

# Oracle GoldenGate Best Practices: Instantiation from an Oracle Source Database

Document ID 1276058.1





## DISCLAIMER

*This document touches briefly on many important and complex concepts and does not provide a detailed explanation of any one topic since the intent is to present the material in the most expedient manner. The goal is simply to help the reader become familiar enough with the product to successfully design and implement an Oracle GoldenGate environment. To that end, it is important to note that the activities of design, unit testing and integration testing which are crucial to a successful implementation have been intentionally left out of the guide. All the sample scripts are provided as is. Oracle consulting service is highly recommended for any customized implementation.*

*Any sample code provided here is for educational purposes only and not supported by Oracle Support Services. It has been tested internally, however, and works as documented. We do not guarantee that it will work for you, so be sure to test it in your environment before relying on it.*

*Proofread the sample code before using it! Due to the differences in the way text editors, e-mail packages and operating systems handle text formatting (spaces, tabs and carriage returns), the sample code may not be in an executable state when you first receive it. Check over the sample code to ensure that errors of this type are corrected.*

<b>Introduction</b>	<b>1</b>
Audience	1
Scope	1
<b>Prerequisites</b>	<b>2</b>
Database requirements	2
Oracle Net services requirements	2
Oracle GoldenGate requirements	2
<b>Overview</b>	<b>3</b>
Homogenous Instantiation	3
Heterogenous Instantiation	3
Oracle GoldenGate Highlights	3
<b>Oracle GoldenGate Token – LOGCSN</b>	<b>4</b>
<b>Delivery Process (Replicat) Use of LOGCSN</b>	<b>5</b>
<b>Oracle Data Pump Integration for Table Instantiation</b>	<b>5</b>
<b>Item To Review Before Starting Instantiation</b>	<b>7</b>
<b>Homogeneous Instantiation Methods</b>	<b>8</b>
Cloning with Oracle RMAN	8
<b>Source Setup</b>	<b>8</b>
Install Oracle GoldenGate	8
Create DB user for OGG Extract	9
Create GG UserID Alias	10
Enable Supplemental Logging at the Schema or Table level	11
Create and Start Extract Process on the Source Database	12
Create Distribution Service	13
Start OGG Processes before Target Instantiation	14
RMAN Tasks on the Source	15
RMAN Tasks on the Target	15
<b>Target Setup</b>	<b>17</b>
Install Oracle GoldenGate	17
Create DB User for OGG Replicat	17
Create GG UserID alias	18
Login to Target DB and create Checkpoint table	18
Create Replicat Process on Target Database	19
Start the Replicat Process	20



<b>Notes for Oracle RAC and ASM</b>	<b>20</b>
<b>Additional Information on RMAN approach</b>	<b>20</b>
<b>Oracle Data Guard</b>	<b>20</b>
<b>Oracle Snapshot Standby</b>	<b>21</b>
At Physical Standby	21
At the Target environment	23
<b><i>Heterogeneous Instantiation Methods</i></b>	<b>24</b>
<b>Transportable Tablespaces</b>	<b>25</b>
<b>GoldenGate Initial Load (Oracle to Any)</b>	<b>25</b>
<b>Oracle Data Pump</b>	<b>25</b>
<b>Example 1: Full Export Using Oracle Data Pump with “flashback_scn” option</b>	<b>27</b>
OGG Configuration on the Source System	27
OGG Configuration on the Target System	27
Start OGG Processes	27
Data Pump Export (Full) on Source System	27
Data Pump Import on Target System	28
Start Replicat on Target System	28
<b>Example 2: Full Export Using Oracle Data Pump &amp; OGG Instantiation Filtering Feature</b>	<b>29</b>
OGG Configuration on the Source System	29
OGG Configuration on the Target System	29
Start OGG Processes	30
Data Pump Export (Full) on Source System	30
Data Pump Import on Target System	31
Start Replicat on Target System	31
<b><i>Where To Go for More Information</i></b>	<b>32</b>
<b><i>Appendix – A: TNS Information</i></b>	<b>33</b>



## Introduction

Oracle GoldenGate provides data capture from transaction logs and delivery for heterogeneous databases and messaging systems. It provides a flexible, de-coupled architecture that can be used to implement virtually all replication scenarios.

One of the primary challenges in setting up a target database for replication is ensuring that change data can be applied without causing collisions such as reapplying data that was already copied after it was captured and avoiding data loss during the application of change data following the initial instantiation. Both risks can compromise data integrity if not handled carefully.

Oracle GoldenGate (OGG) addresses these concerns by enabling change data to be applied starting from an appropriate Commit Sequence Number (CSN), which helps prevent both collisions and data loss. It's important to note that OGG uses "CSN" instead of "SCN" (System Change Number); while functionally equivalent, they differ syntactically. CSN is an identifier that OGG constructs to identify a transaction for the purpose of maintaining transactional consistency and data integrity and it uniquely identifies a point in time in which a transaction commits to the database. SCN is Oracle's internal clock for identifying a specific point in time within the database and every commit increments the SCN.

To leverage this capability without incurring data loss, users must perform the target instantiation from a read-consistent image of the source database and capture the corresponding CSN of that consistent image.

This document provides further guidance and examples on how to use LOGCSN, which is recorded in Oracle GoldenGate Trail Files. This document is also an introduction to Oracle GoldenGate's best practices and guidelines for instantiation of a target database from an Oracle Source Database.

## Audience

The target audience for this best practice guide includes but is not limited to GoldenGate administrators, field consultants, database administrators, Oracle developers, solution architects who wanted to instantiate a target database from an Oracle source database. A working knowledge of Oracle RDBMS Database, Oracle GoldenGate Microservices Architecture and operating system are assumed.

## Scope

The document applies to the following products and versions:

- **Oracle GoldenGate Microservices Architecture 23ai** (23.5 and newer)
- **Oracle Database Enterprise Edition 19c** (does not include Autonomous Database)

Instantiating from and to an Autonomous Database (ADB) will be covered in a separate document.

## Prerequisites

### Database requirements

- The source and target databases must include the following parameter enabled.  
`alter system set ENABLE_GOLDENGATE_REPLICATION = TRUE SCOPE=BOTH;`
- The source must be in archive log mode.  
`alter database archive log;`
- All databases must be in force logging mode.  
`alter database force logging;`
- Source database must have minimal supplemental logging enabled.  
`alter database add supplemental log data;`

### Oracle Net services requirements

- The Oracle Net services configured for all the databases and working.  
See Appendix-A for the sample TNS configurations.

The Oracle Net services are configured so GoldenGate instances can communicate between the source and target databases.


Ensure that the necessary ports are opened between the systems for the source and target databases.

### Oracle GoldenGate requirements

Oracle GoldenGate 23ai can be installed in a mesh or hub configuration. The following table outlines items referenced throughout the document that is specific to GoldenGate Microservices architecture. Installation-specific values must be identified and substituted accordingly.

Item	Reference	Variable	Description
GoldenGate Home	<ogg_install_location>	OGG_HOME	Directory of the GoldenGate installation.
OGG configuration files	/<ogg deployment location>/etc/conf	OGG_CONF_HOME	Location in which each deployment information and configuration artifacts are stored.
Deployment data home	/<ogg_dep_loc>/var/lib/data	OGG_DATA_HOME	Location in which each deployment data (trail files) are stored.
Deployment variable home	/<ogg_dep_loc>/var	OGG_VAR_HOME	Location in which each deployment logging and reporting artifacts are stored.
Deployment etc Home	/<ogg_dep_loc>/etc	OGG_ETC_HOME	Location in which each deployment configuration files are stored including parameter files.

Once the source and target GoldenGate are installed, the host details of both GG systems should be updated in the /etc/hosts across both systems. Please see the Microservices Architecture documentation for [securing the deployments using certificates](#). If the setup



involves two deployments across different systems, follow the section [“Connecting Two Deployments using External RootCA Certificate”](#) from the documentation.

## Overview

This document will address different approaches to Instantiation of a target database. All approaches will have similar steps for the OGG configuration. Only the instantiation methods will differ.

### Homogenous Instantiation

In this approach, the Oracle Database and the operating system version are the same between the source and the target databases. There are a few Oracle methods that can be used to create the target database in a homogeneous environment. Each method can be researched in more depth in the appropriate Oracle documentation guide. However, this document will touch upon most applications.

### Heterogenous Instantiation

In this approach, the source and target are of different configurations. The difference could be as simple as a database version or an OS version. Or the difference could be a completely different database on a different platform, e.g., Oracle on Linux to Microsoft SQL Server on Windows.

### Oracle GoldenGate Highlights

Extract, Distribution Service and Replicat work together to keep the databases in sync near real-time via incremental transaction replication. In all examples this function is accomplished by

- Adding supplemental transaction log data for update operations on the source system.
- Creating/Running the real-time Extract to retrieve and store the incremental changed data from the Oracle tables into Trail files on the source system.
- Creating/Running the real-time Distribution Service to send incremental changed data from the source environment to the target environment.

After initial instantiation (Heterogenous/Homogenous),

- Creating/Starting the real-time Replicat to replicate extracted data.

Once Extract and Replicat are running, changes are replicated perpetually.

## Oracle GoldenGate Token – LOGCSN

The Oracle GoldenGate Trail File contains a header token called **LOGCSN**, which represents the **Commit Sequence Number (CSN)** of a transaction. In the context of an Oracle database, LOGCSN corresponds to the Oracle **System Change Number (SCN)**. This value can be viewed using the logdump utility.

The LOGCSN token appears only in records with a Transaction Indicator (**TransInd**) of:

- 0 – indicating the beginning of a transaction.
- 3 – indicating a single-operation transaction.

For multi-operation transactions:

- The first operation will have TransInd = 0.
- Intermediate operations will have TransInd = 1.
- The final operation will have TransInd = 2.

If a transaction consists of only one operation, that operation will have TransInd = 3.

An example of logdump output is shown below.

```
Logdump > ghdr on
Logdump > ggstoken detail
Logdump > n
```

Hdr-Ind	:	E (x45)	Partition	:	.	(x0c)	
IOType	:	5 (x05)	OrigNode	:	255	(xff)	
TransInd	:	.	(x00)	FormatType	:	R (x52)	
SyskeyLen	:	0	(x00)	Incomplete	:	.	(x00)
AuditRBA	:	184	AuditPos	:	6714384		
Continued	:	N	(x00)	RecCount	:	1	(x01)

```
2025/05/02 00:06:58.000.000 Insert          Len      22 RBA 2139
Name: PDB1.SRCAPP.TEST1 (TDR Index: 1)
After Image:
0000 0600 0000 0200 3130 0100 0800 0000 0400 7265 | .....10.....re
6331 | c1
Partition x0c G b
```

```
GGG tokens:
TokenID x52 'R' ORAROWID          Info x00 Length 20
4141 4153 4c50 4141 4d41 4141 4144 6641 4142 0001 | AAASLPAAMAAAdFAAB..
TokenID x74 't' ORATAG            Info x01 Length 0
TokenID x4c 'L' LOGCSN           Info x00 Length 8
3132 3533 3139 3137 | 12531917
TokenID x36 '6' TRANID            Info x00 Length 19
3332 3632 3437 3539 3031 2e32 2e34 2e31 3133 35 | 3262475901.2.4.1135
TokenID x69 'i' ORATHREADID      Info x01 Length 2
0001 | ..
TokenID x42 'B' ORABLKSIZE        Info x01 Length 2
0200 | ..
```



## Delivery Process (Replicat) Use of LOGCSN

Users can now specify the **LOGCSN** from which to begin applying the changed data. This can be done using the START REPLICAT command.

```
START REPLICAT <group name> ATCSN <csn> | AFTERCSN <csn>]
```

- ATCSN <csn> causes Replicat to skip transactions in the Trail File until it finds the transaction indicator that contains the specified commit sequence number (CSN). All transactions with a CSN less than the specified one are skipped. The transaction whose begin-transaction record in the Trail File contains the specified CSN is applied, as are transactions after that CSN.
- AFTERCSN <csn> causes Replicat to skip transactions in the Trail File until it finds the first transaction after the one that contains the specified CSN. All transactions whose begin-transaction record in the Trail File contain a CSN less than, or equal to, the specified one is skipped.

The next step is to determine the appropriate CSN value from which the Replicat should begin applying the data. The location of this CSN depends on the method used for instantiating the target system.

## Oracle Data Pump Integration for Table Instantiation

Oracle GoldenGate now offers seamless integration with Oracle Data Pump, streamlining the instantiation process. During an Oracle Data Pump export, the commit sequence number (CSN) for each table is captured. Upon import, this CSN is applied to system tables and views in the target database. Replicat references these tables and views to automatically determine where to begin applying changes, eliminating the need for administrators to manually identify the starting CSN. When the Replicat parameter `DBOPTIONS ENABLE_INSTANTIATION_FILTERING1` is enabled, this feature also removes the requirement for individual MAP statements with `@FILTER(@GETENV('TRANSACTION', 'CSN'))` or the `HANDLECOLLISIONS` clause for each table.

### At the Source Database:

- Source system tables are automatically prepared when issuing the command `ADD TRANDATA / ADD SCHEMATRANDATA`

```
OGG (https://localhost Marketplace) 3> dblogin user=alias source
Successfully logged into database CDB$ROOT.

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 4> add schematradata
DBCSASH_PDB1.APPS
2025-04-30T10:41:23Z INFO OGG-01788 SCHEMATRANDATA has been added on schema "APPS".
2025-04-30T10:41:23Z INFO OGG-01976 SCHEMATRANDATA for scheduling columns has been added on
schema "APPS".
```

<sup>1</sup> To use this feature to replicate DDLs, DataPump has a fix starting from Oracle DB 23.4. Any Oracle DB version prior to that should request a backport of bug 35940664.

```

2025-04-30T10:41:23Z  INFO      OGG-10154  Schema level PREPARECSN set to mode NOWAIT on schema
"APPS"
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 5> info schematrandata
DBCSASH_PDB1.APPS

Schema level supplemental logging, excluding non-validated keys, is enabled on schema "APPS".
Schema level supplemental logging is enabled on schema "APPS" for all scheduling columns.
Schema "APPS" has 2 prepared tables for instantiation

```

```
SQL> select table_name, scn from dba_capture_prepared_tables where table_owner = 'APPS';
```

TABLE_NAME	SCN
CUSTOMERS	17000336
ORDERS	17000342

Note: The DBA\_CAPTURE\_PREPARED\_TABLES does not get populated till the first export of the tables. The SCN is the smallest system change number (SCN) for which the table can be instantiated. It is not the export SCN.

- Oracle Data Pump Export will automatically generate import actions to set instantiation CSN for each table at target upon import.

#### At the Target Database:

- Data Pump Import will populate system tables and views with instantiation CSNs

```
SQL> select source_object_name, instantiation_scn, ignore_scn from dba_apply_instantiated_objects
where source_object_owner = 'APPS';
```

SOURCE_OBJECT_NAME	INSTANTIATION_SCN	IGNORE_SCN
CUSTOMERS	16968822	0
ORDERS	16968822	0

- Use Replicat parameter (DBOPTIONS ENABLE\_INSTANTIATION\_FILTERING) to enable table level instantiation filtering.
- When Replicat is started, it queries the instantiation CSN for each new table mapping and filters both DDL and DML records accordingly. Only records with CSN greater than or equal to the table's instantiation CSN will be applied. The report file generated by Replicat will list each table and the specific CSN from which data application will begin.

```

2025-04-30 12:40:20  INFO      OGG-10155  Instantiation CSN filtering is enabled on table
APPS.CUSTOMERS at CSN 16,968,822.

2025-04-30 12:40:20  INFO      OGG-10155  Instantiation CSN filtering is enabled on table APPS.ORDERS
at CSN 16,968,822.

```



## Item To Review Before Starting Instantiation

Before starting any instantiation method, make sure no transactions were initiated or underway before the Extract process was registered/added.

The following query can be run to check for any open transactions before the Capture/Extract was added. The SQL should return no rows to ensure no transactions were initiated since the time of registering the Extract.

```
select start_scn, start_time
  from gv$transaction
 where start_scn < (select max(start_scn) from DBA_CAPTURE);
```

If there are transactions started before the Extract was registered, wait until the transactions are committed or rolled back. Starting instantiation in the middle will lead to missing data at the target database. If a particular transaction is running for longer than expected work with the user in committing the transaction or kill the session to roll back the transaction. Exercise caution and check with the business users to determine the criticality of the transactions before taking the corrective action to start the instantiation.

## Homogeneous Instantiation Methods

In a homogeneous instantiation, both the Oracle Database and the operating system version are identical across the environment. There are several Oracle methods available to create the target database in such a setting. Each method can be explored in greater detail through the relevant Oracle documentation guides. However, this document will provide an overview of the most used methods.

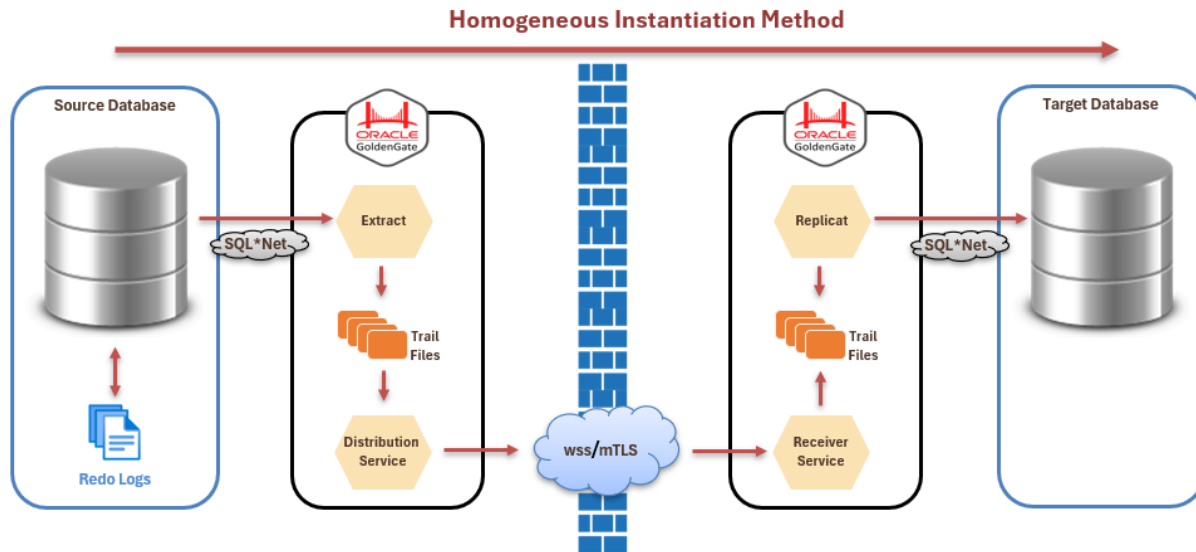


Figure1: Source and Target Systems are the same version of the database and operating system

### Cloning with Oracle RMAN

In this approach, the target database is instantiated by cloning the source database using RMAN. While this method is relatively straightforward, it still involves copying the necessary files to the target system. After the database is recovered, the GoldenGate CSN (or Oracle Database SCN) required to start the Replicat process will be recorded in the Oracle `alert.log` file on the target system.

The following steps outline the typical procedure for instantiating a homogeneous Oracle target system using RMAN.

---

*Note: This is applicable only in like-to-like environments. It assumes that the target system has an identical directory structure to the source. Additionally, the source system must have a listener configured, with `tnsnames.ora` entries defined for both the source and target databases.*

---

### Source Setup

#### Install Oracle GoldenGate

1. [Install GoldenGate 23ai Microservices Architecture.](#)

2. Enable minimal supplemental logging in the source database.

To enable minimal supplemental logging at the PDB level, first issue the following command on the source database system. Starting Oracle RDBMS 23ai, supplemental logging can be enabled at PDB level even if it is not enabled at the CDB\$ROOT level. Prior to 23ai, supplemental logging should be enabled at the CDB root level.

```
SQL> alter database add supplemental log data;
Database altered.
SQL> alter system switch logfile;
System altered.
SQL> alter session set container = DBCSASH_PDB1;
Session altered.
SQL> alter pluggable database add supplemental log data
Pluggable database altered.
```

3. Increase UNDO to allowed maximum (recommend 24 hours).

```
SQL> alter session set container = DBCSASH_PDB1;
Session altered.
SQL> show parameter undo_retention;
NAME                                TYPE          VALUE
-----
undo_retention                      integer       900
SQL> alter system set undo_retention = 86400; -- 86400 seconds = 24 hours
System altered.
SQL> show parameter undo_retention;
NAME                                TYPE          VALUE
-----
undo_retention                      integer       86400
```

4. Turn on FORCE LOGGING in the tablespaces where tables are created with NOLOGGING.

```
SQL> alter database force logging;
Database altered.
```

### Create DB user for OGG Extract

In a multitenant environment, the Extract user must be a common user and must log into the root container. This user must be created in the source database. In the following example, the Extract user ID is c##ggadmin. If the environment includes additional security features requiring extra permissions, those should be granted accordingly.

```
SQL> create user c##ggadmin identified by xxxxxxxxxxxx;
```

```
User created.
SQL> exec dbms_goldengate_auth.grant_admin_privilege('C##GGADMIN',container=>'ALL');
PL/SQL procedure successfully completed.
SQL> grant connect, resource to c##ggadmin container=all;
Grant succeeded.
```

Please see the [Oracle GoldenGate documentation](#) for more details on the relevant privileges required for this common user.

Starting Oracle Database 23ai, the root-level Extract is not supported and per-PDB Extract is the only supported Extract option. Hence the need for common user like c##ggadmin is not used. In addition, the role-based approach to grant privileges for replication has been introduced in Oracle Database 23ai simplifying the overall permissions needed to run Oracle GoldenGate. With the new roles there is no need to run the DBMS\_GOLDENGATE\_AUTH.GRANT\_ADMIN\_PRIVILEGE command. Following are the new roles being introduced in Oracle Database 23ai.

ROLE	Description
<b>OGG_CAPTURE</b>	Privileges necessary for using and managing Extract processes
<b>OGG_APPLY</b>	Privileges necessary for using and managing Replicat processes
<b>OGG_APPLY_PROCREP</b>	Privileges necessary to execute packages for procedural replication

With the above new roles, the create DB user for Extract on Oracle Database 23ai would be the following within the PDB:

```
SQL> create user ggext identified by xxxxxxxxxxxx;
User created.
SQL> grant OGG_CAPTURE to ggext;
Grant succeeded.
```

Similarly, for Replicat, it would be:

```
SQL> create user ggrep identified by xxxxxxxxxxxx;
User created.
SQL> grant OGG_APPLY, OGG_APPLY_PROCREP to ggrep;
Grant succeeded.
```

For more information on database users, privileges and Extract options please see the [documentation](#).

## Create GG UserId Alias

Setup the userid alias within GoldenGate to connect to the source database. This can be performed through the GoldenGate command line interface, `adminclient`, or through the Web UI. The `adminclient` executable is available as part of the GoldenGate software under the `$OGG_HOME/bin` directory.

Note: Starting GoldenGate 23ai, for security reasons, the use of `USERID` along with password for database connections is not allowed. To login to the database the `USERIDALIAS` must be used.

To login to the source database, setup the alias (source) using `C##GGADMIN` user.

```
$ cd $OGG_HOME/bin
$ ./adminclient

OGG (not connected) 1> connect https://localhost as oggadmin deployment Marketplace !
Password for 'oggadmin' at 'https://localhost':

OGG (https://localhost Marketplace) 2> alter credentialstore add user c##ggadmin@SOURCE_CDB
password xxxxxxxx alias source domain OracleGoldenGate

2025-04-28T15:31:02Z  INFO      OGG-15114  Credential store altered.

OGG (https://localhost Marketplace) 3> info credentialstore
Default domain: OracleGoldenGate
Alias: source
Userid: c##ggadmin@SOURCE_CDB
```

### Enable Supplemental Logging at the Schema or Table level

Setup the supplemental logging in the Source database either at the schema or table level.

```
OGG (https://localhost Marketplace) 4> dblogin useridalias source
Successfully logged into database CDB$ROOT.

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 8> ADD SCHEMATRANDATA
DBCSASH_PDB1.APPS

2025-04-28T15:39:00Z  INFO      OGG-01788  SCHEMATRANDATA has been added on schema "APPS".
2025-04-28T15:39:00Z  INFO      OGG-01976  SCHEMATRANDATA for scheduling columns has been added on
schema "APPS".
2025-04-28T15:39:00Z  INFO      OGG-10154  Schema level PREPARECSN set to mode NOWAIT on schema
"APPS"

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 5> info trandata
DBCSASH_PDB1.APPS.*
```



```
Logging of supplemental transaction log data is enabled for table DBCSASH_PDB1.APPS.CUSTOMERS.

Logical Replication for table DBCSASH_PDB1.APPS.CUSTOMERS is enabled.

Columns supplementally logged for table DBCSASH_PDB1.APPS.CUSTOMERS:
- "CUSTOMER_ID"

Prepared CSN for table DBCSASH_PDB1.APPS.CUSTOMERS: 16188376
Logging of supplemental transaction log data is enabled for table DBCSASH_PDB1.APPS.ORDERS.

Logical Replication for table DBCSASH_PDB1.APPS.ORDERS is enabled.

Columns supplementally logged for table DBCSASH_PDB1.APPS.ORDERS:
- "ORDER_ID"

Prepared CSN for table DBCSASH_PDB1.APPS.ORDERS: 16188439
```

Create and Start Extract Process on the Source Database

- 1. Login to Source database using the alias (source) created earlier via adminclient .

```
OGG (https://localhost Marketplace) 7> dblogin useridalias source
Successfully logged into database CDB$ROOT.
```


- 2. Create the parameter file for the Extract.

```
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 8> edit params eapps
2025-04-28T16:31:10Z INFO OGG-10183 Parameter file eapps.prm passed validity check.
```

- 3. The parameter file contents

```
EXTRACT eapps
-- User login
USERIDALIAS source DOMAIN OracleGoldenGate
DISCARDROLLOVER AT 01:00 ON SUNDAY
EXTTRAIL et
-- Report any long running transactions over 1 day and check every 1 hour
WARNLONGTRANS 1DAY CHECKINTERVAL 1HOUR
STATOPTIONS REPORTFETCH
REPORTCOUNT every 10 minutes, RATE
```





```
REPORTROLLOVER AT 01:00 ON SUNDAY
--- DDL Parameters
DDL INCLUDE MAPPED
DDLOPTIONS REPORT
TABLE DBCSASH_PDB1.APPS.* ;
```

4. Add the Extract first within GoldenGate before registering with the source database.

```
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 9> add extract eapps integrated
tranlog begin now
2025-04-28T16:21:20Z INFO OGG-08100 Integrated Extract added.
```

5. Add extrail to create a trail and associate it with the Extract.

```
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 10> add extrail et extract
eapps megabytes 2000
2025-04-28T16:23:42Z INFO OGG-08100 EXTTRAIL added.
```

6. Register the Extract with the SOURCE database. Registering the Extract will dump the required data dictionary to the redo logs.

```
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 11> register extract eapps
database container (DBCSASH_PDB1)
2025-04-28T16:35:54Z INFO OGG-02003 Extract group EAPPS successfully registered with
database at SCN 16193167.

OGG (https://localhost Marketplace) 13> info extract eapps
Extract      EAPPS      Last Started 2025-04-28 16:50      Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:02 ago)
Process ID         409465
Log Read Checkpoint Oracle Integrated Redo Logs
                  2025-04-28 16:50:50
                  SCN 0.16235825 (16235825)
Encryption Profile LocalWallet
```

## Create Distribution Service

1. Create a network credential with the target Oracle GoldenGate deployment details, enabling the distribution service to push trail files to the target system.

```

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 16> alter credentialstore add
user oggadmin password xxxxxxxxxx alias ggnetwork domain Network

2025-04-28T17:12:30Z INFO OGG-15114 Credential store altered.
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 17> info credentialstore domain
Network

Domain: Network

Alias: ggnetwork
Userid: oggadmin

```

## 2. Add Distribution path

```

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 18> add distpath dist_apps
source trail://localhost/services/v2/sources?trail=et target
wss://ogg23aiora.example.com/services/v2/targets?trail=et authentication useridalias ggnetwork
domain Network

2025-04-28T17:38:05Z INFO OGG-08511 The path 'dist_apps' has been added.
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 19> info distpath all

dist_apps      stopped

```

## Start OGG Processes before Target Instantiation

### 1. Start Extract and Distribution Service Process on Source Environment

```

$ cd $OGG_HOME/bin
$ ./adminclient

OGG (not connected) 1> connect https://localhost as oggadmin deployment Marketplace !
Password for 'oggadmin' at 'https://localhost':

OGG (https://localhost Marketplace) 2> info all

```

Program	Status	Group	Type	Lag at Chkpt	Time Since Chkpt
ADMINSVR	RUNNING				
DISTSRVR	RUNNING				
PMSVR	RUNNING				
RECVSRVR	RUNNING				
EXTRACT	STOPPED	EAPPS	INTEGRATED	00:00:00	00:03:06

```

OGG (https://localhost Marketplace) 3> start extract eapps

```

```

2025-04-29T10:43:20Z  INFO      OGG-00975  Extract group EAPPS starting.
2025-04-29T10:43:20Z  INFO      OGG-15426  Extract group EAPPS started.
OGG (https://localhost Marketplace) 4> info extract eapps
Extract      EAPPS      Last Started 2025-04-29 10:43      Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:08 ago)
Process ID      451887
Log Read Checkpoint Oracle Integrated Redo Logs
                  2025-04-29 10:43:05
                  SCN 0.16557085 (16557085)
Encryption Profile LocalWallet

```

## 2. Start Distribution path

```

OGG (https://localhost Marketplace) 5> start distpath dist_apps
2025-04-29T15:15:00Z  INFO      OGG-08513  The path 'dist_apps' has been started.

```

## RMAN Tasks on the Source

### 1. Backup source database

```

$ rman target /
RMAN> backup pluggable database DBCSASH_PDB1 plus archivelog;

```

2. Use scp or sftp to transfer the backup files to the destination server, placing them in the same directory path as on the source system. By default, this location is \$ORACLE\_HOME/fast\_recovery\_area/<CDB Name>backupset, unless it has been modified using the CONFIGURE command in RMAN.
3. Once the backup completes, capture the SCN that will be used with the duplicate command at the target. Record the SCN that is needed to clear the datafile fuzziness as reported below.

```

RMAN> restore database preview summary;
recovery will be done up to SCN 16581329
Media recovery start SCN is 16581329
Recovery must be done beyond SCN 17833514 to clear datafile fuzziness
Finished restoring at 02-MAY-25

```

Note: Save the SCN (**17833514**) that is needed to clear datafile fuzziness for duplicating the database at the target.

## RMAN Tasks on the Target

In RMAN terminology, the target is also referred as auxiliary. Perform the following tasks on the target environment.

1. Add a TNS entry in \$ORACLE\_HOME/network/admin/ tnsnames.ora file for the source database.
2. Create the password file in the \$ORACLE\_HOME/dbs directory.

```
$ orapwd file=orapw<SID> password=<password>
```

3. Startup auxiliary instance. Ensure the \$ORACLE\_SID is set correctly and the init<SID>.ora is placed under \$ORACLE\_HOME/dbs.

```
SQLPLUS> startup nomount
```

4. Create duplicate database. Make sure to specify the SCN captured after the backup at the source. This step will duplicate the database and open it.

```
$ rman target sys/<sys password for source db>@<source db from tnsnames.ora >  
RMAN> connect auxiliary /  
RMAN> duplicate pluggable database <Source_PDB> to <Target PDB> from active database using  
backupset until scn 17833514 nofilenamecheck;
```

---

*Note: The DUPLICATE DATABASE command can also be executed by running RMAN on the source system. In this approach, the source connects using CONNECT TARGET /, while the auxiliary connects using CONNECT AUXILIARY SYS@<target\_db>. Regardless of the method used, backup files must be copied to the destination server. For this scenario, the target database must have an entry in the tnsnames.ora file located on the host where the auxiliary instance is running. Additionally, the source server must have a listener configured to accept connections for the target database while it is in a NOMOUNT state. This is achieved by explicitly defining the service name or SID for the destination database in the listener.ora configuration file.*

---

5. Once complete review the Oracle alert log to get last SCN that the database was recovered to. Look in the Oracle alert log for the following entry:

```
RESETLOGS after incomplete recovery UNTIL CHANGE 17833514
```

In this instance, the Replicat should start processing at SCN **17833514**.

6. Stop and restart the database:

```
SQL> shutdown immediate
```

```
SQL> startup
```

## Target Setup

### Install Oracle GoldenGate

1. [Install GoldenGate 23ai Microservices Architecture.](#)
2. Set Library paths.

### Create DB User for OGG Replicat

In a multitenant environment, the Replicat user can be created in the PDB.

```
SQL> create user ggadmin identified by xxxxxxxxxxxx;  
User created.  
SQL> exec dbms_goldengate_auth.grant_admin_privilege('GGADMIN');  
PL/SQL procedure successfully completed.  
SQL> grant connect, resource to ggadmin;  
Grant succeeded.
```

Starting Oracle Database 23ai, role-based approach to grant privileges for replication has been introduced simplifying the overall permissions needed to run Oracle GoldenGate. With the new roles there is no need to run the `DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE` command. Following are the new roles being introduced in Oracle Database 23ai.

ROLE	Description
OGG_CAPTURE	Privileges necessary for using and managing Extract processes
OGG_APPLY	Privileges necessary for using and managing Replicat processes
OGG_APPLY_PROCREP	Privileges necessary to execute packages for procedural replication

## Create GG UserID alias

To login to the target database, setup the userid alias using the GGADMIN user created above.

```
$ cd $OGG_HOME/bin
$ ./adminclient
OGG (not connected) 3> connect https://localhost as oggadmin deployment Marketplace !
Password for 'oggadmin' at 'https://localhost':
OGG (https://localhost Marketplace) 5> alter credentialstore add user ggadmin@TARGET_PDB1
password xxxxxxxx alias target domain OracleGoldenGate
2025-04-29T09:51:12Z INFO OGG-15114 Credential store altered.
OGG (https://localhost Marketplace) 5> info credentialstore

Default domain: OracleGoldenGate
Alias: target
Userid: ggadmin@TARGET_PDB1
```

## Login to Target DB and create Checkpoint table

Perform the following on the target database to prepare for the Replicat process.

```
$ cd $OGG_HOME/bin
$ ./adminclient
OGG (not connected) 3> connect https://localhost as oggadmin deployment Marketplace !
Password for 'oggadmin' at 'https://localhost':
OGG (https://localhost Marketplace) 6> dblogin useridalias target domain OracleGoldenGate
Successfully logged into database.
OGG (https://localhost Marketplace as target@DBCSPH) 10> edit globals
2025-04-29T09:58:01Z INFO OGG-10183 Parameter file GLOBALS passed validity check.
OGG (https://localhost Marketplace as target@DBCSPH) 11> view globals
GGSCHEMA ggadmin
OGG (https://localhost Marketplace as target@DBCSPH) 12> add checkpointtable ggadmin.checkpoint
OGG (https://localhost Marketplace as target@DBCSPH) 14> info checkpointtable ggadmin.*
2025-04-29T10:01:50Z INFO OGG-08100 Checkpoint table GGADMIN.CHECKPOINT has been created on
2025-04-29 09:59:08.
```

## Create Replicat Process on Target Database

1. Login to the Target database using the useridalias (target) created earlier via adminclient.

```
OGG (https://localhost Marketplace) 7> dblogin useridalias target  
Successfully logged into database.
```

2. Create the parameter file for the Replicat.

```
OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 13> edit params rapps  
2025-04-29T10:28:28Z INFO OGG-10183 Parameter file rapps.prm passed validity check.
```

3. Contents of the parameter file for the Replicat.

```
REPLICAT RAPPS  
--- User login  
USERIDALIAS target DOMAIN OracleGoldenGate  
  
REPORTCOUNT EVERY 30 MINUTES, RATE  
REPORTROLLOVER AT 01:00 ON SUNDAY  
DISCARDROLLOVER AT 01:00 ON SUNDAY  
  
--- DDL Parameters  
DDL INCLUDE ALL  
DDLOPTIONS REPORT  
  
SOURCECATELOG DBCSASH_PDB1  
MAP APPS.*, TARGET APPS.*;
```

4. Add Replicat for the target database using the adminclient.

```
OGG (https://localhost Marketplace as target@DBCSPH) 20> add replicat rapps, parallel,  
exttrail et, checkpointtable ggadmin.checkpoint  
2025-04-29T10:31:54Z INFO OGG-08100 Parallel Replicat added.  
OGG (https://localhost Marketplace as target@DBCSPH) 21> info rapps  
No Extract groups exist.  
  
Replicat RAPPS Initialized 2025-04-29 10:31 Status STOPPED  
Parallel
```

```
Checkpoint Lag      00:00:00 (updated 00:00:56 ago)
Log Read Checkpoint File et0000000000
                   First Record  RBA 0
Encryption Profile   LocalWallet
```

### Start the Replicat Process

1. Start the Replicat process using the ATCSN option and pass on the SCN that was captured as part of the RMAN duplicate.

```
OGG (https://localhost Marketplace) 23> start replicat rapps atcsn 17833514
```

### Notes for Oracle RAC and ASM

The process of duplicating an Oracle RAC database or converting between a single-instance and a RAC configuration is similar to the procedure for duplicating a single-instance database. To restore and recover a RAC database, a dedicated connection to a single instance is required. This necessitates setting the `CLUSTER_DATABASE` initialization parameter to `FALSE`. Once the restore and recovery operations are complete, the parameter can be reset to `TRUE`, and the remaining RAC instances can then be brought online.

As with single-instance recovery using RMAN, the `alert.log` file of the instance used for the recovery will contain the SCN (System Change Number) up to which the database was recovered. This SCN must be used to start the Oracle GoldenGate Replicat process.

When recovering a database into an ASM (Automatic Storage Management) instance, it is generally most efficient to use Oracle Managed Files. The parameters `DB_CREATE_FILE_DEST`, `DB_CREATE_ONLINE_LOG_DEST_1`, and `DB_RECOVERY_FILE_DEST` should be configured to specify the ASM disk groups intended for the data and temp files, redo log files, and archive log files, respectively.

---

For more specific details see the [Oracle Database Backup and Recovery Advanced User's Guide](#).

---

### Additional Information on RMAN approach

There are many more options to the RMAN duplicate command which are beyond the scope of this paper. The RMAN command can be executed with '**active**' option which copies the files over the network rather than restoring the files from backup. This option may be helpful when disk space or the need to transfer backups may be an issue. There is also further syntax which allows for different directory structures and the ability to skip tablespaces which are not of interest.

Much of these options are documented in the *Oracle Database Backup and Recovery Advanced User's Guide*. There are also many knowledge base articles written on this subject, including but not limited to the MOS note DocId [228257.1](#) (RMAN Duplicate Database Feature).

### Oracle Data Guard

An opened Data Guard instance can also be used as a target database in a homogeneous environment. The required CSN can be determined from the following query on the opened Data Guard instance:



```
SQL> select standby_became_primary_scn from v$database;
```

The data returned by this query is the CSN that should be used to start the Replicat.

## Oracle Snapshot Standby

Another option to instantiate the target database is to convert the physical standby (if available) into a Snapshot Standby, making it writable and using Oracle Data Pump to export the data from the Snapshot Standby. As Oracle Data Pump requires the database to be in read write mode, the physical standby must be converted into a Snapshot Standby until the export is complete. Once the target database is instantiated and GoldenGate replication is configured, the snapshot standby can be converted back to physical standby. Use of snapshot standby also ensures data consistency during the initial setup of replication.

The following section covers the steps to convert the physical standby to a snapshot standby, performing export and import and starting the replicat.

### At Physical Standby

1. Prior to converting the physical standby to a snapshot standby, the redo apply must be stopped and the current\_scn should be noted which would be used to start the Replicat using AFTERCSN option.

```
SQL> select open_mode, current_scn from v$database;

OPEN_MODE          CURRENT_SCN
-----
READ ONLY WITH APPLY    5103646953

SQL> alter database recover managed standby database cancel;

Database altered.

SQL> select open_mode, current_scn from v$database;

OPEN_MODE          CURRENT_SCN
-----
READ ONLY          5103647075

SQL> /

OPEN_MODE          CURRENT_SCN
-----
READ ONLY          5103647075
```

2. Convert to snapshot standby which will enable expdp to read write.

```
SQL> alter database convert to snapshot standby;

Database altered.

SQL> select open_mode, current_scn from v$database;
```

OPEN_MODE	CURRENT_SCN
MOUNTED	0

### 3. Open database for expdp.

```
SQL> alter database open;
Database altered.

SQL> select open_mode, current_scn from v$database;
```

OPEN_MODE	CURRENT_SCN
READ WRITE	5103647008

### 4. Perform the export as required either at full, schema or table level.

```
$expdp directory=dumpdir schemas=apps parallel=4 dumpfile=ora_load_%u.dmp
Username: system
Note: Any DB user with DBA privileges will do
Password: *****
Note: The export log needs to be checked for errors.
```


### 5. Restore the database back to physical standby mode and restart the redo apply process.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.
Total System Global Area      2768240640 bytes
Fixed Size                     2928248 bytes
Variable Size                  704643464 bytes
Database Buffers               2046820352 bytes
Redo Buffers                   13848576 bytes
Database mounted.

SQL> alter database convert to physical standby;
Database altered.
SQL> alter database open;
Database altered.
SQL> select open_mode, current_scn from v$database;
```

OPEN_MODE	CURRENT_SCN
READ ONLY WITH APPLY	5103647349

Note: If the OPEN\_MODE was READ ONLY, issue the following command to insure apply was enabled:



```
SQL> alter database recover managed standby database disconnect;
```

#### At the Target environment

1. Import the data into the target database.

```
$impdp system/password DIRECTORY=dumpdir DUMPFILE= ora_load_%u.dmp
```

2. Start the replicat using the ATSCN option using the current\_scn from the physical standby prior to the snapshot standby conversion.

```
OGG (https://localhost Marketplace) 3> start replicat rapps aftercsn 5103647075
```

## Heterogeneous Instantiation Methods

A heterogeneous environment refers to a configuration where the source and target systems are different. These differences can range from variations in database versions or operating systems to completely different database platforms.

In such environments, the process of cloning the source system involves creating a clone using one of the Homogeneous Instantiation Methods as described in earlier sections. This clone will then serve as the source for the initial data load into the target system.

Once the clone is built, the Change Synchronization Number (CSN) recorded during the cloning process will become the point where Replicat starts its data processing, ensuring that changes are applied consistently in the target system.

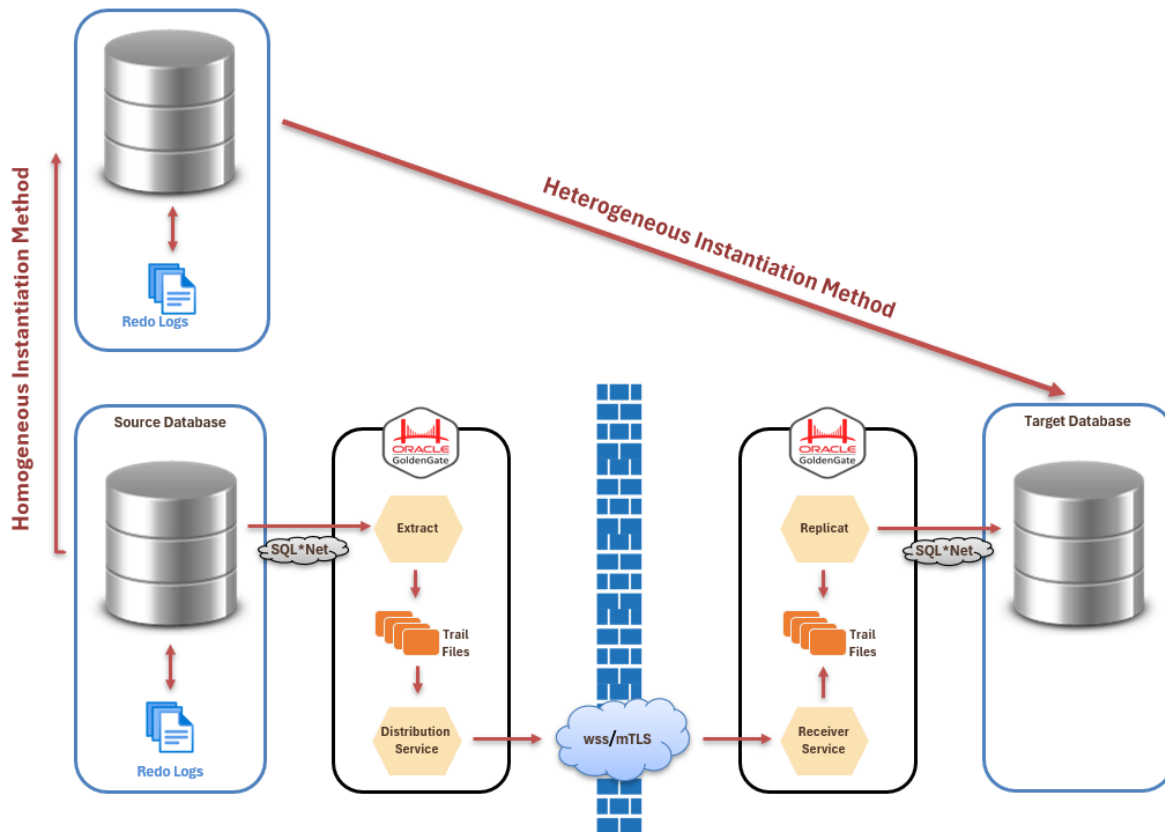


Figure 2: Source and Target Systems have different Operating System, Database, or both.

Some of the methods that can be used to instantiate the target from the clone are

- Oracle Transportable Tablespaces (Oracle to Oracle only)
- Oracle expdp/impdp (Oracle to Oracle only)

- GoldenGate Initial Load Extract/Replicat.

The one heterogeneous instantiation method that does not always require a clone is Oracle EXPDP/IMPDP. These utilities can be used either to extract data from a cloned database or directly from the source database. The choice between using a clone and accessing the production source typically depends on the goal of minimizing overhead on the source system during the data extraction process.

## Transportable Tablespaces

Oracle Transportable Tablespaces allows users to quickly move user tablespaces to Oracle databases across platforms.

This feature requires that the source system be in read only mode which is why this method cannot be used on a highly available database. This method works perfectly if the source for the target database is a clone of the highly available database because the clone database can easily be put in read only mode with no effect on the highly available production database.

---

*For more specific details on Transportable Tablespaces please see the Oracle Database Administrator's Guide.*

---

## GoldenGate Initial Load (Oracle to Any)

GoldenGate's Initial Load process allows users to easily move data cross database and cross platform.

GoldenGate's initial load can be performed from an active source database. Users and applications can access and update data while the load is running. There are different approaches for using GoldenGate Initial Load and each approach is well documented in the GoldenGate for Windows and Unix Administrator's Guide.


## Oracle Data Pump

The Oracle Data Pump allows a more targeted approach to moving only specific tables or schemas to the new target system. These utilities will allow you to instantiate different Oracle versions on different platforms with or without using a clone.

Data Pump Export (expdp) functionality is similar to that of the original Export utility (exp), but they are separate utilities, and their files are not compatible. Data Pump Import (impdp) functionality is like that of the original Import utility (imp), but it can only read files created by Data Pump Export. Visit <http://support.oracle.com> for details on its support for exp, imp, expdp and impdp.

Prior to Oracle GoldenGate 12.2, exports were typically performed directly on the production system using the FLASHBACK\_SCN option. The SCN value specified during the export process would then be used as the CSN value for initializing the Replicat process. It is important to note that the entire export had to be taken using the same FLASHBACK\_SCN value, regardless of whether the export was split across multiple files, executed through parallel sessions, or leveraged Oracle Data Pump's parallelism features.

For OGG Versions 12.2 and above, the FLASHBACK\_SCN is no longer required when using the Replicat Parameter, DBOPTIONS ENABLE\_INSTANTIATION\_FILTERING. Oracle Data Pump and Oracle GoldenGate have a tighter integration. The CSN for each table is captured on an Oracle Data Pump Export. The CSN is then applied to system tables and views on



the target database on an import. These views and system tables are referenced by Replicat when applying data to the target database. This 12.2 feature no longer requires administrators to know what CSN number Replicat should be started with. Replicat will handle it automatically when the Replicat Parameter `DBOPTIONS ENABLE_INSTANTIATION_FILTERING` is enabled.

In either scenario, it is essential to ensure that sufficient undo space is available to complete the export process. If the production system is under heavy load and lacks the disk space necessary to expand undo capacity, a duplicate system can be configured to perform the export instead. Provided that the duplicate system remains idle, a transactionally consistent export can be executed without the need to use the `FLASHBACK_SCN` option.

---

*To use **ENABLE\_INSTANTIATION\_FILTERING** feature to replicate DDLs, Data Pump has a fix starting from Oracle DB 23.4. Any Oracle DB versions prior to 23.4 should request a backport of Bug 35940664.*

*For more specific details on Data Pump Export, Data Pump Import, original export and import utilities, please see the Oracle Database Utilities Guide.*

---

## Example 1: Full Export Using Oracle Data Pump with “flashback\_scn” option

### OGG Configuration on the Source System

- Create an OGG User.
- Add supplemental log data to all schemas being replicated for update operations.
- Create an Extract parameter file on source system.
- Register Extract in database.
- Set up an initial Extract checkpoint on source system.
- Create Local Trail file.
- Create a Distribution Service on source system.
- Create Remote Trail File.

### OGG Configuration on the Target System

- Create an OGG User.
- Create a checkpoint table.
- Create a Replicat parameter file on target system.
- Set up an initial Replicat checkpoint on target system.

### Start OGG Processes

- Start Extract on source system.
- Start Distribution service on source system.

For more details on the above steps and parameter files, reference the detailed examples in the Homogenous section of this document.

### Data Pump Export (Full) on Source System

1. Create a database directory:

```
SQL> create directory DUMPDIR as '<some directory>';  
Directory created.
```

2. Get the current SCN on the source database.

```
SQL> select current_scn from v$database;
```

CURRENT\_SCN

-----  
**16968822**

3. Run the Data Pump Export using the flashback SCN obtained in the previous step. The following example demonstrates the use of the `expdp` utility with a degree of parallelism (DOP) set to 4. When sufficient system resource such as CPU, memory, and I/O capacity are available, a higher DOP can significantly reduce the export duration, potentially by a factor corresponding to the DOP value (e.g., up to 4x for a DOP of 4). It is important to note that `expdp` relies on Oracle Database's parallel execution infrastructure, including parameters like `parallel_max_servers`, which must be properly configured to fully leverage the parallelism.

```
$ expdp directory=DUMPDIR full=y parallel=4 dumpfile=src_pdb1_%u.dmp flashback_scn=16968822
Username: system
Note: Any DB user with DBA privileges will do
Password:
Note: The export log needs to be checked for errors.
```

#### Data Pump Import on Target System

- Start an import using `impdp` to the target database when the export step is complete.

```
$ impdp system/password@TARGETPDB directory=DUMPDIR DUMPFILE=src_pdb1_%u.dmp
```

#### Start Replicat on Target System

- Start Replicat on the Target System using the `AFTERSCN` option with the `CURRENT_SCN` number from above.

```
OGG (https://localhost Marketplace) 1> start replicat rapps aftercsn 16968822
```



## Example 2: Full Export Using Oracle Data Pump & OGG Instantiation Filtering Feature

### OGG Configuration on the Source System

- Create an OGG User.
- Add supplemental log data to all schemas being replicated for update operations.

```
OGG (https://localhost Marketplace) 3> dblogin useridalias source
Successfully logged into database CDB$ROOT.

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 4> add schematradata
DBCSASH_PDB1.APPS
2025-04-30T10:41:23Z INFO OGG-01788 SCHEMATRANDATA has been added on schema "APPS".
2025-04-30T10:41:23Z INFO OGG-01976 SCHEMATRANDATA for scheduling columns has been
added on schema "APPS".
2025-04-30T10:41:23Z INFO OGG-10154 Schema level PREPARECSN set to mode NOWAIT on
schema "APPS"

OGG (https://localhost Marketplace as source@DBCSASH/CDB$ROOT) 5> info schematradata
DBCSASH_PDB1.APPS

Schema level supplemental logging, excluding non-validated keys, is enabled on schema
"APPS".

Schema level supplemental logging is enabled on schema "APPS" for all scheduling columns.

Schema "APPS" has 2 prepared tables for instantiation
```

- Create an Extract parameter file on source system.
- Register Extract in database.
- Set up an initial Extract checkpoint on source system.
- Create Local Trail file.
- Create a Distribution path on source system.
- Create Remote Trail file.

### OGG Configuration on the Target System

- Create an OGG User.
- Create a checkpoint table.
- Create a Replicat parameter file on target system.

- Enable the parameter `DBOPTIONS ENABLE_INSTANTIATION_FILTERING`.

```
REPLICAT RAPPS
--- User login
USERIDALIAS target DOMAIN OracleGoldenGate

DBOPTIONS ENABLE_INSTANTIATION_FILTERING

REPORTCOUNT EVERY 30 MINUTES, RATE
REPORTROLLOVER AT 01:00 ON SUNDAY
DISCARDROLLOVER AT 01:00 ON SUNDAY

--- DDL Parameters
DDL INCLUDE ALL
DDLOPTIONS REPORT

SOURCECATELOG DBCSASH_PDB1
MAP APPS.*, TARGET APPS.*;
```

- Set up an initial Replicat checkpoint on target system.

### Start OGG Processes

- Start Extract on source system.
- Start Distribution path on source system.

For more details on the above steps and parameter files, reference the detailed examples in the Homogenous section of the document.

### Data Pump Export (Full) on Source System

1. Create a database directory:

```
SQL> create directory DUMPDIR as '<some directory>';
Directory created.
```

2. Get the current SCN on the source database.

```
SQL> select current_scn from v$database;
```

```
CURRENT_SCN
-----
16968822
```

3. Run the Data Pump Export using the flashback SCN obtained in the previous step. The following example demonstrates the use of the `expdp` utility with a degree of parallelism (DOP) set to 4. When sufficient system resource such as CPU, memory, and I/O capacity are available, a higher DOP can significantly reduce the export duration, potentially by a factor corresponding to the DOP value (e.g., up to 4x for a DOP of 4). It is important to note that `expdp` relies on Oracle Database's parallel execution infrastructure, including parameters like `parallel_max_servers`, which must be properly configured to fully leverage the parallelism.

```
$ expdp directory=DUMPDIR full=y parallel=4 dumpfile=src_pdb1_%u.dmp
Username: system
Note: Any DB user with DBA privileges will do
Password:
Note: The export log needs to be checked for errors.
```

### Data Pump Import on Target System

- Start an import using `impdp` to the target database when the export step is complete.

```
$ impdp system/password@TARGETPDB directory=DUMPDIR DUMPFILE=src_pdb1_%u.dmp
```


### Start Replicat on Target System

- Start Replicat on the Target System.

```
OGG (https://localhost Marketplace) 1> start replicat rapps
```

- View the report file to verify CSN Filtering is occurring. You should see the following messages.

```
2025-04-30 12:40:20 INFO      OGG-10155  Instantiation CSN filtering is enabled on
table APPS.CUSTOMERS at CSN 16,968,822.
```



```
2025-04-30 12:40:20 INFO OGG-10155 Instantiation CSN filtering is enabled on
table APPS.ORDERS at CSN 16,968,822.
```

## Where To Go for More Information

This document provided a quick overview of both homogeneous and heterogeneous target instantiation options from an Oracle source database. For additional information please refer to the following links.

- Oracle GoldenGate 23ai installation: [Oracle GoldenGate 23ai documentation](#)
- Oracle GoldenGate 23ai: [Oracle GoldenGate 23ai Microservices Documentation](#)
- [Oracle Database Utilities Guide](#) for more details on the data pump utilities.
- RMAN commands: [Oracle Database Backup and Recovery Users Guide](#)
- Oracle GoldenGate 23ai: [Connecting Two Deployments using External RootCA Certificates](#)



## Appendix – A: TNS Information

Following are the TNS information used for the test environment that includes the source and target database connection information.

As a best practice, perform a `tnsping` from each of the servers with the target TNS alias to ensure the connection works.

```
source =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = svr-proddb)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = proddb)
    )
  )

target =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = svr-target)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = targetdb)
    )
  )
```

For a sample TNS configuration in a RAC environment, please see the [Oracle documentation](#).



Oracle Corporation, World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065, USA

Worldwide Inquiries  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200

#### CONNECT WITH US



[blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)

[facebook.com/oracle](https://facebook.com/oracle)

[twitter.com/oracle](https://twitter.com/oracle)

[oracle.com](https://oracle.com)

#### Hardware and Software, Engineered to Work Together

Copyright © 2025, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0623

Oracle GoldenGate Best Practices:  
Instantiation from an Oracle Source Database  
Document ID 1276058.1 June 2025

Author: Tracy West  
Contributing Authors: Sunil Vernekar, Somu Rajarathinam



Oracle is committed to developing practices and products that help protect the environment