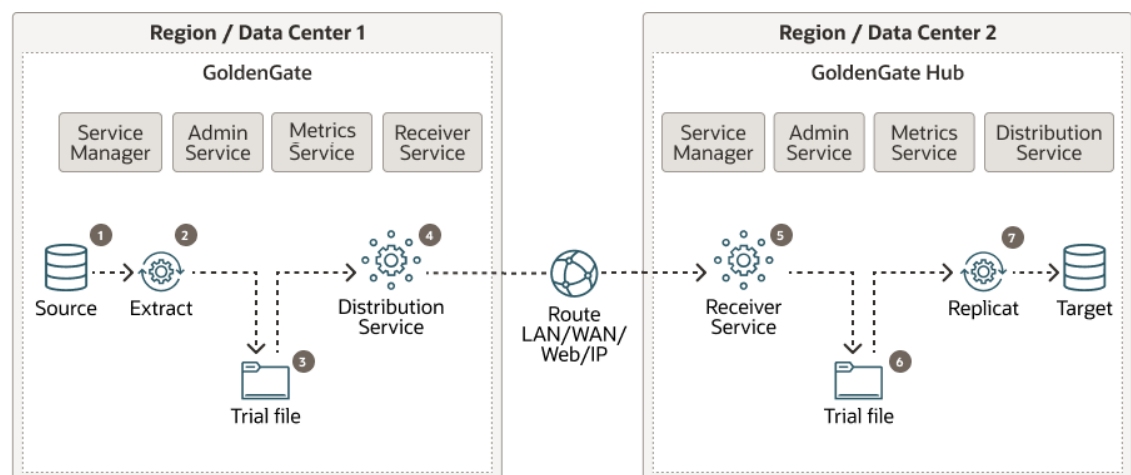
Data Replication Models

Oracle GoldenGate can be installed on the local server as the data source or target, or on a remote server as source or target. Oracle GoldenGate processes can connect to the database by configuring a network connection from Oracle GoldenGate to the database. For example, in Oracle or non-Oracle database server, the connection from the Extract, Replicat, and Microservices are made using Database Network, and can run on a separate server from the database server.

This section covers different aspects of setting up a local deployment when installing Oracle GoldenGate GoldenGate or remote deployment, when installing Oracle GoldenGate on a hub.

Replication Using Local Deployment

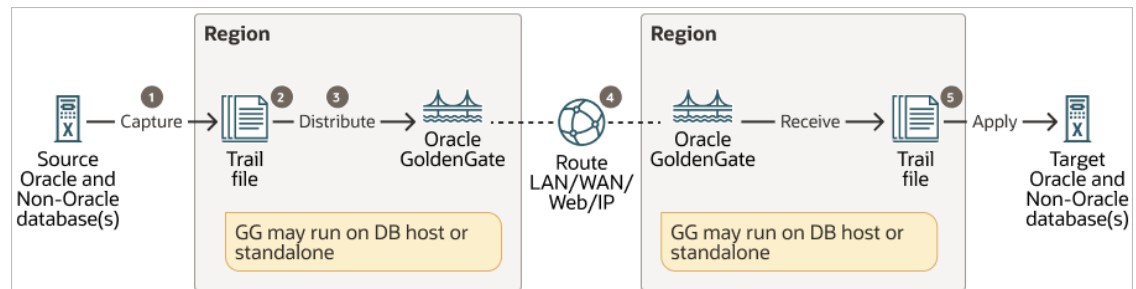
When replicating using a local deployment, the Oracle GoldenGate instance runs on the database server using a local connection to the database, as shown in the following diagram.



In this diagram, two Oracle GoldenGate deployments are set up on two different database servers, which are connected over the network. Consider, Data Center 1 as the source and Data Center 2 as the target. Both the deployment are local to their respective database server.

However, the databases reside in different regions or servers and use a Distribution Service / Receiver Service pair to send the trail files across the network.

The following diagram displays the workflow of an Oracle GoldenGate deployment in a database server within a local hub.



This diagram is described using the following points:

- 1: Capture of committed transactions that can be filtered as they occur by reading the transaction keys)
- 2: Trail is sent using canonical, secure and wire protocol for data transaction and events.
- 3: Distribution/Receiver Services send and receive data across the service mesh.
- 4: Route transactions are compressed, encrypted for routing to targets
- 5: Delivery applies data with transaction integrity

Replication Using a Hub

When replicating using a hub configuration, the Oracle GoldenGate instance runs on an individual Server, Instance (like a compute instance), or Host and connects remotely to the database. This case is used mostly when you want to limit the overhead of Oracle GoldenGate on the source or target databases, or the database environment does not allow the software to be installed locally on the database server.

Most remote Extracts, except for SQL Server, can tolerate up to 80ms round trip ping time between the source database and the server where the Extract process is running. This provides a lot of flexibility in deciding where the server running the Extract process can be physically located. This feature enables capturing remotely in multiple scenarios, especially for cloud databases. For example, it allows capturing remotely where the Extract process is running on a compute node in the same region, or in an on-premises data center, or in another region or cloud provider providing the network ingress and the egress is configured to allow for the client connection.

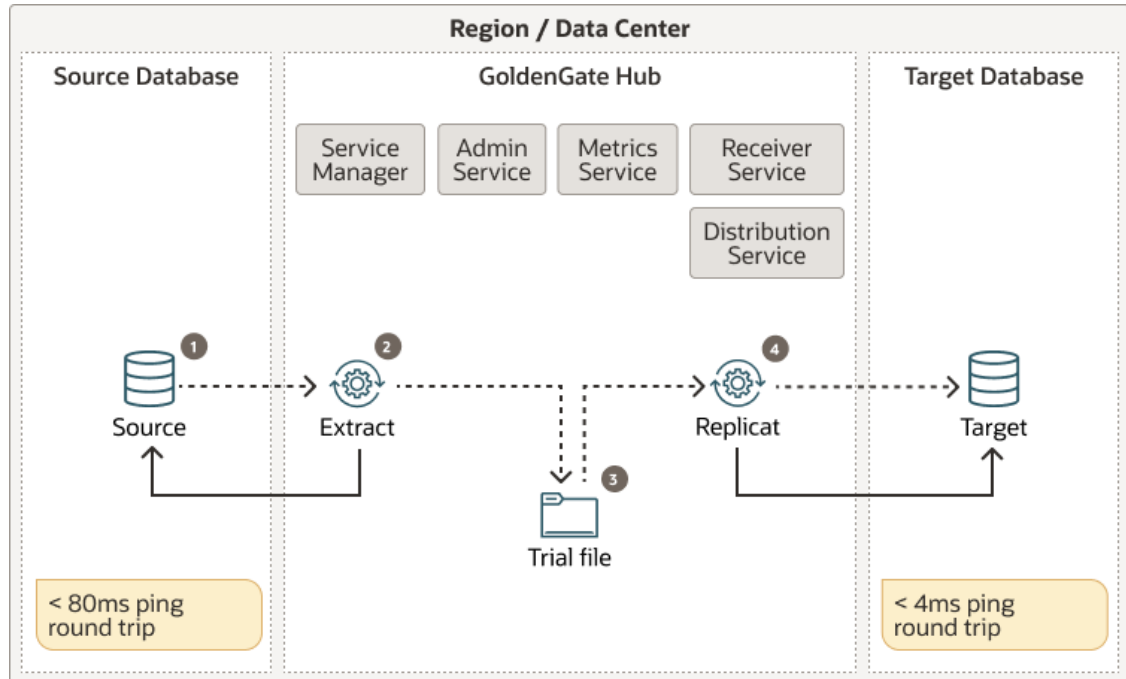
For best performance, Oracle recommends running the Extract process as close to the source database as possible.

Remote Replicats are highly dependent on low network latency because each network operation requires an acknowledgement before the next can be issued. Oracle recommends that the round-trip ping time between the server running the Replicat process and the target database be as low as possible, and typically any values higher than 4 ms will increase the Replicat lag.

In hub models, both the Extract and the Replicat for the same replication stream are deployed on the same server, so, there is no need for a Distribution Path. The Extract writes to a local trail file, and the Replicat reads from that same trail file. However, in some topologies, a hub

might be deployed close to the source database, with an additional hub located near the target database. In these cases, changes are routed through the Distribution Path from the source hub to the target hub.

In the following diagram of a hub deployment, the Extract and Replicat processes operate from the same Oracle GoldenGate server and connect remotely to the source database.

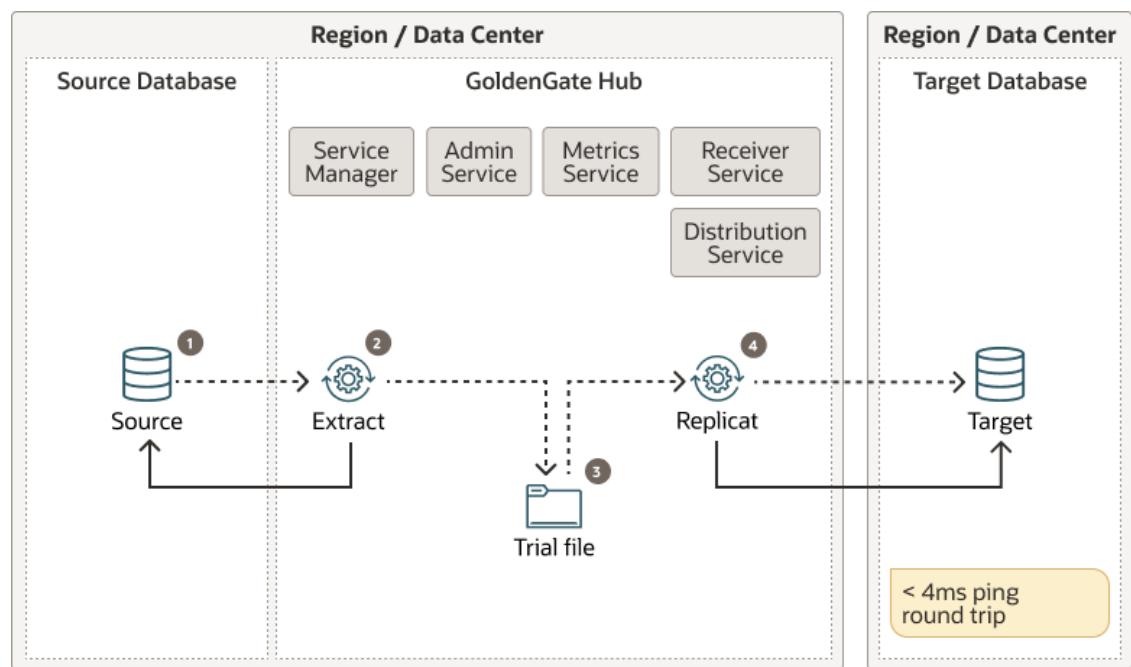


This configuration can have multiple Oracle GoldenGate installations with multiple deployments managing a fleet of replication streams with similar or different topologies and use cases.

Replication Using a Hybrid Model

Replication in a hybrid deployments with Oracle GoldenGate can be done by mixing local and hub deployments together and connect different deployments through Distribution Paths.

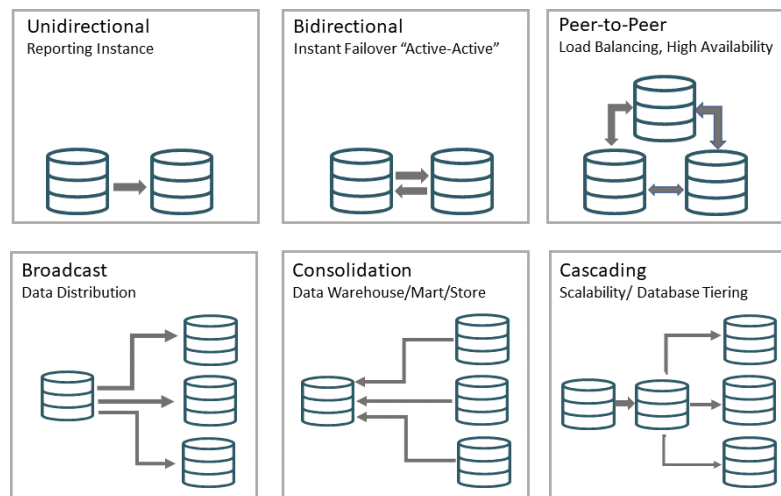
In the following hybrid deployment, Oracle GoldenGate software is installed locally on the source database and connects remotely to the target.



Solutions for Data Replication in Different Relational Database Topologies

Learn about different Oracle GoldenGate topologies configured for different databases.

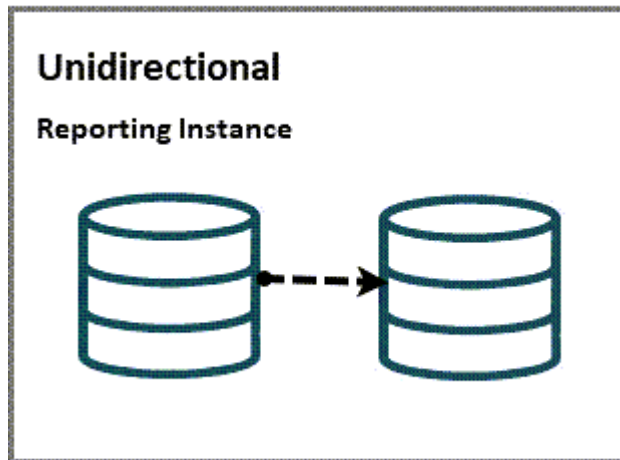
Each topology may be suited to particular use cases, although some use cases may be applicable across multiple topologies. Each topology provides a blueprint framework for configuration and is used in real-world environments with variations and combinations of these topologies. With Oracle GoldenGate's flexibility, the configuration possibilities are virtually limitless.



For full information about processing methodology, supported topologies and functionality, and configuration requirements, see Prepare section in *Oracle GoldenGate Microservices Documentation*.

Unidirectional

The most basic Oracle GoldenGate configuration is a one-to-one configuration that replicates in one direction: from a source database to a target database that is used only for data retrieval purposes such as reporting and analysis.



Oracle GoldenGate supports like-to-like or heterogeneous transfer of data, with capabilities for filtering and conversion on either system in the configuration (support varies by database platform). Oracle GoldenGate supports different reporting topologies that enable you to custom-configure the processes based on your requirements for scalability, availability, and performance. This section contains things to take into consideration when choosing a reporting configuration.

See the [Reporting](#) sample script for this scenario to test the reporting and analysis feature of Oracle GoldenGate.

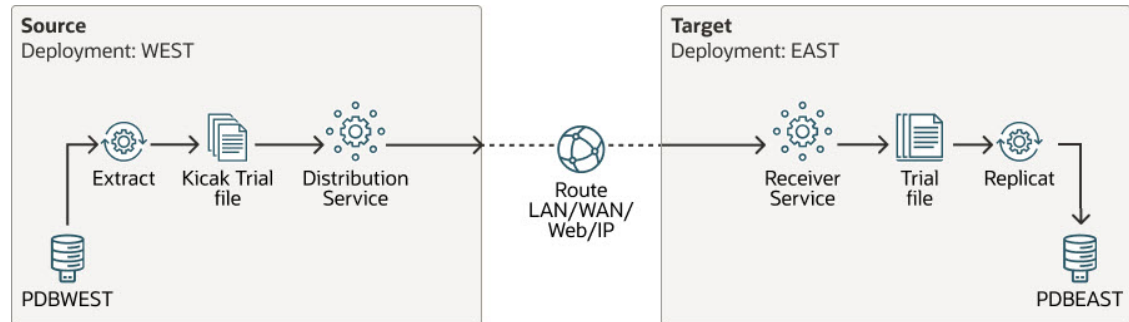
Live Reporting in a Unidirectional Replication Environment

Oracle GoldenGate supports homogenous and heterogeneous transfer of data, with capabilities for transformation, filtration, and conversion on either or both systems in the configuration (support varies by database platform). Oracle GoldenGate supports different reporting topologies that enable you to customize the configuration of processes, based on your requirements for scalability, availability, and performance.

Here are some guidelines for choosing a reporting configuration:

- It is recommended that you use a secondary database to run reports because it reduces the load on the application database, allowing more resources to be available on the replicated database. Another advantage of using Oracle GoldenGate for offloading reporting to a secondary database is that the secondary database can be optimized specifically for those reports.
- Create additional indexes, use materialized views to replace common joins, use modified storage parameters, or other database tools to ensure the reports run at an optimal speed.

- Oracle GoldenGate transformation routines can be used to flatten or modify the structure of the data to improve the performance of the reporting server. This becomes popular in a data fabric architecture, which allows the consumer of the real-time data in many open-source Data Warehouse solutions although the original transactional data only exists in traditional/legacy databases. See [Consolidation](#) to know more about using additional database for generating reports.



See the [Reporting](#) sample script for this scenario to test the reporting and analysis feature of Oracle GoldenGate.

Zero-Downtime System, Database, and Application Upgrades and Migrations

Zero Downtime operations are used to eliminate downtime for upgrades and migrations and fallback strategies. This includes migration of the database from one database technology to another, for example migrating from PostgreSQL to Oracle, or migration of a database into a cloud environment or between cloud vendors, or migration from one operating system to another. The same replication pattern can also be used to eliminate or minimize the outage for upgrades, including application upgrades where the database structure needs to change, or database and operating system upgrades and patches.

Oracle GoldenGate's adaptability and flexibility allows it to replicate between different databases, database versions, operating systems, and even between tables with different structures. While many zero downtime migrations and upgrades using Oracle GoldenGate can be configured unidirectionally, there are some situations where it is configured bi-directionally. Oracle recommends setting up fallback replication stream from the new environment to the old environment to prevent data loss. If there is an issue with the new environment, then the database and Oracle GoldenGate administrators can switch users back to the old environment without any loss of data. This is accomplished by configuring a reverse replication stream from the new environment back to the old database as covered in [Bidirectional Replication Using Live Standby Database](#).

See Best Practices for Logical Database Migration.

Also see [Learn About Migrating with Oracle GoldenGate](#) to know more.

Maintain Lower-Level Environments

Oracle GoldenGate can be used to populate lower-level environments, like QA, development, pilot, and user acceptance databases. Depending on the business needs, these environments are created as full copies of production, for example, for performance testing or pilot environments, while others will be a fraction of the data, such as for development purposes.

If only a subset of data is needed, then the Oracle GoldenGate administrator can configure `FILTER` conditions in the `TABLE` statements in the Extract process. The administrator must pay attention to lookup tables, which are typically not filtered, and normal user tables, which are filtered.

In applications with foreign keys, this can be easier compared to applications without foreign keys. If the application does not contain foreign keys, then you will need to work with the application team to determine the appropriate filters to set for replicating fractional data.

Additionally, there may be a need to obfuscate or mask certain columns. This can be done using Oracle GoldenGate's built in transformation routines, including `SQLEXEC` / `DBFUNCTION` calls, or by triggers on the target tables.

Bidirectional

Bidirectional replication can be done using an active-passive configuration where a Live Standby database is used or using an active-active configuration where the source and target databases are sending and receiving data actively.

Comparing Active-Active and Active-Passive Configurations

A common use case for Oracle GoldenGate is High Availability to reduce the downtime that occurs during planned or unplanned outages.

When implementing Oracle GoldenGate for this use case, you can configure active-active replication, where two or more databases are configured for read and write traffic from the application, or configure active-passive (also called Live Standby), where read and write traffic for the application should only connect to a single database at a time. This also ties in closely with Oracle's Maximum Availability Architecture patterns of Gold and Platinum.

Using Oracle GoldenGate this way can reduce the Recovery Time Objective to zero for nearly any kind of outage.

Use the following guidelines to decide which method you should use for your application.

Both methods are configured similar to a unidirectional topology, where a replication stream is configured from a source to a target. However, the administrator will also configure replication from the target back to the source. This means that loop detection needs to be considered for any Extracts and Replicats involved in bidirectional replication. To learn about loop detection, see [Enabling Bi-Directional Loop Detection](#). It also means that any active-passive or active-active topology can only be configured when the databases are certified for both capture and delivery. See the [certification matrix](#) to learn about the supported operating system and databases.

Active-Active High Availability Purposes

In an active-active environment you can configure bidirectional replication and the application can connect to any database.

- During unplanned outages, users can be immediately connected to another database in the active-active environment.
- For planned outages, you can slowly move users (or drain connections) over to another database prior to taking the server offline for a better user experience.

In both cases the Recovery Time Objective (RTO) is zero because the other database is already open for both read and write activity.

Oracle GoldenGate replication is asynchronous, so there could be a situation where two transactions modify the same row at the same time on different databases. When the system attempts to replicate these changes to the other server, it can create a conflict. To resolve these conflicts, administrators need to be configure Oracle GoldenGate using the `COMPARECOLS / RESOLVECONFLICT (CC/RC)` options or using the Automatic Conflict Detection and Resolution (ACDR) feature, which is only available in Oracle database. Both methods have some requirements and recommendations to ensure smooth operation and eventual consistency of your data.

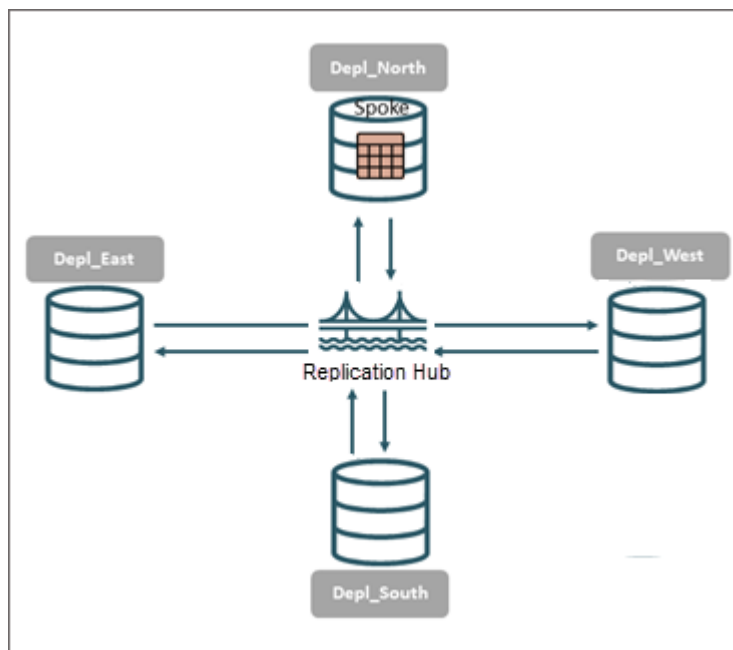
Oracle GoldenGate can support multiple active-active topologies including Data Mesh and Hub and Spoke models.

In a data mesh model, all databases are connected to each other. A mesh style works well when you have just a few databases, usually four or less.

GoldenGate Stream Analytics implements a data mesh platform, allowing you to visually construct data pipelines that perform continuous ETL or advanced analytics on any data that is moving in realtime windows that can be as small as 1 milliseconds. Using Stream Analytics, you can easily ingest data events from all database using Oracle GoldenGate.

See [About Stream Analytics](#) to know more.

In a hub and spoke model, each read-write database is a spoke, and sends data to the hub, the hub then sends data to all the other spokes. The hub-and-spoke model works better for large number of deployments, but due to the extra hop at the hub, the latency is higher which can cause additional conflicts.



For more information about configuring active-active replication with Oracle GoldenGate, refer to this technical paper: <https://www.oracle.com/a/ocom/docs/ogg-best-practices-active-active.pdf>.

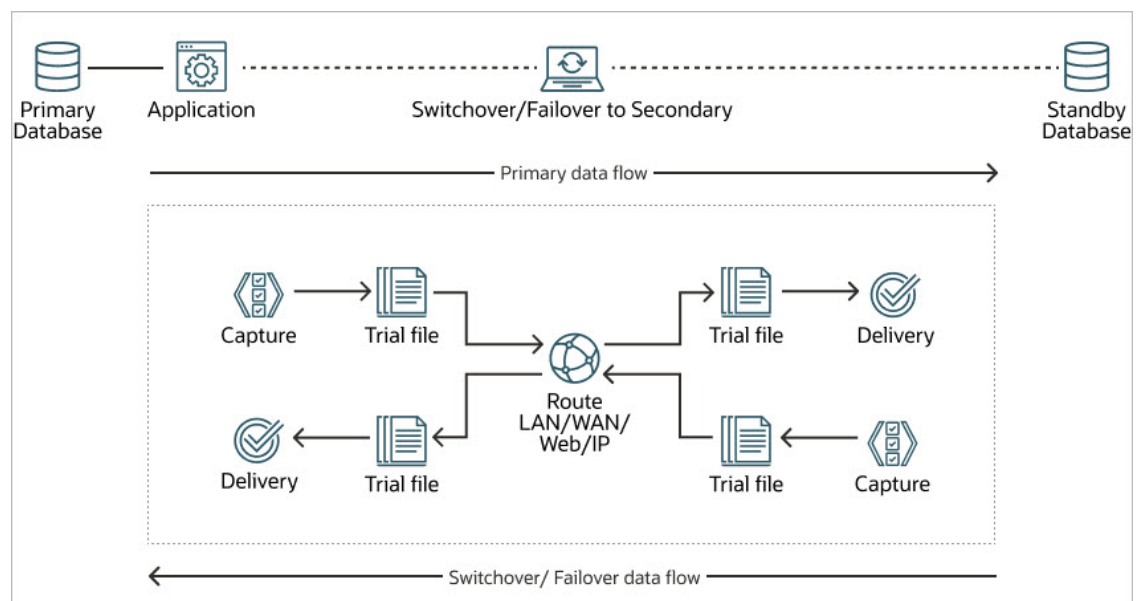
Active-Passive for High Availability Purposes

In an active-passive environment, a bidirectional replication is configured but the application is only connected to one server at a time. In the event of an outage, you cannot connect to the other database until the Oracle GoldenGate Replicat has applied the last transaction, which means the RTO will be equal to the Oracle GoldenGate lag. Typical lag for Oracle GoldenGate is about 4-7 seconds, so this can achieve very low RTO values and because write activity will only happen on one database at a time, there is no need to worry about data collisions. Active-passive doesn't have the application and schema restrictions that exist in an active-active application, so it can be used on third party applications like Siebel and PeopleSoft. A typical active-passive implementation is one passive database for each active database.

If your application can tolerate a few seconds of downtime (RTO of about 4-6 seconds) then it is recommended to use the active-passive architecture. It's easier to implement and maintain, has fewer application and table dependencies, and will work for package applications. However, if you truly need an RTO of zero, then active-active should be the architecture.

Bidirectional Replication Using Live Standby Database

Oracle GoldenGate supports an active-passive bi-directional configuration, where Oracle GoldenGate replicates data from an active primary database to a full replica database on a live standby system that is ready for failover during planned and unplanned outages.



In this configuration, there is an inactive Oracle GoldenGate Extract group and an inactive Distribution Path on the live standby system. Both these groups remain stopped until just before user applications are switched to the live standby system in a switchover or failover. When user activity moves to the standby, those groups begin capturing transactions to a local trail, where the data is stored on disk until the primary database can be used again.

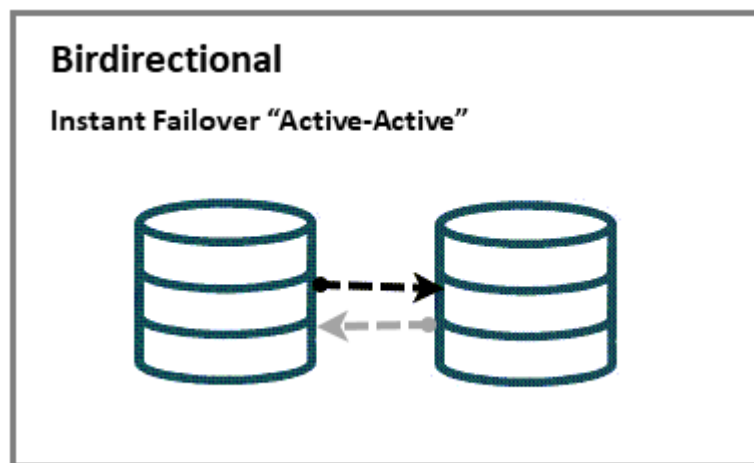
In case of a failure of the primary system, the Oracle GoldenGate Microservices and Replicat processes work in conjunction with a database instantiation taken from the standby to restore parity between the two systems after the primary system is recovered. At the appropriate time, users are moved back to the primary system, and Oracle GoldenGate is configured in ready mode again, in preparation for future failovers.

Advantages of Using Live Standby Database

- Live Standby is used for an immediate fail-over solution that can later re-synchronize with your primary source.
- It helps eliminate downtime and data loss due to unplanned outages and enables continuous availability, with no geographic distance constraints. Oracle GoldenGate moves changed data from primary database to one or more live standby databases so that the applications have a reliable failover system with up-to-date data that they can immediately switchover to. There is no database recovery process required because the live standby database is already open for read and write activity. Oracle GoldenGate will isolate corrupt data during the replication process to make sure the live standby database is reliable and available when it is needed.
- Oracle GoldenGate allows the standby database to be open and accessed by users, so it does not have to sit idle and can be used for reporting or testing. This feature directly boosts the ROI.

Bidirectional Replication Using Active-Active Configuration

Oracle GoldenGate enables bidirectional data movement between two or more databases that actively support an application, with no geographic distance constraints.



The active-active solution allows data updates and changes ("write" activity) to occur on two or more active databases supporting live applications. Oracle GoldenGate synchronizes the two active databases by replicating the data between each at a logical level and allows load distribution to improve system performance. In the case of an outage of one system, there is no downtime for the end user because the other active system continues with operations.

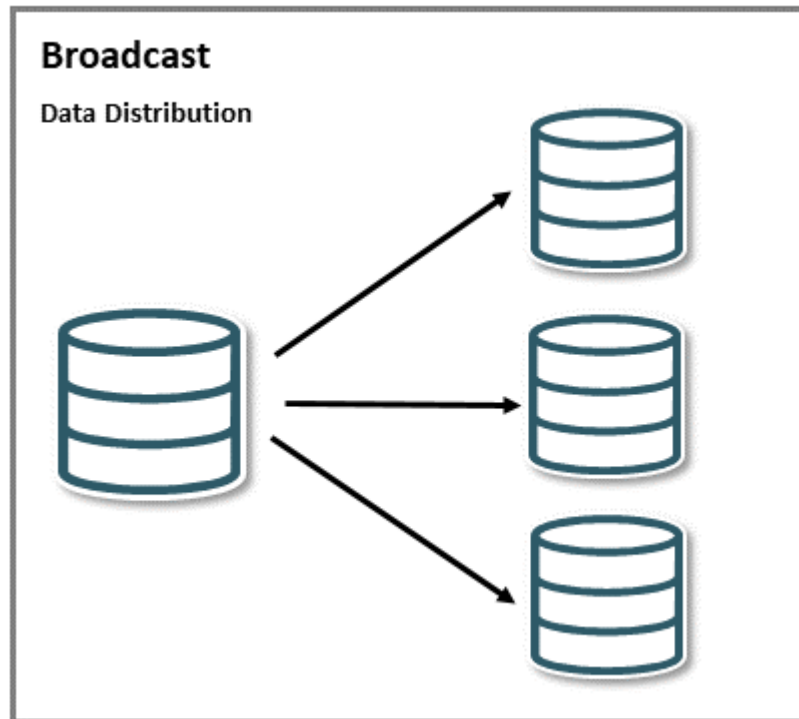
Considering that Oracle GoldenGate is an asynchronous solution, conflict management is required to ensure data accuracy in the event that the same row is changed in two or more databases at (or about) the same time. Oracle GoldenGate provides capabilities to detect and resolve conflicts. A variety of active-active scenarios can be supported – depending on the desired implementations. Oracle GoldenGate provides these active-active solutions for both High Availability as well as Zero-Downtime upgrades and migration projects.

For a sample automation script to set up an active active data replication, see [Bidirectional Replication Using Active-Active Configuration](#).

Also see [Design Different Topologies with Oracle GoldenGate 23ai](#) to set up a test environment and use these automation scripts.

Data Distribution

A data distribution configuration is a one-to-many configuration. Oracle GoldenGate supports synchronization of a source database to any number of target systems. Oracle GoldenGate supports like-to-like or heterogeneous transfer of data, with capabilities for filtering and conversion on any system in the configuration although support varies by database platform.

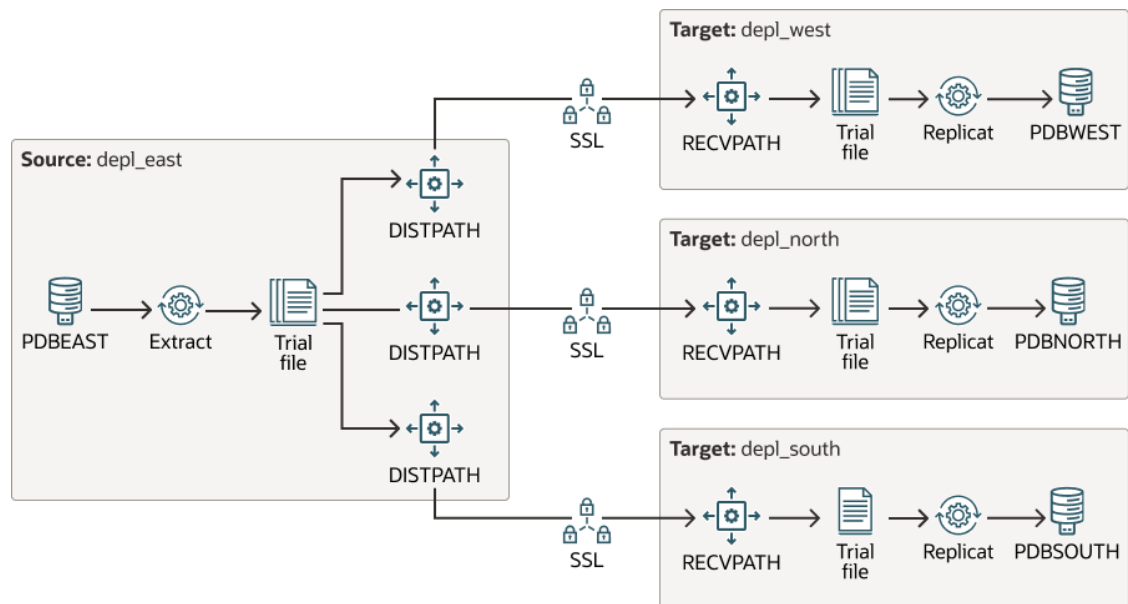


Broadcast topologies can include a mix of unidirectional use cases, including offloading of reporting, replication to a data warehouse, or sending data to a lower level environment. In most cases, administrators configure a single Extract process that writes to multiple trail files. Each trail file is used for a different target database. This reduces the overhead on the source database, and allows each target platform to receive just the data that it requires.

In the following use case, a trail file is replicated across different destinations with the following conditions:

- One trail file is used to replicate across multiple destinations.
- Multiple Distribution Paths are configured to send the trail data with filtered options to specify which tables will be replicated on to a particular destination.

The following diagram displays this configuration where the source deployment, `depl_east` is sending trail data to three different deployments using `DISTPATHS`:



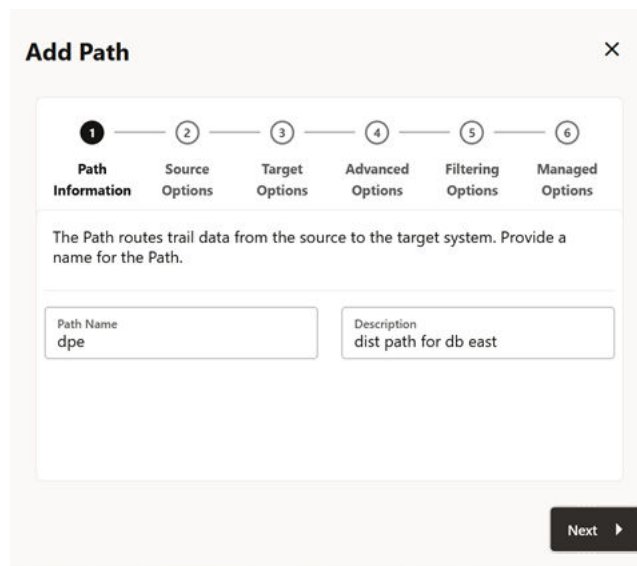
Here's the Extract parameter file:

```

EXTRACT exte
USERIDALIAS ggeast
STATOPTIONS REPORTFETCH
EXTTRAIL ea
TRANLOGOPTIONS INTEGRATEDPARAMS (max_sga_size 512)
LOGALLSUPCOLS
UPDATERECORDFORMAT COMPACT
TABLE hr.EMPLOYEES;
TABLE hr.REGION;
  
```

For this Extract, you can create multiple DISTPATHS to send trail data to different locations. You can also specify the tables that would be replicated for a specific destination, while configuring the Filtering Options for the DISTPATH. The following screens shows the creation of a DISTPATH where the target host is WEST.

1. The following image displays adding a DISTPATH, dpe for the **depl_east** deployment.



Add Path [X]

1 — 2 — 3 — 4 — 5 — 6

Path Information Source Options Target Options Advanced Options Filtering Options Managed Options

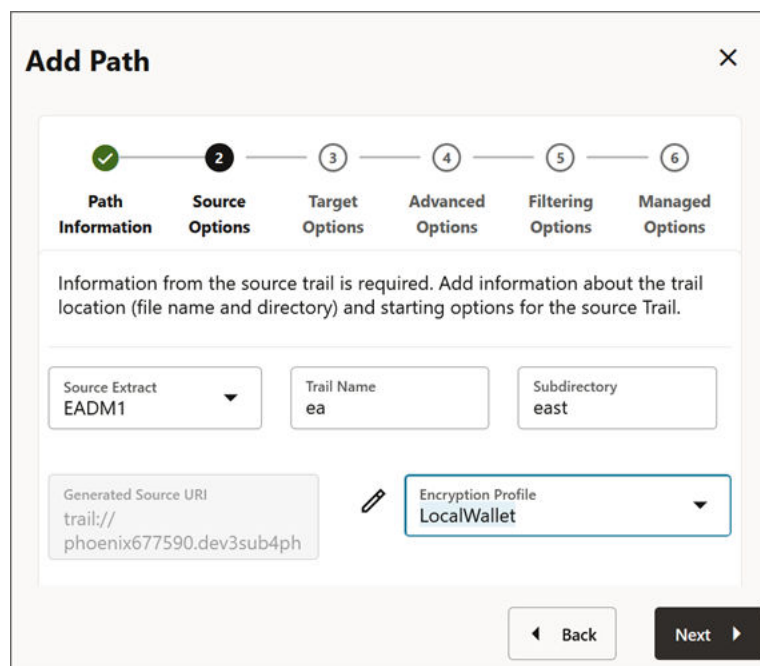
The Path routes trail data from the source to the target system. Provide a name for the Path.

Path Name: dpe

Description: dist path for db east

Next [▶]

- The following screens displays the options to set up on the source side for Extract, trail, source URI and other options. In this step, the source Extract trail file is configured.



Add Path [X]

1 — 2 — 3 — 4 — 5 — 6

Source Options Path Information Target Options Advanced Options Filtering Options Managed Options

Information from the source trail is required. Add information about the trail location (file name and directory) and starting options for the source Trail.

Source Extract: EADM1

Trail Name: ea

Subdirectory: east

Generated Source URI: trail:// phoenix677590.dev3sub4ph

Encryption Profile: LocalWallet

Back [◀] Next [▶]

- The following screen shows the configuration for the target database. At this stage, you can specify the destination where you want the trail file replicated. When you create another DISTPATH, you can provide a different target URI that would allow replication to that destination.

Add Path ✕

✓ ✓ **3** 4 5 6
Path Information **Source Options** **Target Options** **Advanced Options** **Filtering Options** **Managed Options**

Different protocols (WSS, WS, and OGG) are available for the Path. Specify the desired protocol and add the required source details.

Target Protocol
WSS

Reverse proxy enabled
☐

Target Host
localhost

Port Number
1521

Trail Name
ea

Subdirectory

Trail Size (MB)
500

Target E...
NONE

Change Encryption
☐

Generated Target URI
wss://localhost:1521/services/v2/targets?trail=ea

Target Authentication Method
UserID Alias

Domain
Network

Alias
Required

Target Type
GGFormat

◀ Back

Next ▶

- The following step allows you to set up the filtering rules for replication. So, you can specify the filtering rules to select tables, columns, values that you need replicated for each target URI.

Add Path ✕

✓

✓

✓

✓

5

6

Path Information**Source Options****Target Options****Advanced Options****Filtering Options****Managed Options**

Optionally add filtering rules (inclusion and exclusion rules) based on names, types, or column values.

Rule Name
scott.*

Rule Action
☐ Exclude ☒ Include

Filter Type
Object Type ▼

Object Types
DML, DDL 2 ▼

Negate
☐

Add

◀ Back

Next ▶

5. The following screen allows you to set up autostart/autorestart options and create the DISTPATH.

Add Path

✓

✓

✓

✓

✓

6

Path Information
Source Options
Target Options
Advanced Options
Filtering Options
Managed Options

Additional AutoRestart Options can be added to the DistPath. Adjust the AutoRestart Retries and Delays if needed.

Critical
☐

Auto Restart
☒

Autorestart Retries
10

Autorestart Delay
2 Minutes

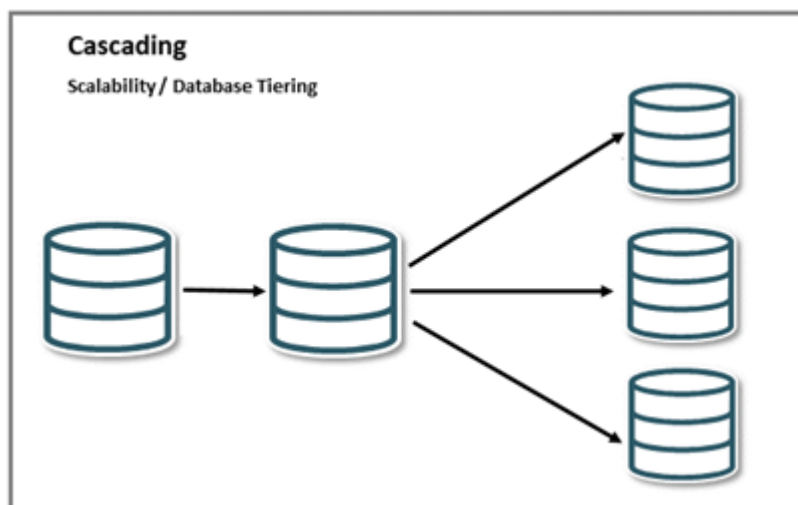
Back
Create Path
Create Path and Run

You can create other DISTPATHS in a similar manner to add multiple destinations for replicating data from the same source trail.

Cascading

Oracle GoldenGate supports cascading synchronization, where Oracle GoldenGate propagates data changes from the source database to a second database, and then on to a third database or multiple target databases.

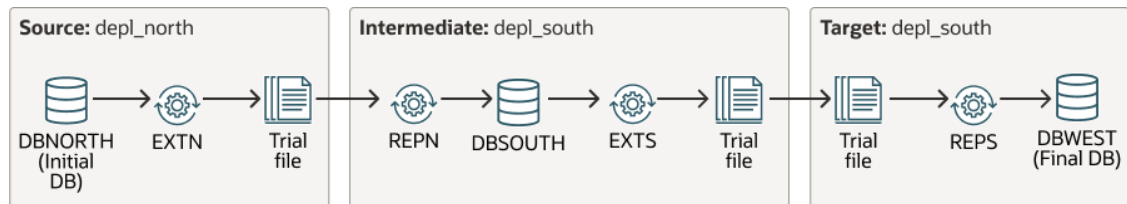
See [Configure a Cascading Topology Replication](#) for sample automation scripts to run a Cascading data replication scenario.



The workflow in this type of configuration is as follows:

1. An Extract on the initial source writes captured data to a local trail, and a DISTPATH sends the data to a remote trail on the transitional system in the cascade.
2. On the second system, Replicat applies the data to the local database.
3. Another Extract on the second system captures the data from the local database and writes it to a local trail.
4. A DISTPATH sends the data to a remote trail on the final system in the cascade, where it is applied to the local database by another Replicat.

This workflow is depicted in the following diagram:



A cascading configuration can be used for any of the following real-world scenarios:

Scalable Data Replication Configuration

To scale the data replication framework on to multiple deployments, when replicating from one source database on to secondary and then eventually a target database.

Data Tiering

To create multiple tiers for data replication set up, you can build a cascaded environment over local and remote hosts.

Extended environments

To configure extended environments where original replication is used for additional databases and applications.

Connecting between disconnected deployments

To allow connectivity between one or more target systems, which do not have a direct connection to the source, but the secondary system can connect in both directions.

Limit the network activities from the source system

To limit the network activities from the source system to secondary and target systems.

Replicating across geographically distant regions

To send data to two or more servers that are geographically far apart. For example, sending data from Chicago to Los Angeles and then from Los Angeles to servers throughout China would be a use case for setting up a cascading topology.

Considerations for Cascading Topologies

This configuration can be used to perform data filtering and conversion if the character sets on all systems are identical. If character sets differ, conversions between character sets cannot be performed, and you must configure Replicat to perform the conversion and transformation on the target.

- In Oracle GoldenGate environments earlier than 23ai, the parameters `IGNOREAPPLOPS` and `GETREPLICATES` were used to setup a cascaded replication environment. On the second system, you needed to configure the Extract group to capture Replicat activity and to ignore local business application activity.
- With Oracle GoldenGate 23ai, the parameters `IGNOREAPPLOPS` and `GETREPLICATES` are desupported and you use tagging to streamline replication changes. In a default configuration, Replicat tags changes with a '00' tag into the transitional database. The Extract process on the transitional database captures all DML changes, by default. However, for DDL operations, DDL tagging needs to be explicitly added if it needs to be replicated to the final target database. This is done by using the `DDLOPTIONS INCLUDETAG 00` parameter setting.

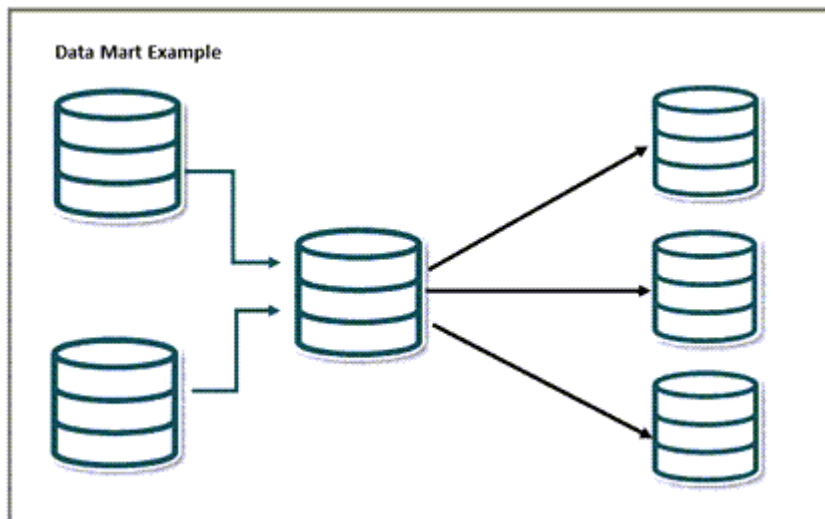
See the [Configure a Cascading Topology Replication](#) to learn about setting up replication and checking the replication statistics using the Admin Client commands and the REST API client.

Cascading Configuration Real-World Use Cases

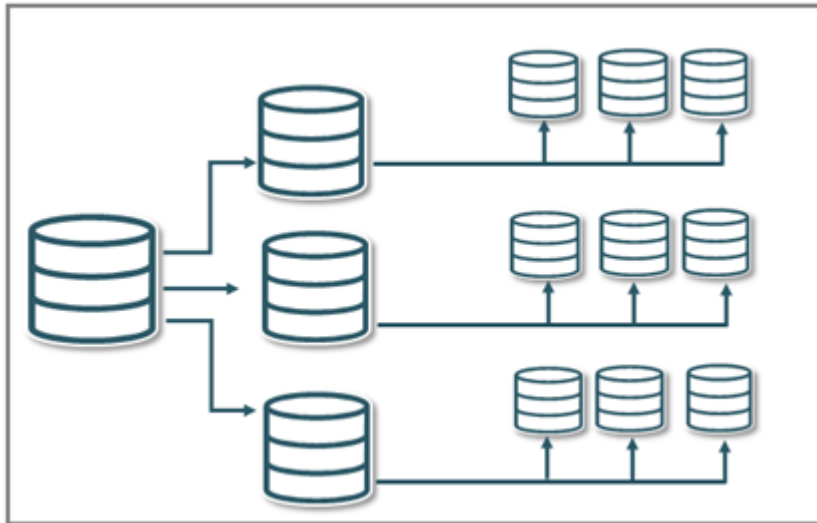
In a transitional database, additional filtering and transformation occurs before changes are replicated to the final dataset. This allows you to replicate changes from a source database to an transitional database, which in turn, is a source database to another end-target database.

You can set up even more complex replication topologies, as shown in the following examples.

The following image shows a **DataMart** configuration where the Data Warehouse is fed by multiple different sources and the data warehouse itself replicates data to different data marts.

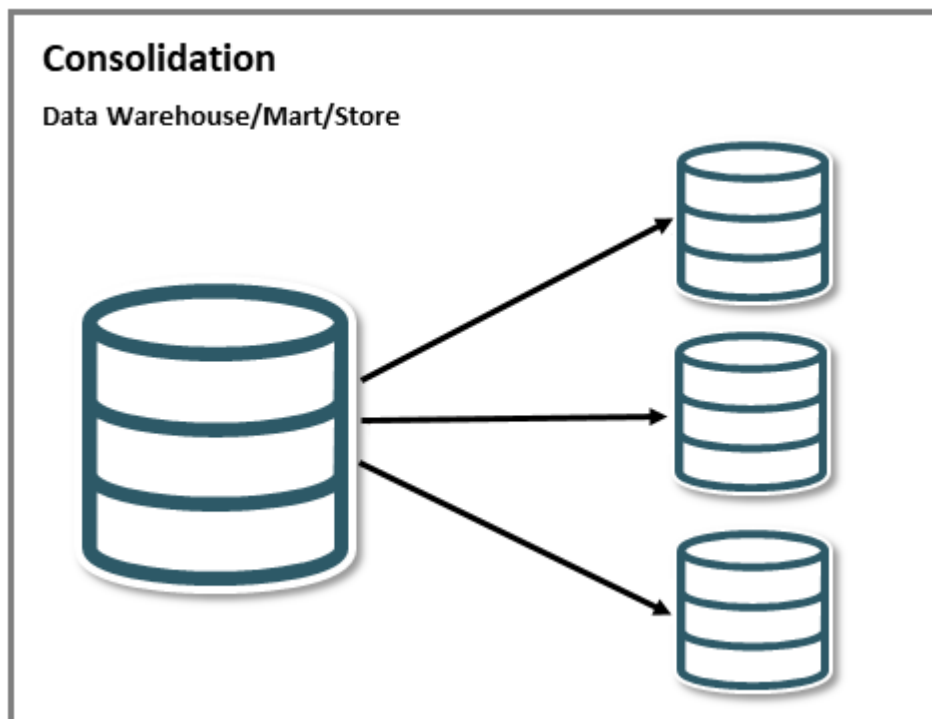


The following image shows central databases that are replicating changes from the Headquarter to the State to the County level in a hierarchical manner.



Consolidation

A data warehousing configuration is a many-to-one configuration. Multiple source databases send data to one target warehouse database. Oracle GoldenGate supports homogeneous and heterogeneous transfer of data, with capabilities for filtering and conversion on any system in the configuration although the support varies by database platform.



If data is being consolidated into a single target database, it is often necessary to track the lineage of that data. Using Oracle GoldenGate Tokens ([link](#)), additional metadata can be written to the target records. This would include the source database information on where this record originated, or when it was committed on the source database.

Oracle GoldenGate transformation routines can be used to further enhance the data as it is applied to the target database or data target.

Advanced Topologies

Apart from the standard topologies described in previous sections, including unidirectional, bidirectional, broadcast, cascading, and consolidation, there are many more patterns that can be configured using these topologies due to the architecture of Oracle GoldenGate.

For example, a broadcast topology can include unidirectional targets and bidirectional targets. Active-Active replication can be enhanced by adding an additional unidirectional replication stream from one of the active systems to a reporting database to offload reports. In an active-passive bidirectional topology, an additional bidirectional server could be added for the purpose of a zero downtime upgrade.

It is up to the businesses to decide the combination of topologies to use in a particular scenario or use case.

4

Replicate between Data Fabrics

This section covers commonly used Oracle GoldenGate architectures when replicating between two or more non-relational databases or *data fabrics*.

Solutions for Data Replication in Different Data Mesh or Data Fabric Environments

The entire purpose of creating a Data Fabric or a Data Mesh is to bring more value to enterprise data, making it better, faster, cheaper to achieve business transformation objectives.

Data Lake Ingestion

Data lake ingestion provides continuous, real-time capture and delivery of changed data between OLTP and object storage, data lake, or data lakehouse targets. Oracle GoldenGate delivers data to Oracle or non-Oracle platforms, such as Snowflake, Azure Data Lake, Google BigQuery, and other cloud object stores.

See Realtime Data Ingestion into Snowflake with GoldenGate for DAA in the *Oracle GoldenGate for Distributed Applications and Analytics*.

Real-time Streaming Pipeline and Analytics

Oracle Stream Analytics ingests real-time data from Oracle GoldenGate changed data, and from messaging environments like Kafka and JMS. A graphical pipeline designer can perform data transformations and apply time-series and geo-spatial analytics on these event streams. Oracle Stream Analytics is available standalone, a deployment option in OCI GoldenGate, and is included with Oracle Distributed Applications and Analytics.

See Realtime Message Ingestion to OCI Streaming with GoldenGate for DAA in the *Oracle GoldenGate for Distributed Applications and Analytics* as an example of this implementation.

Data Streams

Most Oracle GoldenGate DDL and DML capabilities can also be accessed through developer APIs. Deployed using the AsyncAPI standard, developers can subscribe and publish to the Oracle GoldenGate data event model.

See Async API in the Oracle GoldenGate Microservices Documentation.

To view the script for configuring a replication environment using Data Streams, see [Data Streams Script](#).

A

Automation Scripts

The objective of sample scripts provided for different scenarios is to provide a preparatory outline to set up a data replication environment for different business use cases. The scripts are can be used with cURL for the REST API framework and using OBEY commands in Admin Client to set up various environments for real-world scenarios.

Reporting

Scripts are available in cURL and as OBEY commands to be run in Admin Client, to test the business reporting scenario. You can use the scripts available in the following topics to:

- Add a data replication environment and view the processes that are created after the scripts runs successfully.
- Check the statistical reports for Daily, Hourly, and Total committed DML and DDL operations.
- Delete the data replication environment and verify that the environment was deleted after testing.

Business Reporting Configuration Using REST API Commands

Setup a Data Replication Environment Using REST API Commands

Copy and use the following cURL script to set up Oracle GoldenGate data replication environment on a pre-installed database.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

# Add UserIdAlias GGNORTH to connect to the Database instance DBNORTH
curl -s -k -X POST https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
```



```

        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"userid":"ggadmin@dbnorth","password":"xxxxxx"}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/credentials/OracleGoldenGate/
ggnorth \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

# Add UserIdAlias GGSOUTH to connect to the Database instance DBSOUTH
curl -s -k -X POST https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"userid":"ggadmin@dbsouth","password":"xxxxxx"}' | jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/credentials/OracleGoldenGate/
ggsouth \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
# --
# -- Add Schematranda
# -- Add Heartbeattable
# -- Add Checkpointtables
# --
#
-----

# Add Supplemental Logging to Database Schema HR (Schematranda) on source
database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic

```

```

Z2dtYTpHR2lhXzIzYWk='                                \
    -d '{"operation":"add","schemaName":"hr"}' | jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema              \
    -H "Content-Type: application/
json"                                                  \
    -H "Accept: application/
json"                                                  \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                                \
    -d '{"operation":"info","schemaName":"hr"}' | jq '.response'

# Add Heartbeatable on source database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat             \
    -H "Content-Type: application/
json"                                                  \
    -H "Accept: application/
json"                                                  \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                                \
    -d '{"frequency":60}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat             \
    -H "Content-Type: application/
json"                                                  \
    -H "Accept: application/
json"                                                  \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

# Add Checkpointtable on target database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint            \
    -H "Content-Type: application/
json"                                                  \
    -H "Accept: application/
json"                                                  \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                                \
    -d '{"operation":"add","name":"ggadmin.ggs_checkpointtable"}' | jq
'.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint            \
    -H "Content-Type: application/
json"                                                  \
    -H "Accept: application/
json"                                                  \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                                \
    -d '{"operation":"info","name":"ggadmin.ggs_checkpointtable"}' | jq
'.messages'

# Add Heartbeatable on target database GGSOUTH

```

```

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
-----
# --
# -- Add Extracts on source database GGNORTH
# --
#
-----
-----

curl -s -k -X POST https://north:9001/services/v2/extracts/
EXTN \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Extract - Region North"
, "config":["EXTRACT EXTN"
, "USERIDALIAS ggnorth"
, "EXTTRAIL north/ea"
, "DDL INCLUDE MAPPED"
, "DDLOPTIONS REPORT"
, "REPORTCOUNT EVERY 10 MINUTES, RATE"
, "WARNLONGTRANS 15MINUTES, CHECKINTERVAL 5MINUTES"
, "TABLE hr.*;"
]
, "source": "tranlogs"
, "credentials":{"alias":"ggnorth"}
, "registration": {"optimized": false}
, "begin":"now"
, "targets":[{"name":"ea", "path":"north/"}]
, "status":"running"
}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \

```

```

-H "Accept: application/
json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
# --
# -- Add DistPath from source to target system
# --
#
-----

curl -s -k -X POST https://north:9002/services/v2/sources/
DPNS
-H 'Content-Type: application/
json'
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
-d '{"description": "Region: From: North, To: South"
  , "name": "DPNS"
  , "source":
    { "uri": "trail://north:9002/services/v2/sources?trail=north/ea"
    , "details": { "encryption": { "algorithm": "NONE" } }
    }
  , "target":
    { "uri": "wss://south:9103/services/v2/targets?trail=north/da"
    , "authenticationMethod": { "certificate": "default" }
    , "details":
      { "trail": { "sizeMB": 100 }
      , "encryption": { "algorithm": "NONE" }
      , "compression": { "enabled": false }
      }
    }
  , "options":
    { "eofDelayCSecs": 10
    , "checkpointFrequency": 10
    , "critical": false
    , "autoRestart": { "retries": 10, "delay": 2 }
    , "streaming": true
    }
  , "begin": "now"
  , "encryptionProfile": "LocalWallet"
  , "status": "running"
}' | jq '.messages'

curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

```

```

#
-----
# --
# -- Add Replicat at target Database GGSOUTH
# --
#
-----

curl -s -k -X POST https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Replicat - Region South"
  ,"config":["REPLICAT repn"
    ,"USERIDALIAS ggsouth DOMAIN OracleGoldenGate"
    ,"DDL INCLUDE MAPPED"
    ,"DDLOPTIONS REPORT"
    ,"DDLERROR DEFAULT, DISCARD"
    ,"REPORTCOUNT EVERY 10 MINUTES, RATE"
    ,"REPEROR (DEFAULT, DISCARD)"
    ,"MAP hr.*, TARGET hr.*;"
  ]
  ,"credentials": {"alias": "ggsouth"}
  ,"mode": {"parallel":true,"type": "nonintegrated"}
  ,"source": {"name": "da", "path": "north"}
  ,"checkpoint":{"table": "ggadmin.ggs_checkpointtable"}
  ,"status": "running"
}' | jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----

```

Check the Statistical Reports Using REST API Commands

Copy and use the following cURL script to check the statistical data for the DDL and DML operations.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Extract EXN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Extract EXTN status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTphR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTN lag:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTphR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTN statistics:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
```

```

json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command":"STATS",
"arguments":"TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\t/\\t/g'| grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- DistPath DPNS"
echo "--"
echo
"-----"
-----"
echo " "

echo "DistPath DPNS status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

curl -s -k -X GET https://north:9002/services/v2/sources/DPNS/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\t/\\t/g'| grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Replicat REPN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Replicat REPN status:"

```

```

curl -s -k -X GET https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat REPN lag:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPN statistics:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

exit

#
-----
-----

```

Delete the Replication Environment Using REST API Commands

Copy and use the following cURL script to delete the data replication environment after you have tested it and do not need it.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
# --
# -- stop & Delete Replicat
# --
#
-----

curl -s -k -X PATCH https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"status":"stopped"}' |
jq '.messages'

curl -s -k -X DELETE https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

#
-----

# --
# -- Stop & Delete DistPath
# --
#
-----

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
```

```

        -d '{"status":"stopped"}'
    '.messages'

curl -s -k -X DELETE https://north:9002/services/v2/sources/
DPNS
    -H 'Content-Type: application/
json'
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
    '.messages'

#
-----
# --
# -- Stop & Delete Extracts
# --
#
-----

curl -s -k -X PATCH https://north:9001/services/v2/extracts/
EXTN
    -H 'Content-Type: application/
json'
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
    -d '{"status":"stopped"}'
jq '.messages'

curl -s -k -X DELETE https://north:9001/services/v2/extracts/
EXTN
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
    '.messages'

#
-----
# --
# -- Delete Schematranda
# -- Delete Checkpointtable
# -- Delete Heartbeattables
# --
#
-----

# Delete SchemaTrandata at Database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema
    -H "Content-Type: application/
json"
    -H "Accept: application/

```

```

json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"delete","schemaName":"hr"}' | jq
'.messages'

# Delete Checkpointtable at Database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json"
\
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"delete","name":"ggadmin.ggs_checkpointtable"}' | jq
'.messages' ###

# Delete Heartbeatable at Database GGNORTH
curl -s -k -X DELETE https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

# Delete Heartbeatable at Database GGSOUTH
curl -s -k -X DELETE https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

#
-----
# --
# --Delete USERIDALIAS from GoldenGate
# --
#
-----

curl -s -k -X DELETE https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq

```

```

'.messages'

curl -s -k -X DELETE https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

#
-----
# --
# -- Delete Trail Files
# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
    , "purgeType": "trails"
    , "trails": [{"name": "ea", "path": "north/"}]
    , "useCheckpoints": false
    , "keep": [{"type": "min", "units": "files", "value": 0}]
    }' |
jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
    , "purgeType": "trails"
    , "trails": [{"name": "da", "path": "north/"}]
    , "useCheckpoints": false
    , "keep": [{"type": "min", "units": "files", "value": 0}]
    }' |
jq '.messages'

```

Business Reporting Configuration Using Admin Client Commands

Set Up a Data Replication Environment Using Admin Client Commands

Copy and use the following OBEY script to set up Oracle GoldenGate data replication environment on a pre-installed database.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
--
-- Connect the GoldenGate Deployment depl_north
--
CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD xxxxxx !

ALTER CREDENTIALSTORE ADD USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD HEARTBEATTABLE

ADD EXTRACT extn INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT extn database
ADD EXTTRAIL north/ea, EXTRACT extn
START EXTRACT extn

ADD DISTPATH dpns SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://south:9103/services/v2/targets?trail=north/da !
START DISTPATH dpns

--
-- Connect the GoldenGate Deployment depl_south
--

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD xxxxxx !

ALTER CREDENTIALSTORE ADD USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

ADD REPLICAT repn, PARALLEL, EXTTRAIL north/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
```

```
START REPLICAT repn
INFO ALL
INFO DISTPATH ALL

DISCONNECT
```

After creating the OBEY file, create a shell script to run in Admin Client:

```
cp EXTN.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/
cp REPN.prm /u01/app/oracle/deployments/depl_south/etc/conf/ogg/

echo "obey add_replication_reporting.oby" | adminclient
```

Check the Statistical Reports Using Admin Client Commands

Copy and use the following OBEY script by adjusting the values for each of the options with the options in your environment, to check the statistical data for the DDL and DML operations.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
--
-- Connect the GoldenGate Deployment depl_north
--
CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !

INFO EXTRACT extn
SEND EXTRACT extn, GETLAG
STATS EXTRACT extn, TOTAL, TOTALONLY *.*

INFO DISTPATH DPNS
STATS DISTPATH DPNS
--
-- Connect the GoldenGate Deployment depl_south
--

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !

INFO REPLICAT repn
SEND REPLICAT repn, GETLAG
STATS REPLICAT repn, TOTAL, TOTALONLY *.*

DISCONNECT
```

After you create the preceding .oby file, create a shell script to run in Admin Client that would run the .oby commands:

```
echo "obey check_replication_reporting.oby" | adminclient
```

Delete the Replication Environment Using Admin Client Commands

Copy and use the following OBEY script by adjusting the values for each of the options with the options in your environment, to delete the data replication environment.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
--
-- Connect the GoldenGate Deployment depl_south
--

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !

DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate
KILL REPLICAT repn
DELETE REPLICAT repn

DELETE CHECKPOINTTABLE ggadmin.ggs_checkpointtable !
DELETE HEARTBEATABLE !

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate
INFO CREDENTIALSTORE

PURGE EXTTRAIL north/da

-- Connect the GoldenGate Deployment depl_north
--

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
STOP DISTPATH dpns
DELETE DISTPATH dpns

DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
KILL EXTRACT extn
DELETE EXTRACT extn

INFO ALL
INFO DISTPATH ALL

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate
INFO CREDENTIALSTORE
```

```
PURGE EXTTRAIL north/ea  
DISCONNECT
```

After you create the preceding .oby file, create a shell script to run in Admin Client that would run the .oby commands:

```
rm -f /u01/app/oracle/deployments/depl_north/etc/conf/ogg/EXTN.prm  
rm -f /u01/app/oracle/deployments/depl_south/etc/conf/ogg/REPN.prm  
  
echo "obey delete_replication_reporting.oby" | adminclient
```

Common Files and Scripts Used with Business Reporting in a Unidirectional Topology

Extract Parameter File in Unidirectional Replication

Use the following Extract parameter file for configuring the business reporting use case when using the Admin Client commands in the OBEY script. This is required because you need to edit or copy the parameter file to the `etc/conf/ogg/`.

The Extract name used in this parameter file is `EXTN`:

```
EXTRACT extn  
USERIDALIAS ggnorth  
EXTTRAIL north/ea  
  
DDL INCLUDE MAPPED  
DDLOPTIONS REPORT  
  
REPORTCOUNT EVERY 10 MINUTES, RATE  
WARNLONGTRANS 15MINUTES, CHECKINTERVAL 5MINUTES  
  
TABLE hr.*;
```

Replicat Parameter File in Unidirectional Replication

Use the following Replicat parameter file for configuring the business reporting use case when using the Admin Client commands in the OBEY script: This is required because you need to edit or copy the parameter file to the `/etc/conf/ogg/`.

```
REPN  
  
REPLICAT repn  
USERIDALIAS ggsouth DOMAIN OracleGoldenGate  
  
DDLOPTIONS REPORT  
DDLERROR DEFAULT, DISCARD  
  
REPORTCOUNT EVERY 10 MINUTES, RATE
```



```
REPERROR (DEFAULT, DISCARD)
MAP hr.*, TARGET hr.*;
```

Source DML Operations

```
#!/usr/bin/bash

sqlplus -s ggadmin/ggadmin@dbnorth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 999999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

PROMPT Run DML operations on source table HR.EMPLOYEES

begin
  for i in 1..10 loop
    update hr.employees set salary = 0.95 * salary where department_id = 60;
    dbms_lock.sleep(1);
    commit;
  end loop;
end;
/

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

EOF
exit
```

Sample Select Statements for Source and Target PDBs

```
#!/usr/bin/bash

sqlplus -s /nolog << EOF

set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 999999.99
col department_id   format 999

-- Connect to DBNorth
connect ggadmin/ggadmin@dbnorth
```

```
PROMPT
PROMPT Source Database:

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

-- Connect to DBSouthh
connect ggadmin/ggadmin@dbsouth

PROMPT Target Database:

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Bidirectional Replication Using Active-Active Configuration

You can use the scripts available in the following topics to:

- Add a data replication environment and view the processes that are created after the scripts runs successfully.
- Check for bidirectional data replication between DBWEST and DBEAST PDBs.
- Delete the data replication environment and verify that the environment was deleted after testing.

Active-Active Configuration Using REST API Commands

Create Bidirectional Configuration Using REST API Commands

Copy and use the following cURL script to set up Oracle GoldenGate data replication environment on a pre-installed database and Oracle GoldenGate deployment.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/bin/bash

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

#
```

```

-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----

# Add UserIdAlias GGNORTH to connect to the Database instance DBNORTH
curl -s -k -X POST https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"userid":"ggadmin@dbnorth","password":"ggadmin"}' |
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/credentials/OracleGoldenGate/
ggnorth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add UserIdAlias GGSOUTH to connect to the Database instance DBSOUTH
curl -s -k -X POST https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"userid":"ggadmin@dbsouth","password":"ggadmin"}' |
jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/credentials/OracleGoldenGate/
ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

#
-----
# --
# -- Add Schematranda

```

```

# -- Add Heartbeatable
# -- Add Checkpointtables
# --
#
-----

# Add Supplemental Logging to Database Schema HR (Schematrandata) on database
GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","schemaName":"hr"}' |
jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","schemaName":"hr"}' |
jq '.response'

# Add Supplemental Logging to Database Schema HR (Schematrandata) on database
GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","schemaName":"hr"}' |
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","schemaName":"hr"}' |
jq '.response'

# Add Heartbeatable on source database GGNORTH

```

```

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' |
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add Heartbeattable on target database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' |
jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add Checkpointtable on database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/checkpoint \
-H "Content-Type: application/

```

```

json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

# Add Checkpointtable on database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

#
-----
# --
# -- Add Extracts on Databases
# --
#
-----

# Add Extract EXTN on Database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/extracts/
EXTN \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Extract - Region North"
, "config":["EXTRACT EXTN"
, "USERIDALIAS ggnorth"
, "EXTTRAIL north/ea"
, "TRANLOGOPTIONS EXCLUDETAG +"
, "DDL INCLUDE MAPPED"

```

```

        , "DDLOPTIONS REPORT"
        , "REPORTCOUNT EVERY 10 MINUTES, RATE"
        , "WARNLONGTRANS 15 MINUTES, CHECKINTERVAL 5 MINUTES"
        , "TABLE hr.*;"
    ]
    , "source": "tranlogs"
    , "credentials": { "alias": "ggnorth" }
    , "registration": { "optimized": false }
    , "begin": "now"
    , "targets": [ { "name": "ea", "path": "north/" } ]
    , "status": "running"
} '
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

# Add Extract EXTS on Database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/extracts/
EXTS \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{ "description": "Extract - Region South"
    , "config": [ "EXTRACT EXTS"
        , "USERIDALIAS ggsouth"
        , "EXTTRAIL south/ea"
        , "TRANLOGOPTIONS EXCLUDETAG +"
        , "DDL INCLUDE MAPPED"
        , "DDLOPTIONS REPORT"
        , "REPORTCOUNT EVERY 10 MINUTES, RATE"
        , "WARNLONGTRANS 15 MINUTES, CHECKINTERVAL 5 MINUTES"
        , "TABLE hr.*;"
    ]
    , "source": "tranlogs"
    , "credentials": { "alias": "ggsouth" }
    , "registration": { "optimized": false }
    , "begin": "now"
    , "targets": [ { "name": "ea", "path": "south/" } ]
    , "status": "running"
} ' | jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/extracts/
EXTS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \

```

```

        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.response'

#
-----
# --
# -- Add DistPaths to Systems to route Trailfiles in each direction
# --
#
-----

# Add DistPath DPNS from North to South
curl -s -k -X POST https://north:9002/services/v2/sources/
DPNS \
        -H 'Content-Type: application/
json' \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"description": "Region: From: North, To: South"
        , "name": "DPNS"
        , "source":
        { "uri": "trail://north:9002/services/v2/sources?trail=north/ea"
        , "details": { "encryption": { "algorithm": "NONE" } }
        }
        , "target":
        { "uri": "wss://south:9103/services/v2/targets?trail=north/da"
        , "authenticationMethod": { "certificate": "default" }
        , "details":
        { "trail": { "sizeMB": 100 }
        , "encryption": { "algorithm": "NONE" }
        , "compression": { "enabled": false }
        }
        }
        , "options":
        { "eofDelayCSecs": 10
        , "checkpointFrequency": 10
        , "critical": false
        , "autoRestart": { "retries": 10, "delay": 2 }
        , "streaming": true
        }
        , "begin": "now"
        , "encryptionProfile": "LocalWallet"
        , "status": "running"
        }'
jq '.messages'

curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \

```



```

        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.response'

# Add DistPath DPSN from South to North
curl -s -k -X POST https://south:9102/services/v2/sources/
DPSN \
        -H 'Content-Type: application/
json' \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"description": "Region: From: South, To: North"
        , "name": "DPSN"
        , "source":
        { "uri": "trail://south:9102/services/v2/sources?trail=south/ea"
        , "details": { "encryption": { "algorithm": "NONE" } }
        }
        , "target":
        { "uri": "wss://north:9003/services/v2/targets?trail=south/da"
        , "authenticationMethod": { "certificate": "default" }
        , "details":
        { "trail": { "sizeMB": 100 }
        , "encryption": { "algorithm": "NONE" }
        , "compression": { "enabled": false }
        }
        }
        , "options":
        { "eofDelayCsecs": 10
        , "checkpointFrequency": 10
        , "critical": false
        , "autoRestart": { "retries": 10, "delay": 2 }
        , "streaming": true
        }
        , "begin": "now"
        , "encryptionProfile": "LocalWallet"
        , "status": "running"
        }'
jq '.messages'

curl -s -k -X GET https://south:9102/services/v2/sources/
DPSN \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'
#
-----
# --
# -- Add Replicats to Databases
# --
#
-----
-----

```

```

# Add Replicat REPS on Database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/replicats/
REPS
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
    -d '{"description":"Replicat - Region North"
        , "config": ["REPLICAT reps"
            , "USERIDALIAS ggnorth DOMAIN OracleGoldenGate"
            , "DDL INCLUDE MAPPED"
            , "DDLOPTIONS REPORT"
            , "DDLERROR DEFAULT, DISCARD"
            , "REPORTCOUNT EVERY 10 MINUTES, RATE"
            , "REPEROR (DEFAULT, DISCARD)"
            , "MAP hr.*, TARGET hr.*;"
        ]
        , "credentials": {"alias": "ggnorth"}
        , "mode": {"parallel": true, "type": "integrated"}
        , "source": {"name": "da", "path": "south"}
        , "checkpoint": {"table": "ggadmin.ggs_checkpointtable"}
        , "status": "running"
    }'
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/replicats/
REPS
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

# Add Replicat REPN on Database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/replicats/
REPN
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
    -d '{"description":"Replicat - Region South"
        , "config": ["REPLICAT repn"
            , "USERIDALIAS ggsouth DOMAIN OracleGoldenGate"
            , "DDL INCLUDE MAPPED"
            , "DDLOPTIONS REPORT"
            , "DDLERROR DEFAULT, DISCARD"
            , "REPORTCOUNT EVERY 10 MINUTES, RATE"
            , "REPEROR (DEFAULT, DISCARD)"
            , "MAP hr.*, TARGET hr.*;"
        ]
        , "credentials": {"alias": "ggsouth"}
    }'

```

```

        , "mode": { "parallel": true, "type": "integrated" }
        , "source": { "name": "da", "path": "north" }
        , "checkpoint": { "table": "ggadmin.ggs_checkpointtable" }
        , "status": "running"
      }'
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/replicats/
REPN \
      -H "Content-Type: application/
json" \
      -H "Accept: application/
json" \
      -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.response'

#
-----
-----

```

Check the Statistics in a Bidirectional Environment Using REST API Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

#!/bin/bash

#
-----
-----

# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

echo " "
echo
"-----"
-----"
echo "--"
echo "-- GoldenGate Instance NORTH: Status of Extract (EXTN), DistPath
(DPNS), and Replicat (REPS) "
echo "--"
echo
"-----"
-----"
echo " "

```

```

echo "Extract EXTN status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTN lag:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTN statistics:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPS status:"
curl -s -k -X GET https://north:9001/services/v2/replicats/
REPS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "DistPath DPNS status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

curl -s -k -X GET https://north:9002/services/v2/sources/DPNS/

```

```

stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\t/\\t/g'| grep -v OKNODOT

echo "Replicat REPS lag:"
curl -s -k -X POST https://north:9001/services/v2/replicats/REPS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g'| grep -v OKNODOT

echo "Replicat REPS statistics:"
curl -s -k -X POST https://north:9001/services/v2/replicats/REPS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g'| grep -v
OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- GoldenGate Instance NORTH: Status of Extract (EXTN), DistPath
(DPNS), and Replicat (REPS) "
echo "--"
echo
"-----"
-----"
echo " "

echo "Extract EXTS status:"
curl -s -k -X GET https://south:9101/services/v2/extracts/
EXTS \
-H "Content-Type: application/
json" \
-H "Accept: application/

```

```

json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTS lag:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply'
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTS statistics:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
-d '{"command": "STATS",
"arguments": "TOTAL"}'
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo "DistPath DPSN status:"
curl -s -k -X GET https://south:9102/services/v2/sources/
DPSN
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

curl -s -k -X GET https://south:9102/services/v2/sources/DPSN/
stats
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPN status:"
curl -s -k -X GET https://south:9101/services/v2/replicats/
REPN
-H "Content-Type: application/
json"
-H "Accept: application/

```

```

json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status' \

echo "Replicat REPN lag:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/"
json" \
-H "Accept: application/"
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPN statistics:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/"
json" \
-H "Accept: application/"
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

exit

#
-----
-----

```

Remove the Bidirectional Replication Setup Using REST API Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

#!/bin/bash

#
-----
-----
# --
# -- stop & Delete Replicat
# --

```

```

#
-----

curl -s -k -X PATCH https://north:9001/services/v2/replicats/
REPS \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' |
jq '.messages'

curl -s -k -X DELETE https://north:9001/services/v2/replicats/
REPS \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

curl -s -k -X PATCH https://south:9101/services/v2/replicats/
REPN \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' |
jq '.messages'

curl -s -k -X DELETE https://south:9101/services/v2/replicats/
REPN \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

#
-----

# --
# -- Stop & Delete DistPath
# -- Purge Target Trail File
# --
#
-----

```



```

-----

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' |
jq '.messages'

curl -s -k -X DELETE https://north:9002/services/v2/sources/
DPNS \
    -H 'Content-Type: application/
json' \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

curl -s -k -X PATCH https://south:9102/services/v2/sources/
DPSN \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' |
jq '.messages'

curl -s -k -X DELETE https://south:9102/services/v2/sources/
DPSN \
    -H 'Content-Type: application/
json' \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

#
-----
# --
# -- Stop & Delete Extracts
# --
#
-----

curl -s -k -X PATCH https://north:9001/services/v2/extracts/
EXTN \
    -H 'Content-Type: application/
json' \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' |

```

```
jq '.messages'
```

```
curl -s -k -X DELETE https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'
```

```
curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
, "purgeType": "trails"
, "trails": [{"name": "ea", "path": "north/"}]
, "useCheckpoints": false
, "keep": [{"type": "min", "units": "files", "value": 0}]
}' |
jq '.messages'
```

```
curl -s -k -X PATCH https://south:9101/services/v2/extracts/
EXTS \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"status": "stopped"}' |
jq '.messages'
```

```
curl -s -k -X DELETE https://south:9101/services/v2/extracts/
EXTS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'
```

```
curl -s -k -X POST https://south:9101/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
```

```

        , "purgeType": "trails"
        , "trails": [{ "name": "ea", "path": "south/" }]
        , "useCheckpoints": false
        , "keep": [{ "type": "min", "units": "files", "value": 0 }]
    },
  ],
  "messages": [
    {
      "id": 1,
      "type": "delete",
      "target": "SchemaTrandata",
      "details": {
        "operation": "delete",
        "schemaName": "hr"
      }
    }
  ]
}
jq '.messages'

#
-----

# --
# -- Delete Schematranda
# -- Delete Checkpointtable
# -- Delete Heartbeattables
# --
#
-----

# Delete SchemaTrandata at Databases
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation": "delete", "schemaName": "hr"}'
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation": "delete", "schemaName": "hr"}'
jq '.messages'

# Delete Checkpointtables at Databases
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation": "delete", "name": "ggadmin.ggs_checkpointtable"}'
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \

```

```

        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"operation":"delete","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

# Delete Heartbeatable at Database GGNORTH
curl -s -k -X DELETE https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

# Delete Heartbeatable at Database GGSOUTH
curl -s -k -X DELETE https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

#
-----
# --
# -- Delete USERIDALIAS from GoldenGate
# --
#
-----
-----

```

```

curl -s -k -X DELETE https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

curl -s -k -X DELETE https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/credentials/
OracleGoldenGate \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

#
-----
# --
# -- Delete Trail Files
# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
    , "purgeType": "trails"
    , "trails": [{"name": "da", "path": "north/"}]
    , "useCheckpoints": false
    , "keep": [{"type": "min", "units": "files", "value": 0}]
    }' |
jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/

```

```

execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
    , "purgeType": "trails"
    , "trails": [{"name": "da", "path": "south/"}]
    , "useCheckpoints": false
    , "keep": [{"type": "min", "units": "files", "value": 0}]
  }'
jq '.messages'

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

```

Active-Active Configuration Using Admin Client Commands

Bidirectional Configuration Using the Admin Client Commands

Copy and use the following OBEY script to set up Oracle GoldenGate data replication environment on an pre-installed database.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate

```

```

ADD SCHEMATRANDATA hr
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate

ADD EXTRACT extn INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT extn database
ADD EXTTRAIL north/ea, EXTRACT extn
START EXTRACT extn

ADD DISTPATH dpns SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://south:9103/services/v2/targets?trail=north/da !
START DISTPATH dpns

ADD REPLICAT reps, PARALLEL, EXTTRAIL south/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
START REPLICAT reps

INFO ALL
INFO DISTPATH ALL

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate

ADD EXTRACT exts INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT exts database
ADD EXTTRAIL south/ea, EXTRACT exts
START EXTS

ADD DISTPATH dpsn SOURCE trail://south:9102/services/v2/sources?
trail=south/ea TARGET wss://north:9103/services/v2/targets?trail=south/da !
START DISTPATH dpsn

ADD REPLICAT repn, PARALLEL, EXTTRAIL north/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
START REPLICAT repn

INFO ALL
INFO DISTPATH ALL

DISCONNECT

```

After creating the OBEY file, create a shell script to run in Admin Client. This script runs triggers the bidirectional replication setup using the .oby file:

```

#!/bin/bash

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

# Copy parameter file (usually edited with EDIT PARAMs)
cp EXTN.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/

```

```

cp REPS.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/

cp EXTS.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/
cp REPN.prm /u01/app/oracle/deployments/depl_south/etc/conf/ogg/

# Run the GoldenGate Obey script
echo "obey add_replication_ActiveActive.oby" | adminclient | tee
add_replication_ActiveActive.log

```

Check the Processes in the Bidirectional Environment Using Admin Client

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

--
-- Connect the GoldenGate Deployment depl_north
--
CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !

INFO  EXTRACT extn
SEND  EXTRACT extn, GETLAG
STATS EXTRACT extn, TOTAL, TOTALSONLY *.*

INFO DISTPATH DPNS
STATS DISTPATH DPNS

INFO REPLICAT reps
SEND  REPLICAT reps, GETLAG
STATS REPLICAT reps, TOTAL, TOTALSONLY *.*

--
-- Connect the GoldenGate Deployment depl_south
--

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !

INFO  EXTRACT exts
SEND  EXTRACT exts, GETLAG
STATS EXTRACT exts, TOTAL, TOTALSONLY *.*

INFO DISTPATH DPSN
STATS DISTPATH DPSN

INFO REPLICAT repn
SEND  REPLICAT repn, GETLAG
STATS REPLICAT repn, TOTAL, TOTALSONLY *.*

DISCONNECT

```


After creating the OBEY file, create a shell script to run in Admin Client:

```
echo "obey check_replication_ActiveActive.oby" | adminclient
```

Remove the Bidirectional Replication Environment Using Admin Client Commands

① Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !  
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
```

```
KILL    EXTRACT  extn  
DELETE  EXTRACT  extn  
PURGE   EXTTRAIL north/ea
```

```
STOP    DISTPATH dpns  
DELETE  DISTPATH dpns  
PURGE   EXTTRAIL south/da
```

```
KILL    REPLICAT reps  
DELETE  REPLICAT reps
```

```
INFO ALL  
INFO DISTPATH ALL
```

```
DELETE SCHEMATRANDATA hr  
DELETE CHECKPOINTTABLE ggadmin.ggs_checkpointtable !  
DELETE HEARTBEATABLE !
```

```
ALTER CREDENTIALSTORE DELETE USER ggadmin@dbnorth ALIAS ggnorth DOMAIN  
OracleGoldenGate  
INFO CREDENTIALSTORE
```

```
CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !  
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate
```

```
KILL    EXTRACT  exts  
DELETE  EXTRACT  exts  
PURGE   EXTTRAIL south/ea
```

```
STOP    DISTPATH dpsn  
DELETE  DISTPATH dpsn  
PURGE   EXTTRAIL north/da
```

```
KILL    REPLICAT repn  
DELETE  REPLICAT repn
```

```
INFO ALL
INFO DISTPATH ALL

DELETE SCHEMATRANDATA hr
DELETE CHECKPOINTTABLE ggadmin.ggs_checkpointtable !
DELETE HEARTBEATABLE !

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate
INFO CREDENTIALSTORE

DISCONNECT
```

After creating the OBEY file, create a shell script to run in Admin Client:

```
#!/bin/bash

# Run the GoldenGate Obey script
echo "obey delete_replication_ActiveActive.oby" | adminclient | tee
delete_replication_ActiveActive.log

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null
```

Common Files and Scripts Used in a Bidirectional Replication

DBNORTH DML Operations

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbnorth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

begin
  for i in 1..10 loop
    update hr.employees set salary = 1.05 * salary where department_id =60;
    dbms_lock.sleep(1);
    commit;
  end loop;
end;
/
```

```
select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Select Statement for DBSOUTH

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbsouth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Source DML Operations

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbsouth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

begin
  for i in 1..10 loop
    update hr.employees set salary = 1.05 * salary where department_id =60;
    dbms_lock.sleep(1);
    commit;
  end loop;
end;
/
```

```
select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Select Statement for DBSOUTH PDBs

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbsouth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Configure a Cascading Topology Replication

Scripts are available in cURL and as OBEY commands to be run in a REST API client or Admin Client, to test the cascading topology across three PDBs. You can use the scripts available in the following topics to:

- Add a data replication environment where the source PDB has an Extract process that captures from the source PDB and transmits trail data to the intermediate PDB which contains a Replicat and an Extract process. On the intermediate instance, the Replicat applies the trail data from source PDB and then the Extract captures the committed transactions and transmits trail data to the target PDB. See [Cascading Configuration Real-World Use Cases](#) for a conceptual understanding of this type of topology.
- Check the statistical reports for Daily, Hourly, and Total committed DML and DDL operations.
- Delete the data replication environment and verify that the environment was deleted after testing.

Cascading Topology Data Replication using REST API Commands

Create a Cascading Data Replication Configuration Using REST API Commands

Copy and use the following cURL script to set up Oracle GoldenGate data replication environment on a pre-installed database.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/bin/bash

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

#
-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----

# Add UserIdAlias GGNORTH to connect to the Database instance DBNORTH
curl -s -k -X POST https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"userid":"ggadmin@dbnorth","password":"ggadmin"}' |
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/credentials/OracleGoldenGate/
ggnorth \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add UserIdAlias GGSOUTH to connect to the Database instance DBSOUTH
curl -s -k -X POST https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"userid":"ggadmin@dbsouth","password":"ggadmin"}' |
jq '.messages'
```

```

curl -s -k -X GET https://south:9101/services/v2/credentials/OracleGoldenGate/
ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add UserIdAlias GGWEST to connect to the Database instance DBSOUTH
curl -s -k -X POST https://west:9201/services/v2/credentials/OracleGoldenGate/
ggwest \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"userid":"ggadmin@dbwest","password":"ggadmin"}' |
jq '.messages'

curl -s -k -X GET https://west:9201/services/v2/credentials/OracleGoldenGate/
ggwest \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

#
-----
# --
# -- Add Schematranda
# -- Add Heartbeattable
# -- Add Checkpointtables
# --
#
-----

# Add Supplemental Logging to Database Schema HR (Schematranda) on source
database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","schemaName":"hr"}' |
jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/connections/

```

```

OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info", "schemaName":"hr"}' |
jq '.response'

# Add Supplemental Logging to Database Schema HR (Schematrandata) on
intermediate database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add", "schemaName":"hr"}' |
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info", "schemaName":"hr"}' |
jq '.response'

# Add Heartbeatable on source database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' |
jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add Heartbeatable on target database GGSOUTH

```

```

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' |
jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add Heartbeattable on target database GGWEST
curl -s -k -X POST https://west:9201/services/v2/connections/
OracleGoldenGate.ggwest/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' |
jq '.messages'

curl -s -k -X GET https://west:9201/services/v2/connections/
OracleGoldenGate.ggwest/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

# Add Checkpointtable on intermediate database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/

```



```

json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

# Add Checkpointtable on target database GGWEST
curl -s -k -X POST https://west:9201/services/v2/connections/
OracleGoldenGate.ggwest/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

curl -s -k -X POST https://west:9201/services/v2/connections/
OracleGoldenGate.ggwest/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

#
-----
# --
# -- Add Extracts on source database GGNORTH
# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/extracts/
EXTN \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Extract - Region North"
, "config":["EXTRACT extn"
, "USERIDALIAS ggnorth"
, "EXTTRAIL north/ea"
, "DDL INCLUDE MAPPED"
, "DDLOPTIONS REPORT"
, "REPORTCOUNT EVERY 10 MINUTES, RATE"
, "WARNLONGTRANS 15MINUTES, CHECKINTERVAL 5MINUTES"
, "TABLE hr.*;"]}'

```

```

        ]
        ,"source": "tranlogs"
        ,"credentials":{"alias":"ggnorth"}
        ,"registration": {"optimized": false}
        ,"begin":"now"
        ,"targets":[{"name":"ea", "path":"north/"}]
        ,"status":"running"
    }'
'.messages'

curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

#
-----
# --
# -- Add DistPath from source to intermediate system
# --
#
-----

curl -s -k -X POST https://north:9002/services/v2/sources/
DPNS \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description": "Region: From: North, To: South"
    ,"name": "DPNS"
    ,"source":
      {"uri": "trail://north:9002/services/v2/sources?trail=north/ea"
      ,"details": {"encryption": {"algorithm": "NONE"}}
    }
    ,"target":
      {"uri": "wss://south:9103/services/v2/targets?trail=north/da"
      ,"authenticationMethod": {"certificate": "default"}
      ,"details":
        {"trail": {"sizeMB": 100}
        ,"encryption": {"algorithm": "NONE"}
        ,"compression": {"enabled": false}
        }
      }
    ,"options":
      {"eofDelayCSecs": 10
      ,"checkpointFrequency": 10
      ,"critical": false

```

```

        , "autoRestart": { "retries": 10, "delay": 2 }
        , "streaming": true
      }
    , "begin": "now"
    , "encryptionProfile": "LocalWallet"
    , "status": "running"
  }'
jq '.messages'

curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

#
-----
# --
# -- Add Replicat at target Database GGSOUTH
# --
#
-----

curl -s -k -X POST https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description": "Replicat - Region South"
  , "config": [ "REPLICAT repn"
    , "USERIDALIAS ggsouth DOMAIN OracleGoldenGate"
    , "DDL INCLUDE MAPPED"
    , "DDLOPTIONS REPORT"
    , "DDLERROR DEFAULT, DISCARD"
    , "REPORTCOUNT EVERY 10 MINUTES, RATE"
    , "REPEROR (DEFAULT, DISCARD)"
    , "MAP hr.*, TARGET hr.*;"
  ]
  , "credentials": { "alias": "ggsouth" }
  , "mode": { "parallel": true, "type": "nonintegrated" }
  , "source": { "name": "da", "path": "north" }
  , "checkpoint": { "table": "ggadmin.ggs_checkpointtable" }
  , "status": "running"
}' |
jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/replicats/
REPN \

```

```

        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

#
-----
# --
# -- Add 2nd Extract EXTS on source database GGSOUTH
# --
#
-----

curl -s -k -X POST https://south:9101/services/v2/extracts/
EXTS \
        -H 'Content-Type: application/
json' \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"description":"Extract - Region South"
            , "config":["EXTRACT exts"
                        , "USERIDALIAS ggsouth"
                        , "EXTTRAIL south/ea"
                        , "DDL INCLUDE MAPPED"
                        , "DDLOPTIONS INCLUDETAG 00"
                        , "DDLOPTIONS REPORT"
                        , "REPORTCOUNT EVERY 10 MINUTES, RATE"
                        , "WARNLONGTRANS 15MINUTES, CHECKINTERVAL 5MINUTES"
                        , "TABLE hr.*;"
                        ]
            , "source": "tranlogs"
            , "credentials":{"alias":"ggsouth"}
            , "registration": {"optimized": false}
            , "begin":"now"
            , "targets":[{"name":"ea", "path":"south/"}]
            , "status":"running"
            }' |
jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/extracts/
EXTS \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.response'

#
-----
# --

```

```

# -- Add 2nd DistPath DPSW from target to target system
# --
#
-----

curl -s -k -X POST https://south:9102/services/v2/sources/
DPSW \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description": "Region: From: South, To: West"
, "name": "DPSW"
, "source":
{ "uri": "trail://south:9102/services/v2/sources?trail=south/ea"
, "details": { "encryption": { "algorithm": "NONE" } }
}
, "target":
{ "uri": "wss://west:9203/services/v2/targets?trail=south/da"
, "authenticationMethod": { "certificate": "default" }
, "details":
{ "trail": { "sizeMB": 100 }
, "encryption": { "algorithm": "NONE" }
, "compression": { "enabled": false }
}
}
, "options":
{ "eofDelayCSecs": 10
, "checkpointFrequency": 10
, "critical": false
, "autoRestart": { "retries": 10, "delay": 2 }
, "streaming": true
}
, "begin": "now"
, "encryptionProfile": "LocalWallet"
, "status": "running"
}'
jq '.messages'

curl -s -k -X GET https://south:9102/services/v2/sources/
DPSW \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.response'

#
-----

# --
# -- Add Replicat REPS at target Database GGWEST

```

```

# --
#
-----
-----

curl -s -k -X POST https://west:9201/services/v2/replicats/
REPS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Replicat - Region West"
  ,"config":["REPLICAT reps"
    ,"USERIDALIAS ggwest DOMAIN OracleGoldenGate"
    ,"DDL INCLUDE MAPPED"
    ,"DDLOPTIONS REPORT"
    ,"DDLERROR DEFAULT, DISCARD"
    ,"REPORTCOUNT EVERY 10 MINUTES, RATE"
    ,"REPEROR (DEFAULT, DISCARD)"
    ,"MAP hr.*, TARGET hr.*;"
  ]
  ,"credentials": {"alias": "ggwest"}
  ,"mode": {"parallel":true,"type": "nonintegrated"}
  ,"source": {"name": "da", "path": "south"}
  ,"checkpoint":{"table": "ggadmin.ggs_checkpointtable"}
  ,"status": "running"
}'
jq '.messages'

curl -s -k -X GET https://west:9201/services/v2/replicats/
REPS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.response'

#
-----
-----

```

Check the Transaction Statistics for a Cascading Configuration Using REST API Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/bin/bash

#
-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Extract EXN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Extract EXTN status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTN lag:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
```

```

        -d '{"command": "GETLAG", "isReported": true}' | jq
    '.response.reply'
        | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTN statistics:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
        -d '{"command": "STATS",
"arguments": "TOTAL"}'
        | jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
echo "--"
echo "-- DistPath DPNS"
echo "--"
echo
"-----"
echo " "

echo "DistPath DPNS status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

curl -s -k -X GET https://north:9002/services/v2/sources/DPNS/
stats
    -H "Content-Type: application/
json"
    -H "Accept: application/
json"
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
        | jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"

```



```

-----"
echo "--"
echo "-- Replicat REPN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Replicat REPN status:"
curl -s -k -X GET https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat REPN lag:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPN statistics:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Extract EXTS "
echo "--"
echo
"-----"
-----"
echo " "

```

```

echo "Extract EXTS status:"
curl -s -k -X GET https://south:9101/services/v2/extracts/
EXTS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTS lag:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTS statistics:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- DistPath DPSW"
echo "--"
echo
"-----"
-----"
echo " "

echo "DistPath DPSW status:"
curl -s -k -X GET https://south:9102/services/v2/sources/
DPSW \
-H "Content-Type: application/
json" \
-H "Accept: application/

```

```

json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status' \

curl -s -k -X GET https://south:9102/services/v2/sources/DPSW/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Replicat REPS "
echo "--"
echo
"-----"
-----"
echo " "

echo "Replicat REPS status:"
curl -s -k -X GET https://west:9201/services/v2/replicats/
REPS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat REPS lag:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPS statistics:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic

```

```

Z2dtYTpHR2lhXzIzYWk='                                     \
    -d '{"command":"STATS",
"arguments":"TOTAL"}'                                       \
    | jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

exit

#
-----
-----

```

Delete the Cascading Environment Set Up Using REST API Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

#!/bin/bash

#
-----
-----
# --
# -- Stop & Delete Replicat
# --
#
-----
-----

curl -s -k -X PATCH https://west:9201/services/v2/replicats/
REPS                                     \
    -H "Content-Type: application/
json"                                     \
    -H "Accept: application/
json"                                     \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                   \
    -d '{"status":"stopped"}'           |
jq '.messages'

curl -s -k -X DELETE https://west:9201/services/v2/replicats/
REPS                                     \
    -H "Content-Type: application/
json"                                     \
    -H "Accept: application/
json"                                     \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

```

```

curl -s -k -X PATCH https://south:9101/services/v2/replicats/
REPN \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' |
jq '.messages'

```

```

curl -s -k -X DELETE https://south:9101/services/v2/replicats/
REPN \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

```

```

#

```

```

-----
# --
# -- Stop & Delete DistPath
# --
#
-----

```

```

curl -s -k -X PATCH https://south:9102/services/v2/sources/
DPSW \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"status":"stopped"}' | jq
'.messages'

```

```

curl -s -k -X DELETE https://south:9102/services/v2/sources/
DPSW \
    -H 'Content-Type: application/
json' \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

```

```

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic

```

```

Z2dtYTpHR2lhXzIzYWk='                                \
    -d '{"status":"stopped"}'                            |
jq '.messages'

curl -s -k -X DELETE https://north:9002/services/v2/sources/
DPNS                                                    \
    -H 'Content-Type: application/
json'                                                    \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='        |
jq '.messages'

#
-----
# --
# -- Stop & Delete Extracts
# --
#
-----

curl -s -k -X PATCH https://south:9101/services/v2/extracts/
EXTS                                                    \
    -H 'Content-Type: application/
json'                                                    \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                                    \
    -d '{"status":"stopped"}'                            |
jq '.messages'

curl -s -k -X DELETE https://south:9101/services/v2/extracts/
EXTS                                                    \
    -H "Content-Type: application/
json"                                                    \
    -H "Accept: application/
json"                                                    \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='        |
jq '.messages'

curl -s -k -X PATCH https://north:9001/services/v2/extracts/
EXTN                                                    \
    -H 'Content-Type: application/
json'                                                    \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='                                    \
    -d '{"status":"stopped"}'                            |
jq '.messages'

curl -s -k -X DELETE https://north:9001/services/v2/extracts/
EXTN                                                    \
    -H "Content-Type: application/
json"                                                    \
    -H "Accept: application/
json"                                                    \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='        |

```

```

jq '.messages'

#
-----
# --
# -- Delete Schematranda
# -- Delete Checkpointtable
# -- Delete Heartbeatables
# --
#
-----

# Delete SchemaTrandata at Database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"delete","schemaName":"hr"}'
jq '.messages'

# Delete SchemaTrandata at Database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"delete","schemaName":"hr"}'
jq '.messages'

# Delete Checkpointtable at Database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"delete","name":"ggadmin.ggs_checkpointtable"}'
jq '.messages'

# Delete Checkpointtable at Database GGWEST
curl -s -k -X POST https://south:9201/services/v2/connections/
OracleGoldenGate.ggwest/tables/checkpoint \

```

```

        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"operation":"delete","name":"ggadmin.ggs_checkpointtable"}' |
jq '.messages'

# Delete Heartbeatable at Database GGNORTH
curl -s -k -X DELETE https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

# Delete Heartbeatable at Database GGSOUTH
curl -s -k -X DELETE https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

# Delete Heartbeatable at Database GGWEST
curl -s -k -X DELETE https://west:9201/services/v2/connections/
OracleGoldenGate.ggwest/tables/heartbeat \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' |
jq '.messages'

#
-----
# --
# --Delete USERIDALIAS from GoldenGate
# --
#
-----

curl -s -k -X DELETE https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
        -H "Content-Type: application/
json" \
        -H "Accept: application/

```



```

json"
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.messages'
curl -s -k -X DELETE https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth
    -H "Content-Type: application/"
json"
    -H "Accept: application/"
json"
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.messages'

curl -s -k -X DELETE https://west:9201/services/v2/credentials/
OracleGoldenGate/ggwest
    -H "Content-Type: application/"
json"
    -H "Accept: application/"
json"
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
jq '.messages'

#
-----
# --
# -- Delete Trail Files
# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/commands/
execute
    -H "Content-Type: application/"
json"
    -H "Accept: application/"
json"
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
    -d '{"name": "purge"
        , "purgeType": "trails"
        , "trails": [{"name": "ea", "path": "north/"}]
        , "useCheckpoints": false
        , "keep": [{"type": "min", "units": "files", "value": 0}]
        }'
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/commands/
execute
    -H "Content-Type: application/"
json"
    -H "Accept: application/"
json"
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='

```

```

        -d '{"name": "purge"
            , "purgeType": "trails"
            , "trails": [{"name": "da", "path": "north/"}]
            , "useCheckpoints": false
            , "keep": [{"type": "min", "units": "files", "value": 0}]
          }'
jq '.messages'

curl -s -k -X POST https://south:9101/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
    , "purgeType": "trails"
    , "trails": [{"name": "ea", "path": "south/"}]
    , "useCheckpoints": false
    , "keep": [{"type": "min", "units": "files", "value": 0}]
  }'
jq '.messages'

curl -s -k -X POST https://west:9201/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
    , "purgeType": "trails"
    , "trails": [{"name": "da", "path": "south/"}]
    , "useCheckpoints": false
    , "keep": [{"type": "min", "units": "files", "value": 0}]
  }'
jq '.messages'

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

```

Cascading Topology Data Replication using Admin Client Commands

Create a Cascading Data Replication Configuration Using Admin Client Commands

Copy and use the following OBEY script to set up a cascaded Oracle GoldenGate data replication environment on an pre-installed database.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
--
-- Add Credentials
--

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate PASSWORD ggadmin

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate PASSWORD ggadmin

CONNECT https://west:9201 DEPLOYMENT depl_west AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbwest ALIAS ggwest DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE

--
-- Add Schematandata, Checkpointtable, and Heartbeatable
--

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD HEARTBEATTABLE

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

CONNECT https://west:9201 DEPLOYMENT depl_west AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggwest DOMAIN OracleGoldenGate
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

--
-- Add Extract and DistPath at source instance
--

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate

ADD EXTRACT extn INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT extn database
ADD EXTTRAIL north/ea, EXTRACT extn
START EXTRACT extn
```

```

ADD DISTPATH dpns SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://south:9103/services/v2/targets?trail=north/da !
START DISTPATH dpns

INFO ALL
INFO DISTPATH ALL

--
-- Add Replicat, Extract, and DistPath at intermediate instance
--

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate

ADD REPLICAT repn, PARALLEL, EXTTRAIL north/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
START REPLICAT repn

ADD EXTRACT exts INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT exts database
ADD EXTTRAIL south/ea, EXTRACT exts
START EXTRACT exts

ADD DISTPATH dpsw SOURCE trail://north:9102/services/v2/sources?
trail=south/ea TARGET wss://west:9203/services/v2/targets?trail=south/da !
START DISTPATH dpsw

INFO ALL
INFO DISTPATH ALL

--
-- Add Replicat at target instance
--

CONNECT https://west:9201 DEPLOYMENT depl_west AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggwest DOMAIN OracleGoldenGate

ADD REPLICAT reps, PARALLEL, EXTTRAIL south/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
START REPLICAT reps

INFO ALL
INFO DISTPATH ALL

DISCONNECT

```

After creating the OBEY file, create a shell script to run in Admin Client:

```

#!/bin/bash

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

# Copy parameter file (usually edited with EDIT PARAMs)

```

```
cp EXTN.prm /u01/app/oracle/deployments/depl_north/conf/ogg
cp REPN.prm /u01/app/oracle/deployments/depl_south/conf/ogg
cp EXTS.prm /u01/app/oracle/deployments/depl_south/conf/ogg
cp REPS.prm /u01/app/oracle/deployments/depl_west/conf/ogg

# Run the GoldenGate Obey script
echo "obey add_replication_cascading.oby" | adminclient
```

Check a Cascading Environment Replication from Source to Intermediate to Target Server

Copy and use the following OBEY script to check if the replicated data is transmitted from source (dbnorth) to intermediate (dbsouth) and intermediate to the target (dbwest) host.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/usr/bin/bash

sqlplus -s /nolog << EOF

set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 999999.99
col department_id   format 999

-- Connect to DBNorth
connect ggadmin/ggadmin@dbnorth

PROMPT
PROMPT Source Database:

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

-- Connect to DBSouth
connect ggadmin/ggadmin@dbsouth

PROMPT Intermediate Database:

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

-- Connect to DBWest
connect ggadmin/ggadmin@dbwest
```

PROMPT Target Database:

```
select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;
```

```
EOF
exit
```

Remove the Cascading Environment Set Up Using Admin Client Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
--
-- Remove objects from WEST
--

CONNECT https://west:9201 DEPLOYMENT depl_west  AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggwest DOMAIN OracleGoldenGate

KILL REPLICAT reps
DELETE REPLICAT reps

DELETE CHECKPOINTTABLE ggadmin.ggs_checkpointtable !
DELETE HEARTBEATABLE !

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbwest ALIAS ggwest DOMAIN
OracleGoldenGate

PURGE EXTTRAIL south/da

--
-- Remove objects from SOUTH
--

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate

KILL REPLICAT repn
DELETE REPLICAT repn

STOP DISTPATH dpsw
DELETE DISTPATH dpsw

KILL EXTRACT exts
DELETE EXTRACT exts
```

```
DELETE SCHEMATRANDATA hr
DELETE CHECKPOINTTABLE ggadmin.ggs_checkpointtable !
DELETE HEARTBEATTABLE !

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate

PURGE EXTTRAIL north/da
PURGE EXTTRAIL south/ea

--
-- Remove objects from NORTH
--

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate

STOP DISTPATH dpns
DELETE DISTPATH dpns

KILL    EXTRACT extn
DELETE EXTRACT extn

INFO ALL
INFO DISTPATH ALL

DELETE SCHEMATRANDATA hr
DELETE HEARTBEATTABLE !

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate

PURGE EXTTRAIL north/ea

DISCONNECT
```

After creating the OBEY file, create a shell script to run in Admin Client:

```
#!/bin/bash

# Run the GoldenGate Obey script
echo "obey delete_replication_cascading.oby" | adminclient

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null
```

Common Files and Scripts Used in Cascading Data Replication

Source Database DBNORTH DML Operation

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbnorth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 999999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

begin
  for i in 1..10 loop
    update hr.employees set salary = 1.05 * salary where department_id =60;
    dbms_lock.sleep(1);
    commit;
  end loop;
end;
/

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

EOF
exit
```

Intermediate Database DBSOUTH DML Operations

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbnorth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 999999.99
col department_id   format 999

drop table if exists hr.employees_test;
create table hr.employees_test as
select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

begin
  for i in 1..10 loop
```



```

        update hr.employees_test set salary = 100 + 1.05 * salary where
department_id =60;
        dbms_lock.sleep(1);
        commit;
    end loop;
end;
/

select employee_id, first_name, last_name, salary, department_id
from hr.employees_test
where department_id = 60;

EOF
exit

```

Check the Status and Statistics for Processes in a Cascaded Replication Environment

```

#!/bin/bash

#
-----
-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Extract EXN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Extract EXTN status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTN lag:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \

```

```

        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
        | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTN statistics:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"command": "STATS",
"arguments": "TOTAL"}' \
        | jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
"-----"
echo "--"
echo "-- DistPath DPNS"
echo "--"
echo
"-----"
"-----"
echo " "

echo "DistPath DPNS status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

curl -s -k -X GET https://north:9002/services/v2/sources/DPNS/
stats \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        | jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\

```

```

\t\t\t/g'| grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Replicat REPN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Replicat REPN status:"
curl -s -k -X GET https://south:9101/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat REPN lag:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g'| grep -v OKNODOT

echo "Replicat REPN statistics:"
curl -s -k -X POST https://south:9101/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g'| grep -v
OKNODOT

echo " "
echo
"-----"
-----"
echo "--"

```

```

echo "-- Extract EXS "
echo "--"
echo
"-----"
echo " "

echo "Extract EXTS status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTS \
-H "Content-Type: application/" \
-H "Accept: application/" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTS lag:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command \
-H "Content-Type: application/" \
-H "Accept: application/" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTS statistics:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command \
-H "Content-Type: application/" \
-H "Accept: application/" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
echo "--"
echo "-- DistPath DPSW"
echo "--"
echo
"-----"
echo " "

```

```

echo "DistPath DPSW status:"
curl -s -k -X GET https://south:9102/services/v2/sources/
DPSW \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

curl -s -k -X GET https://south:9102/services/v2/sources/DPSW/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Replicat REPW "
echo "--"
echo
"-----"
-----"
echo " "

echo "Replicat REPW status:"
curl -s -k -X GET https://west:9201/services/v2/replicats/
REPW \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat REPW lag:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPW/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPW statistics:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPW/

```

```

command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command":"STATS",
"arguments":"TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

exit

#
-----
-----

```

Select Statement for Source to Intermediate Transactions

```

#!/usr/bin/bash

sqlplus -s /nolog << EOF

set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary           format 999999.99
col department_id   format 999

-- Connect to DBNorth
connect ggadmin/ggadmin@dbnorth

PROMPT
PROMPT Source Database:

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

-- Connect to DBSouth
connect ggadmin/ggadmin@dbsouth

PROMPT Intermediate Database:

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

-- Connect to DBWest
connect ggadmin/ggadmin@dbwest

PROMPT Target Database:

```

```
select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Select Statement for Intermediate Database DBSOUTH

```
#!/usr/bin/bash

sqlplus -s ggadmin/ggadmin@dbsouth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Select Statement for Target Database DBWEST

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbsouth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
  from hr.employees
 where department_id = 60;

EOF
exit
```

Configure a Data Distribution Topology

Scripts are available in cURL and as OBEY commands to be run in a REST API client or Admin Client, to test the one to many data replication or the data distribution topology. You can use the scripts available in the following topics to:

- Add a data replication environment where the source PDB has an Extract process that captures from the source PDB and transmits trail data to the intermediate PDB which contains a Replicat and an Extract process. On the intermediate instance, the Replicat applies the trail data from source PDB and then the Extract captures the committed transactions and transmits trail data to the target PDB. See [Data Distribution](#) for a conceptual understanding of this type of topology.
- Add DML transactions on the source PDB to observe how Extract captures data and writes to trail.
- Check the statistical reports for Daily, Hourly, and Total committed DML and DDL operations.
- Delete the data replication environment and verify that the environment was deleted after testing.

Data Distribution Topology Data Replication using REST API Commands

Create a Broadcast Topology Based Data Replication Using REST API Commands

Copy and use the following cURL script to set up Oracle GoldenGate data replication environment on an pre-installed database and Oracle GoldenGate deployment.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/bin/bash

#
-----
-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

# Add UserIdAlias GGNORTH to connect to the Database instance DBNORTH
curl -s -k -X POST https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTphR2lhXzIzYWk=' \
```



```

-d '{"userid":"ggadmin@dbnorth","password":"ggadmin"}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/credentials/OracleGoldenGate/
ggnorth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
# --
# -- Add Schematranda
# -- Add Heartbeatable
# -- Add Checkpointtables
# --
#
-----

# Add Supplemental Logging to Database Schema HR (Schematranda) on source
database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","schemaName":"hr"}' | jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","schemaName":"hr"}' | jq '.response'

# Add Heartbeatable on source database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"frequency":60}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/connections/

```

```

OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.messages'

#
-----
# --
# -- Add Extracts on source database GGNORTH
# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/extracts/
EXTN \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Extract - Region North"
  ,"config":["EXTRACT EXTN"
    ,"USERIDALIAS ggnorth"
    ,"DDL INCLUDE MAPPED"
    ,"DDLOPTIONS REPORT"
    ,"REPORTCOUNT EVERY 10 MINUTES, RATE"
    ,"WARNLONGTRANS 15MINUTES, CHECKINTERVAL 5MINUTES"
    ,"EXTTRAIL north/ea"
    ,"TABLE hr.job_history_a;"
    ,"EXTTRAIL north/eb"
    ,"TABLE hr.job_history_b;"
    ,"EXTTRAIL north/ec"
    ,"TABLE hr.job_history_c;"
  ]
  ,"source": "tranlogs"
  ,"credentials":{"alias":"ggnorth"}
  ,"registration": {"optimized": false}
  ,"begin":"now"
  ,"targets":[{"name":"ea", "path":"north/"
    ,{"name":"eb", "path":"north/"
    ,{"name":"ec", "path":"north/"
  ]
  ,"status":"running"
}' | jq '.messages'

curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

```

```

#
-----
# --
# -- Add DistPath from source to target system
# --
#
-----

#ea: north -> A
curl -s -k -X POST https://north:9002/services/v2/sources/
DPNS_A \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description": "Region: From: North, To: A"
, "name": "DPNS_A"
, "source":
{ "uri": "trail://north:9002/services/v2/sources?trail=north/ea"
, "details": { "encryption": { "algorithm": "NONE" } }
}
, "target":
{ "uri": "wss://south:9103/services/v2/targets?trail=north/da"
, "authenticationMethod": { "certificate": "default" }
, "details":
{ "trail": { "sizeMB": 100 }
, "encryption": { "algorithm": "NONE" }
, "compression": { "enabled": false }
}
}
, "options":
{ "eofDelayCSecs": 10
, "checkpointFrequency": 10
, "critical": false
, "autoRestart": { "retries": 10, "delay": 2 }
, "streaming": true
}
, "begin": "now"
, "encryptionProfile": "LocalWallet"
, "status": "running"
}' | jq '.messages'

#eb: north -> B
curl -s -k -X POST https://north:9002/services/v2/sources/
DPNS_B \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description": "Region: From: North, To: B"

```

```

    , "name": "DPNS_B"
    , "source":
      { "uri": "trail://north:9002/services/v2/sources?trail=north/eb"
      , "details": { "encryption": { "algorithm": "NONE" } }
      }
    , "target":
      { "uri": "wss://south:9103/services/v2/targets?trail=north/db"
      , "authenticationMethod": { "certificate": "default" }
      , "details":
        { "trail": { "sizeMB": 100 }
        , "encryption": { "algorithm": "NONE" }
        , "compression": { "enabled": false }
        }
      }
    , "options":
      { "eofDelayCSecs": 10
      , "checkpointFrequency": 10
      , "critical": false
      , "autoRestart": { "retries": 10, "delay": 2 }
      , "streaming": true
      }
    , "begin": "now"
    , "encryptionProfile": "LocalWallet"
    , "status": "running"
  }' | jq '.messages'

curl -s -k -X POST https://north:9002/services/v2/sources/
DPNS_C \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{ "description": "Region: From: North, To: C"
, "name": "DPNS_C"
, "source":
  { "uri": "trail://north:9002/services/v2/sources?trail=north/ec"
  , "details": { "encryption": { "algorithm": "NONE" } }
  }
, "target":
  { "uri": "wss://south:9103/services/v2/targets?trail=north/dc"
  , "authenticationMethod": { "certificate": "default" }
  , "details":
    { "trail": { "sizeMB": 100 }
    , "encryption": { "algorithm": "NONE" }
    , "compression": { "enabled": false }
    }
  }
, "options":
  { "eofDelayCSecs": 10
  , "checkpointFrequency": 10
  , "critical": false
  , "autoRestart": { "retries": 10, "delay": 2 }
  , "streaming": true
  }
}
```

```

        ,"begin": "now"
        ,"encryptionProfile": "LocalWallet"
        ,"status": "running"
    }' | jq '.messages'

curl -s -k -X GET https://north:9002/services/v2/
sources \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS_A \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS_B \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS_C \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

```

Check the Statistics in a Broadcast Topology Based Data Replication Using REST API Commands

① Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/bin/bash
```

```
#
```

```
-----
```

```

# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

echo " "
echo
"-----"
-----"
echo "--"
echo "-- Extract EXN "
echo "--"
echo
"-----"
-----"
echo " "

echo "Extract EXTN status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTN lag:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTN statistics:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

```

```

echo " "
echo
"-----"
-----"
echo "--"
echo "-- DistPath DPNS"
echo "--"
echo
"-----"
-----"
echo " "

echo "DistPath DPNS_A status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS_A \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "DistPath DPNS_A statistics:"
curl -s -k -X GET https://north:9002/services/v2/sources/DPNS_A/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo "DistPath DPNS_B status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS_B \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "DistPath DPNS_B statistics:"
curl -s -k -X GET https://north:9002/services/v2/sources/DPNS_B/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

```

```

echo "DistPath DPNS_C status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNS_C \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "DistPath DPNS_C statistics:"
curl -s -k -X GET https://north:9002/services/v2/sources/DPNS_C/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

exit

#
-----
-----

```

Remove Broadcast Topology Setup Using REST API Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

#!/bin/bash

#
-----

# --
# -- Stop & Delete DistPath
# -- Purge Target Trail File
# --
#
-----

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS_A \
-H "Content-Type: application/

```



```

json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS_B \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS_C \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X PATCH https://north:9002/services/v2/sources/
DPNS_C \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X DELETE https://north:9002/services/v2/sources/
DPNS_A \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

curl -s -k -X DELETE https://north:9002/services/v2/sources/
DPNS_B \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

curl -s -k -X DELETE https://north:9002/services/v2/sources/
DPNS_C \
-H 'Content-Type: application/
json' \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/

```

```

json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
  , "purgeType": "trails"
  , "trails": [{"name": "da", "path": "north/"}]}
  , "useCheckpoints": false
  , "keep": [{"type": "min", "units": "files", "value": 0}]
}' | jq
'.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
  , "purgeType": "trails"
  , "trails": [{"name": "db", "path": "north/"}]}
  , "useCheckpoints": false
  , "keep": [{"type": "min", "units": "files", "value": 0}]
}' | jq
'.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
  , "purgeType": "trails"
  , "trails": [{"name": "dc", "path": "north/"}]}
  , "useCheckpoints": false
  , "keep": [{"type": "min", "units": "files", "value": 0}]
}' | jq
'.messages'

#
-----
# --
# -- Stop & Delete Extracts
# --
#
-----

curl -s -k -X PATCH https://north:9001/services/v2/extracts/
EXTN \
-H 'Content-Type: application/

```

```

json'
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
-d '{"status": "stopped"}'
'.messages'

curl -s -k -X DELETE https://north:9001/services/v2/extracts/
EXTN
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
'.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
-d '{"name": "purge"
, "purgeType": "trails"
, "trails": [{"name": "ea", "path": "north/"}]}
, "useCheckpoints": false
, "keep": [{"type": "min", "units": "files", "value": 0}]
}'
'.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
-d '{"name": "purge"
, "purgeType": "trails"
, "trails": [{"name": "eb", "path": "north/"}]}
, "useCheckpoints": false
, "keep": [{"type": "min", "units": "files", "value": 0}]
}'
'.messages'

curl -s -k -X POST https://north:9001/services/v2/commands/
execute
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='

```

```

        -d '{"name": "purge"
            , "purgeType": "trails"
            , "trails": [{"name": "ec", "path": "north/"}]
            , "useCheckpoints": false
            , "keep": [{"type": "min", "units": "files", "value": 0}]
            }'
'.messages'

#
-----
-----

# --
# -- Delete Schematranda
# -- Delete Checkpointtable
# -- Delete Heartbeatables
# --
#
-----
-----

# Delete SchemaTrandata at Database GGNORTH
curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation": "delete", "schemaName": "hr"}'
'.messages'

# Delete Heartbeatable at Database GGNORTH
curl -s -k -X DELETE https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
'.messages'

#
-----
-----

# --
# --Delete USERIDALIAS from GoldenGate
# --
#
-----
-----

curl -s -k -X DELETE https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
-H "Content-Type: application/

```

```

json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

curl -s -k -X GET https://north:9001/services/v2/credentials/
OracleGoldenGate \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

#
-----
-----

```

Data Distribution Topology Data Replication using Admin Client Commands

Create a Broadcast Topology Based Data Replication Using Admin Client Commands

Copy and use the following OBEY script to set up Oracle GoldenGate data replication environment on a pre-installed database.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

--
-- Connect the GoldenGate Deployment depl_north
--
CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !

ALTER CREDENTIALSTORE ADD USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD HEARTBEATTABLE

ADD EXTRACT extn INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT extn database
ADD EXTTRAIL north/ea, EXTRACT extn
START EXTRACT extn

ADD DISTPATH dpns_a SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://south:9103/services/v2/targets?trail=north/da !

```

```

ADD DISTPATH dpns_b SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://south:9103/services/v2/targets?trail=north/db !
ADD DISTPATH dpnw_a SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://west:9203/services/v2/targets?trail=north/da !
ADD DISTPATH dpnw_b SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://west:9203/services/v2/targets?trail=north/db !

START DISTPATH dpns_a
START DISTPATH dpns_b
START DISTPATH dpnw_a
START DISTPATH dpnw_b

DISCONNECT

```

After creating the OBEY file, create a shell script to run in Admin Client. This script runs triggers the bidirectional replication setup using the .oby file:

```

#!/bin/bash

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

# Copy parameter file (usually edited with EDIT PARAMS)
cp EXTN.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/

# Run the GoldenGate Obey script
echo "obey add_replication_datadistribution.oby" | adminclient

```

Check the Statistics in a Broadcast Topology Based Data Replication Using Admin Client Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

--
-- Connect the GoldenGate Deployment depl_north
--
CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !

INFO EXTRACT extn
SEND EXTRACT extn, GETLAG
STATS EXTRACT extn, TOTAL, TOTALSONLY *.*

INFO DISTPATH ALL
STATS DISTPATH dpns_a
STATS DISTPATH dpns_b

```

```
STATS DISTPATH dpnw_a
STATS DISTPATH dpnw_b
```

After creating the OBEY file, create a shell script to run in Admin Client:

```
echo "obey check_replication_datadistribution.oby" | adminclient

ls -ltrR /u01/app/oracle/deployments/depl_north/var/lib/data/north/
ls -ltrR /u01/app/oracle/deployments/depl_south/var/lib/data/north/
ls -ltrR /u01/app/oracle/deployments/depl_west/var/lib/data/north/
```

Remove the Broadcast Topology Setup Using Admin Client Commands

① Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
-- Connect the GoldenGate Deployment depl_north
--

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !

DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
KILL EXTRACT extn
DELETE EXTRACT extn

INFO ALL

STOP DISTPATH dpns_a
STOP DISTPATH dpns_b
STOP DISTPATH dpnw_a
STOP DISTPATH dpnw_b

DELETE DISTPATH dpns_a
DELETE DISTPATH dpns_b
DELETE DISTPATH dpnw_a
DELETE DISTPATH dpnw_b

INFO DISTPATH ALL

ALTER CREDENTIALSTORE DELETE USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate
INFO CREDENTIALSTORE

PURGE EXTTRAIL north/ea

--
-- Connect the GoldenGate Deployment depl_south
--
```

```
CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !

PURGE EXTTRAIL north/da
PURGE EXTTRAIL north/db
```

After creating the OBEY file, create a shell script to run in Admin Client:

```
#!/bin/bash

# Run the GoldenGate Obey script
echo "obey delete_replication_datadistribution.oby" | adminclient

ls -ltrR /u01/app/oracle/deployments/depl_north/var/lib/data/north/
ls -ltrR /u01/app/oracle/deployments/depl_south/var/lib/data/north/
ls -ltrR /u01/app/oracle/deployments/depl_west/var/lib/data/north/

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null
```

Common Files and Scripts Used in Data Distribution Topology

DBNORTH DML Operations

```
#!/usr/bin/bash

sqlplus -s system/oracle4GG@dbnorth << EOF
set lines 100
col first_name      format a30
col last_name       format a30
col employee_id     format 9999
col salary          format 99999.99
col department_id   format 999

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

begin
  for i in 1..10 loop
    update hr.employees set salary = 1.05 * salary where department_id =60;
    dbms_lock.sleep(1);
    commit;
  end loop;
end;
/

select employee_id, first_name, last_name, salary, department_id
   from hr.employees
  where department_id = 60;

EOF
exit
```


Data Replication in a Data Consolidation Topology Configuration

Scripts are available in cURL and as OBEY commands to be run in a REST API client or Admin Client, to test the one to many data replication or the data distribution topology. You can use the scripts available in the following topics to:

- See [Consolidation](#) for a conceptual understanding of this type of topology.
- Add DML transactions on the source PDB to observe how Extract captures data and writes to trail.
- Check the statistical reports for Daily, Hourly, and Total committed DML and DDL operations.
- Delete the data replication environment and verify that the environment was deleted after testing.

Data Replication in a Data Consolidation Topology Using REST API

Create a Consolidation Topology Based Data Replication Using REST API Commands

Copy and use the following cURL script to set up Oracle GoldenGate data replication environment on a pre-installed Oracle Database 23ai Free and Oracle GoldenGate deployment.

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```
#!/bin/bash

#
-----
-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

# Add UserIdAlias GGSOUTH to connect to the Database instance DBSOUTH
curl -s -k -X POST https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
```

```

json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"userid":"ggadmin@dbsouth","password":"ggadmin"}' | jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/credentials/OracleGoldenGate/
ggsouth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
# --
# -- Add Checkpointtables
# -- Add Heartbeattable
# --
#
-----

# Add Checkpointtable on target database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"add","name":"ggadmin.ggs_checkpointtable"}' | jq
'.messages'

curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"operation":"info","name":"ggadmin.ggs_checkpointtable"}' | jq
'.messages'

# Add Heartbeattable on target database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \

```

```

-d '{"frequency":60}' | jq '.messages'

curl -s -k -X GET https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
# --
# -- Add Replicat at target Database GGSOUTH
# --
#
-----

# 1st Replicat: REPNA

curl -s -k -X POST https://south:9101/services/v2/replicats/
REPNA \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"description":"Replicat - Region South (Consuming from A)"
, "config":["REPLICAT repna"
, "USERIDALIAS ggsouth DOMAIN OracleGoldenGate"
, "DDL INCLUDE MAPPED"
, "DDLOPTIONS REPORT"
, "DDLERROR DEFAULT, DISCARD"
, "REPORTCOUNT EVERY 10 MINUTES, RATE"
, "REPEROR (DEFAULT, DISCARD)"
, "MAP hr.*, TARGET hr.*;"
]
, "credentials": {"alias": "ggsouth"}
, "mode": {"parallel":true, "type": "nonintegrated"}
, "source": {"name": "da", "path": "north"}
, "checkpoint":{"table": "ggadmin.ggs_checkpointtable"}
, "status": "running"
}' | jq '.messages'

# 2nd Replicat: REPNB

curl -s -k -X POST https://south:9101/services/v2/replicats/
REPNB \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic

```

```

Z2dtYTpHR2lhXzIzYWk='
  -d '{"description":"Replicat - Region South (Consuming from B)"
    , "config":["REPLICAT repnb"
      , "USERIDALIAS ggsouth DOMAIN OracleGoldenGate"
      , "DDL INCLUDE MAPPED"
      , "DDLOPTIONS REPORT"
      , "DDLERROR DEFAULT, DISCARD"
      , "REPORTCOUNT EVERY 10 MINUTES, RATE"
      , "REPEROR (DEFAULT, DISCARD)"
      , "MAP hr.*, TARGET hr.*;"
    ]
    , "credentials": {"alias": "ggsouth"}
    , "mode": {"parallel":true,"type": "nonintegrated"}
    , "source": {"name": "da", "path": "north"}
    , "checkpoint":{"table": "ggadmin.ggs_checkpointtable"}
    , "status": "running"
  }' | jq '.messages'

# 3rd Replicat: REPNC

curl -s -k -X POST https://south:9101/services/v2/replicats/
REPNC
  -H "Content-Type: application/
json"
  -H "Accept: application/
json"
  -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk='
  -d '{"description":"Replicat - Region South (Consuming from C)"
    , "config":["REPLICAT repnc"
      , "USERIDALIAS ggsouth DOMAIN OracleGoldenGate"
      , "DDL INCLUDE MAPPED"
      , "DDLOPTIONS REPORT"
      , "DDLERROR DEFAULT, DISCARD"
      , "REPORTCOUNT EVERY 10 MINUTES, RATE"
      , "REPEROR (DEFAULT, DISCARD)"
      , "MAP hr.*, TARGET hr.*;"
    ]
    , "credentials": {"alias": "ggsouth"}
    , "mode": {"parallel":true,"type": "nonintegrated"}
    , "source": {"name": "da", "path": "north"}
    , "checkpoint":{"table": "ggadmin.ggs_checkpointtable"}
    , "status": "running"
  }' | jq '.messages'

# List all Replicats
curl -s -k -X GET https://south:9101/services/v2/
replicats
  -H "Content-Type: application/
json"
  -H "Accept: application/
json"
  -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

# List each individual Replicat (REPNA - REPND):
curl -s -k -X GET https://south:9101/services/v2/replicats/

```

```

REPNA                                \
    -H "Content-Type: application/
json"                                \
    -H "Accept: application/
json"                                \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

curl -s -k -X GET https://south:9101/services/v2/replicats/
REPNB                                \
    -H "Content-Type: application/
json"                                \
    -H "Accept: application/
json"                                \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

curl -s -k -X GET https://south:9101/services/v2/replicats/
REPNC                                \
    -H "Content-Type: application/
json"                                \
    -H "Accept: application/
json"                                \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response'

#
-----
-----

```

Check the Statistics in a Consolidation Topology Based Data Replication Using REST API Commands

① Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

#!/bin/bash

#
-----
-----

# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

echo " "
echo
"-----
-----"

```

```

echo "---"
echo "--- Extract EXN "
echo "---"
echo
"-----"
echo " "

echo "Extract EXTN status:"
curl -s -k -X GET https://north:9001/services/v2/extracts/
EXTN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTN lag:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTN statistics:"
curl -s -k -X POST https://north:9001/services/v2/extracts/EXTN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
echo "---"
echo "--- DistPath DPNW"
echo "---"
echo
"-----"
echo " "

```

```

echo "DistPath DPNW status:"
curl -s -k -X GET https://north:9002/services/v2/sources/
DPNW \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "DistPath DPNW statistics:"
curl -s -k -X GET https://north:9002/services/v2/sources/DPNW/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "---"
echo "--- Extract EXTS "
echo "---"
echo
"-----"
-----"
echo " "

echo "Extract EXTS status:"
curl -s -k -X GET https://south:9101/services/v2/extracts/
EXTS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Extract EXTS lag:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply'

```

```

        | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Extract EXTS statistics:"
curl -s -k -X POST https://south:9101/services/v2/extracts/EXTS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command":"STATS",
"arguments":"TOTAL"}' \
| jq '.response.reply' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

echo " "
echo
"-----"
-----"
echo "--"
echo "-- DistPath DPSW"
echo "--"
echo
"-----"
-----"
echo " "

echo "DistPath DPSW status:"
curl -s -k -X GET https://south:9102/services/v2/sources/
DPSW \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "DistPath DPSW statistics:"
curl -s -k -X GET https://south:9102/services/v2/sources/DPSW/
stats \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
| jq '.response' | json_reformat | sed 's/\\n/\\n/g' | sed 's/\\
\\t/\\t/g' | grep -v OKNODOT

ls -ltrR /u01/app/oracle/deployments/depl_north/var/lib/data/north
ls -ltrR /u01/app/oracle/deployments/depl_south/var/lib/data/south

echo " "
echo

```



```

"-----"
-----"
echo "--"
echo "-- Replicats REPN and REPS "
echo "--"
echo
"-----"
-----"
echo " "

echo "Replicat REPN status:"
curl -s -k -X GET https://west:9201/services/v2/replicats/
REPN \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat REPN lag:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPN statistics:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPN/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

echo "Replicat REPS status:"
curl -s -k -X GET https://west:9201/services/v2/replicats/
REPS \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq '.response.status'

echo "Replicat RESN lag:"

```

```

curl -s -k -X POST https://west:9201/services/v2/replicats/REPS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "GETLAG", "isReported": true}' | jq
'.response.reply' \
| sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v OKNODOT

echo "Replicat REPS statistics:"
curl -s -k -X POST https://west:9201/services/v2/replicats/REPS/
command \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"command": "STATS",
"arguments": "TOTAL"}' \
| jq '.response.reply' | sed 's/\\n/\\n/g' | sed 's/\\t/\\t/g' | grep -v
OKNODOT

ls -ltrR /u01/app/oracle/deployments/depl_west/var/lib/data/north
ls -ltrR /u01/app/oracle/deployments/depl_west/var/lib/data/south
exit

```

Remove the Consolidation Topology Replication Setup Using REST API Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

#!/bin/bash

#
-----
# --
# -- stop & Delete Replicat
# --
#
-----

curl -s -k -X PATCH https://south:9101/services/v2/replicats/
REPNA \

```

```

        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X PATCH https://south:9101/services/v2/replicats/
REPNB \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X PATCH https://south:9101/services/v2/replicats/
REPNC \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
        -d '{"status":"stopped"}' | jq
'.messages'

curl -s -k -X DELETE https://south:9101/services/v2/replicats/
REPNA \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

curl -s -k -X DELETE https://south:9101/services/v2/replicats/
REPNB \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'
curl -s -k -X DELETE https://south:9101/services/v2/replicats/
REPNC \
        -H "Content-Type: application/
json" \
        -H "Accept: application/
json" \
        -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

```

```

curl -s -k -X POST https://south:9101/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
, "purgeType": "trails"
, "trails": [{"name": "ea", "path": "north/"}]}
, "useCheckpoints": false
, "keep": [{"type": "min", "units": "files", "value": 0}]
}' | jq
'.messages'
curl -s -k -X POST https://south:9101/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
, "purgeType": "trails"
, "trails": [{"name": "da", "path": "north/"}]}
, "useCheckpoints": false
, "keep": [{"type": "min", "units": "files", "value": 0}]
}'

curl -s -k -X POST https://south:9101/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
, "purgeType": "trails"
, "trails": [{"name": "db", "path": "north/"}]}
, "useCheckpoints": false
, "keep": [{"type": "min", "units": "files", "value": 0}]
}'

curl -s -k -X POST https://south:9101/services/v2/commands/
execute \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
-d '{"name": "purge"
, "purgeType": "trails"

```

```

        , "trails": [{"name": "dc", "path": "north/"}]
        , "useCheckpoints": false
        , "keep": [{"type": "min", "units": "files", "value": 0}]
    }'

#
-----
-----

# --
# -- Delete Schematranda
# -- Delete Checkpointtable
# -- Delete Heartbeattables
# --
#
-----
-----

# Delete Checkpointtable at Database GGSOUTH
curl -s -k -X POST https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/checkpoint \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic
Z2dtYTpHR2lhXzIzYWk=' \
    -d '{"operation": "delete", "name": "ggadmin.ggs_checkpointtable"}' | jq
'.messages'

# Delete Heartbeattable at Database GGSOUTH
curl -s -k -X DELETE https://south:9101/services/v2/connections/
OracleGoldenGate.ggsouth/tables/heartbeat \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

#
-----
-----

# --
# --Delete USERIDALIAS from GoldenGate
# --
#
-----
-----

curl -s -k -X DELETE https://south:9101/services/v2/credentials/
OracleGoldenGate/ggsouth \
    -H "Content-Type: application/
json" \
    -H "Accept: application/
json" \
    -H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk=' | jq
'.messages'

```

```

curl -s -k -X GET https://south:9101/services/v2/
credentials
-H "Content-Type: application/
json"
-H "Accept: application/
json"
-H 'Authorization: Basic Z2dtYTpHR2lhXzIzYWk='
'.messages'

#
-----
-----

```

Data Replication in a Data Consolidation Topology Using Admin Client

Create a Consolidation Topology Based Data Replication Using Admin Client Commands

Copy and use the following OBEY script to set up Oracle GoldenGate data replication environment on an pre-installed database.

① Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

```

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbnorth ALIAS ggnorth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE

```

```

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
ALTER CREDENTIALSTORE ADD USER ggadmin@dbsouth ALIAS ggsouth DOMAIN
OracleGoldenGate PASSWORD ggadmin
INFO CREDENTIALSTORE

```

```

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

```

```

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate
ADD SCHEMATRANDATA hr
ADD CHECKPOINTTABLE ggadmin.ggs_checkpointtable
ADD HEARTBEATTABLE

```

```

CONNECT https://north:9001 DEPLOYMENT depl_north AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggnorth DOMAIN OracleGoldenGate

ADD EXTRACT extn INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT extn database
ADD EXTTRAIL north/ea, EXTRACT extn
START EXTRACT extn

ADD DISTPATH dpns SOURCE trail://north:9002/services/v2/sources?
trail=north/ea TARGET wss://south:9103/services/v2/targets?trail=north/da !
START DISTPATH dpns

ADD REPLICAT reps, PARALLEL, EXTTRAIL south/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
START REPLICAT reps

INFO ALL
INFO DISTPATH ALL

CONNECT https://south:9101 DEPLOYMENT depl_south AS ggma PASSWORD GGma_23ai !
DBLOGIN USERIDALIAS ggsouth DOMAIN OracleGoldenGate

ADD EXTRACT exts INTEGRATED TRANLOG BEGIN NOW
REGISTER EXTRACT exts database
ADD EXTTRAIL south/ea, EXTRACT exts
START EXTS

ADD DISTPATH dpsn SOURCE trail://south:9102/services/v2/sources?
trail=south/ea TARGET wss://north:9103/services/v2/targets?trail=south/da !
START DISTPATH dpsn

ADD REPLICAT repn, PARALLEL, EXTTRAIL north/da, CHECKPOINTTABLE
ggadmin.ggs_checkpointtable
START REPLICAT repn

INFO ALL
INFO DISTPATH ALL

DISCONNECT

```

After creating the OBEY file, create a shell script to run in Admin Client. This script runs triggers the bidirectional replication setup using the .oby file:

```

#!/bin/bash

# Clean up environment
/home/oracle/scripts/misc/cleanup.sh > /dev/null

# Copy parameter file (usually edited with EDIT PARAMs)
cp EXTN.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/
cp REPS.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/

cp EXTS.prm /u01/app/oracle/deployments/depl_north/etc/conf/ogg/

```

```
cp REPN.prm /u01/app/oracle/deployments/depl_south/etc/conf/ogg/  
  
# Run the GoldenGate Obey script  
echo "obey add_replication_ActiveActive.oby" | adminclient | tee  
add_replication_ActiveActive.log
```

Check Processes and Statistics in a Consolidation Topology Based Data Replication Using Admin Client Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

After creating the OBEY file, create a shell script to run in Admin Client:

Remove the Consolidation Topology Based Data Replication Using Admin Client Commands

Note

The given sample script uses names and values of database server, parameter values, Extract, Replicat, and other processes. You must change these values according to your environment for this script to work.

After creating the OBEY file, create a shell script to run in Admin Client:

Data Streams Script

Use this section to copy and test the automation scripts for scenarios where you would need to set up Oracle GoldenGate Data Streams.

The following script performs the following tasks:

- Creates a `USERIDALIAS` to connect Oracle GoldenGate to the database
- Adds `TRANSDATA`
- Creates an Extract

- Adds a user for the Data Stream
- Adds a Data Stream

Data Streams Sample Script

Copy and use the following cURL script to test an Oracle GoldenGate Data Streams environment.

```
#---
-----
-----
# --
# -- Create USERIDALIAS to connection from GoldenGate to the Databases
# --
#
-----
-----

curl -s -k -X POST https://north:9001/services/v2/credentials/
OracleGoldenGate/ggnorth \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
*****' \
-d '{"userid":"ggadmin@dbnorth","password":"***"}' | jq '.messages'

#
-----
-----
# --
# -- Add Schematranda/Trandata
# -- Note that the JSON DV and JCT are explicitly called for trandata
# --
#
-----
-----

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/schema \
-H "Content-Type: application/
json" \
-H "Accept: application/
json" \
-H 'Authorization: Basic
*****' \
-d '{"operation":"add","schemaName":"hr"}' | jq '.messages'

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/table \
-H "Content-Type: application/
json" \
```

```

        -H "Accept: application/
json"
        -H 'Authorization: Basic
*****'
        -d '{"operation":"add", "tableName":"hr.students_dv"}' | jq '.response'

curl -s -k -X POST https://north:9001/services/v2/connections/
OracleGoldenGate.ggnorth/trandata/table
        -H "Content-Type: application/
json"
        -H "Accept: application/
json"
        -H 'Authorization: Basic
*****'
        -d '{"operation":"add", "tableName":"hr.demo_jct"}' | jq '.response'

#
-----
# --
# -- Add Extracts on source database GGNORTH
# -- Note that the wildcard within the TABLE parameter captures
# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/extracts/
EXTN
        -H 'Content-Type: application/
json'
        -H 'Authorization: Basic
*****'
        -d '{"description":"Extract - Region North"
            , "config":["EXTRACT EXTN"
                        , "USERIDALIAS ggnorth"
                        , "EXTTRAIL north/ea"
                        , "DDL INCLUDE MAPPED"
                        , "DDLOPTIONS REPORT"
                        , "REPORTCOUNT EVERY 10 MINUTES, RATE"
                        , "WARNLONGTRANS 15MINUTES, CHECKINTERVAL 5MINUTES"
                        , "TABLE hr.*;"
                        ]
            , "source": "tranlogs"
            , "credentials":{"alias":"ggnorth"}
            , "registration": {"optimized": false}
            , "begin":"now"
            , "targets":[{"name":"ea", "path":"north/"}]
            , "status":"running"
            }' | jq '.messages'

#
-----
# --
# -- Add User for DataStream

```

```

# --
#
-----

curl -s -k -X POST https://north:9001/services/v2/authorizations/Operator/
dsadmin \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
*****' \
-d '{"credential":"****", "info":"Dedicated DataStream user"}' | jq
'.response'

#
-----

# --
# -- Add DataStream
# --
#
-----

curl -s -k -X POST https://north:9002/services/v2/stream/
DS01 \
-H 'Content-Type: application/
json' \
-H "Accept: application/
json" \
-H 'Authorization: Basic
*****' \
-d '{"source":{"trail":"ea"
, "path":"north"
}
, "cloudEventsFormat":false
, "encoding":"json"
, "bufferSize":1048576
, "qualityOfService":"exactlyOnce"
, "description":"Data Stream Test #01"
}' | jq '.response'

#
-----

```