

Oracle GoldenGate Best Practices:

Instantiation from an Oracle Source Database

Version 12c

Document ID 1276058.1



Tracy West
Consulting Solution Architect
A-Team – Cloud Solution Architects



DISCLAIMER

This sample code is provided 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 this 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), this 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.

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.

Introduction	5
Prerequisites	1
Additional OGG Considerations	1
Overview	2
Homogenous Instantiation	2
Heterogenous Instantiation	2
Oracle GoldenGate Highlights	2
Oracle GoldenGate Token - LOGCSN	3
Delivery Process (Replicat) Use of LOGCSN	3
Oracle Data Pump Integration for Table Instantiation	4
Before Starting Instantiation	5
Homogeneous Instantiation Methods	6
Cloning with Oracle RMAN	6
Sample Steps - Cloning with Oracle RMAN	6
Source Setup	7
Create OGG user (oraggs)	7
Grant oraggs user permissions	7
Enable Supplemental logging at the schema level or table level	7
Source MGR Parameter File	7
Extract Parameter File	8



Extract Pump Parameter File	8
Creating OGG Processes and Trail Files on Source Server	8
Target Setup	8
Create OGG user (oraggs)	8
Grant oraggs user permissions	9
Target MGR Parameter File	9
Target Replicat Parameter File	9
Syntax to Create Target Groups and Trail Files	10
Start of OGG Processes Before Target Instantiation	10
RMAN Tasks on Source	10
RMAN Tasks On the Target (RMAN refers to as auxiliary)	10
Start Replicat on Target System	11
Notes for Oracle RAC and ASM	12
Additional Information on RMAN approach	12
Oracle Data Guard	12
Heterogeneous Instantiation Methods	13
Transportable Tablespaces	14
GoldenGate Initial Load (Oracle to Any)	14
Oracle Data Pump	14



- Example 1: Full Export Using Oracle Data Pump For OGG Versions Prior to 12.2 16
 - OGG Configuration on the Source System 16
 - OGG Configuration on the Target System 16
 - Start OGG Processes 16
 - Data Pump Export (Full) on Source System 16
 - Data Pump Import on Target System 17
 - Start Replicat on Target System 17

- Example 2: Full Export Using Oracle Data Pump For OGG Versions 12.2 and above 18
 - OGG Configuration on the Source System 18
 - OGG Configuration on the Target System 18
 - Start OGG Processes 19
 - Data Pump Export (Full) on Source System 19
 - Data Pump Import on Target System 19
 - Start Replicat on Target System 19

- Example 3: Schema Level Export Using Oracle Data Pump For OGG Versions Prior to 12.2 20
 - OGG Configuration on the Source System 20
 - OGG Configuration on the Target System 20



Start OGG Processes	20
Data Pump Export (Schema) on Source System	20
Data Pump Import on Target System	21
Start Replicat on Target System	21
Example 4: Schema Level Export Using Oracle Data Pump For OGG Versions 12.2 and above	22
OGG Configuration on the Source System	22
OGG Configuration on the Target System	22
Start OGG Processes	23
Data Pump Export (Schema) on Source System	23
Data Pump Import on Target System	23
Start Replicat on Target System	23
Where to Go for More Information	24



Introduction

This document is an introduction to Oracle GoldenGate's best practices and guidelines for instantiation of a target database from an Oracle Source Database. This document is intended for Oracle Database Administrators (DBAs), and Oracle Developers with some basic knowledge of Oracle GoldenGate software product. The document is intended to be a supplement to the existing series of documentation available from Oracle.

The following assumptions have been made during the writing of this document:

- » The reader has basic knowledge of Oracle GoldenGate products and concepts
- » Referencing Oracle GoldenGate Version 10 and above
- » Referencing Oracle RDBMS Version 10.2 and above
- » Referencing OS: All Oracle GoldenGate supported platforms for Oracle

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

One of the biggest challenges in creating a target database for replication is to insure that no "collisions" (applying data that has been already been "applied" by virtue of copying data after the same change data has been captured) or data loss will occur when applying the change data after the initial instantiation of the target database. Oracle GoldenGate has addressed this issue by including functionality to apply the change data to the target at the appropriate CSN (Commit Sequence Number) where no collisions will occur. Please note that OGG uses "CSN" and not "SCN", which are functionally interchangeable but syntactically distinct. To use this functionality without incurring any data loss you have to make sure to use a read consistent image of your source database to do the instantiation and keep track of the CSN of your consistent image. This document provides more details and examples on how to use LOGCSN which is stored in the Oracle GoldenGate Trail Files starting with Oracle GoldenGate version 10.

Prerequisites

If you plan to execute the instructions in this best practice, make sure all software is already installed. The reader should be familiar with basic OGG architecture and functionality. For Oracle RDBMS 12.1.0.2 and above, the `init.ora` parameter, `enable_goldengate_replication` must be set to `TRUE` in both the source and target databases.

The following table describes items that are referred to throughout the document. You will need to identify your installation-specific values and substitute them as you go.

Item	Reference	Description
Unix Programs	/ggs	Directory of Unix GoldenGate installation.
Unix Parameter Files	/ggs/dirprm	Directory for GoldenGate parameter files.
Unix Report Files	/ggs/dirrpt	Directory for output from GoldenGate programs.
Unix Definitions Files	/ggs/dirdef	Directory for generated Oracle DDL and definition files.
GGG temporary storage	/ggs/dirdat	Directory to hold temporary Extract trails
Oracle Logon	userid, password	User ID and password for the source or target database. When implementing Integrated Extract or Replicat, this user must be granted admin privileges with the DBMS_GOLDENGATE_AUTH procedure on the both source and target databases.
Unix System Network Address	Target Server Source Server	IP address/hostname of the target Unix system in network. IP address/hostname of the source Unix system in network

Additional OGG Considerations

- » Key Management
 - » Need to share encryption key for trail file encryption between OGG source and OGG target
- » Password Security
 - » Passwords used by GoldenGate for Database access encrypted using AES.
- » Command Security
 - » Manager configured with CMDSEC security. Restrict by IP Address and User for IPC Messages using ACCESSRULE Parameter.
- » Process Management
 - » GoldenGate can be configured to auto start / restart processes upon any failure including network failure.
- » Monitoring
 - » All GoldenGate deployments can be monitored by EM with optional Monitor Agent which are not covered in this document.



Overview

This document will address different approaches to Instantiation of a target database. All approaches will have similar setup 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 exactly the same between the source and the target databases. There are a few Oracle methods that can be used 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 different configurations. The difference could be as simple as a database version or 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, Extract pump 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

- » Starting the Manager program in all OGG installed systems.
- » 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 target Unix system.
- » Creating/Running the real-time Extract pump to send incremental changed data from the source environment to the target environment.

After initial instantiation (Heterogenous/Homogenous),

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

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

Notes on Command Syntax: Commands throughout the document make specific references to directories, file names, checkpoint group names, begin times, etc. Unless otherwise noted, these items do not have to correspond exactly in your environment; they are used to illustrate concrete examples. For exact syntax, consult the Oracle GoldenGate Reference Guide.

Oracle GoldenGate Token - LOGCSN

As of Version 10 of Oracle GoldenGate a new header token was introduced into the GoldenGate Trail File. This token represents the commit sequence number for the transaction. This token is named LOGCSN. For an Oracle database LOGCSN maps to the Oracle System Change Number (SCN). LOGCSN can be viewed in logdump. It only exists for records that have a TransInd of 0 (begin Transaction) or 3 (Single Operation Transaction). If a transaction contains multiple operations then the first operation will contain a TransID of 0, denoting the begin of a transaction; an operation after this first operation but within the same transaction will contain a TransInd of 1; the last operation in this transaction will contain a TransID of 2,. If a transaction only contains 1 operation then this operation will contain a TransID of 3. Below is an example of what you would see in logdump.

```
Logdump >ghdr on
Logdump >ggstoken detail
Logdump >n
-----
Hdr-Ind      :      E (x45)      Partition :      . (x0c)
UndoFlag    :      . (x00)      BeforeAfter:    A (x41)
RecLength   :      58 (x003a)   IO Time      : 2017/07/17 14:08:22.000.864
IOType      :      5 (x05)      OrigNode     :    255 (xff)
TransInd    :      . (x00)      FormatType   :      R (x52)
SyskeyLen   :      0 (x00)      Incomplete  :      . (x00)
AuditRBA    :      29739        AuditPos     : 26471856
Continued   :      N (x00)      RecCount    :      1 (x01)

2017/07/17 14:08:22.000.864 Insert          Len      58 RBA 2042
Name: APPS.TCUSTMER (TDR Index: 1)
After Image:
0000 0008 0000 0004 5749 4c4c 0001 0013 0000 000f | .....WILL.....
4247 2053 4f46 5457 4152 4520 434f 2e00 0200 0b00 | BG SOFTWARE CO.....
0000 0753 4541 5454 4c45 0003 0004 0000 5741 | ...SEATTLE.....WA

GGs tokens:
TokenID x52 'R' ORAROWID          Info x00 Length 20
 4141 4167 3966 4141 4541 4149 7270 4d41 4141 0001 | AAAG9fAAEAAIrpMAAA..
TokenID x74 't' ORATAG            Info x01 Length 0
TokenID x4c 'L' LOGCSN           Info x00 Length 9
 3832 3937 3139 3732 37          | 829719727
TokenID x36 '6' TRANID           Info x00 Length 11
 342e 3238 2e39 3437 3537 34    | 4.28.947574
TokenID x69 'i' ORATHREADID      Info x01 Length 2
| ..
```

Delivery Process (Replicat) Use of LOGCSN

As of Oracle GoldenGate version 10, we can specify the LOGCSN we wish to begin to apply the changed data. This is now part of the GGSCI command START REPLICAT.

Syntax START REPLICAT <group name> ATCSN <csn> | AFTERCASN <csn>]

- » ATCSN <csn> causes Replicat to skip transactions in the Trail File until it find 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 are skipped.

The trick now becomes determining what that CSN number is that the Replicat should begin to start applying data. The location of that CSN number is dependent on the form of instantiation that is used.

Oracle Data Pump Integration for Table Instantiation

This is valid for Oracle GoldenGate (OGG) version 12.2 and above. This is a transparent integration of OGG with Oracle Data Pump. This feature requires OGG version 12.2 and above on the source and target system. 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. It also eliminates the need of specification of individual MAP for each imported table with the **@FILTER(@GETENV('TRANSACTION','CSN'))** or **HANDLCOLLISIONS** clause .

At Source Database:

- » Source System tables are automatically prepared when issuing the command **ADD TRANDATA / ADD SCHEMATRANDATA**.

```
GGSCI> add schematrandata apps
```

```
2017-07-17 13:30:27 INFO OGG-01788 SCHEMATRANDATA has been added on
schema apps.
```

```
2017-07-17 13:30:27 INFO OGG-01976 SCHEMATRANDATA for scheduling columns
has been added on schema apps.
```

```
GGSCI> info schematrandata apps
```

```
2017-07-17 13:30:57 INFO OGG-06480 Schema level supplemental logging,
excluding non-validated keys, is enabled on schema APPS.
```

```
2017-07-17 13:30:57 INFO OGG-01980 Schema level supplemental logging is
enabled on schema APPS for all scheduling columns.
```

```
2017-07-17 13:30:57 INFO OGG-10462 Schema APPS have 2 prepared tables
for instantiation.
```

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

TABLE_NAME	SCN
TCUSTMER	829695121
TCUSTORD	829695126

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 Target Database:

- » Datapump 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
TCUSTMER	829723224	0
TCUSTORD	829723223	0

- » New Replicat parameter (DBOPTIONS ENABLE_INSTANTIATION_FILTERING) to enable table level instantiation filtering
- » Start replicat, who will query instantiation CSN on any new mapping and filter records accordingly Filters out DDL and DML records based on each table's instantiation CSN . Output in the report file will show the table name and to what CSN the replicat will start applying data.

```
2017-07-17 15:02:51 INFO OGG-10155 Instantiation CSN filtering is  
enabled on table APPS.TCUSTMER at CSN 829,723,224.
```

```
2017-07-17 15:02:51 INFO OGG-10155 Instantiation CSN filtering is  
enabled on table APPS.TCUSTORD at CSN 829,723,223.
```

Before Starting Instantiation

In order to not miss any data, before starting any instantiation method, you must make sure that all open transactions that existed when the real-time extract was started are completed.

The best source for this information is the V\$TRANSACTION performance view (GV\$TRANSACTION in a RAC database). After starting your extract, determine what transactions exist in this view. When a transaction is complete, it will no longer exist in this view. You may have to query this view many times. If a particular transaction is running for longer than expected, you will need to investigate who is running that transaction and what that transaction doing. Ultimately, you may discover that you will need to kill the session that owns the long running transaction in order to begin the instantiation of your target system.

Homogeneous Instantiation Methods

For a homogeneous instantiation the Oracle Database and the operating system version are exactly the same. There are a few Oracle methods that can be used 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.

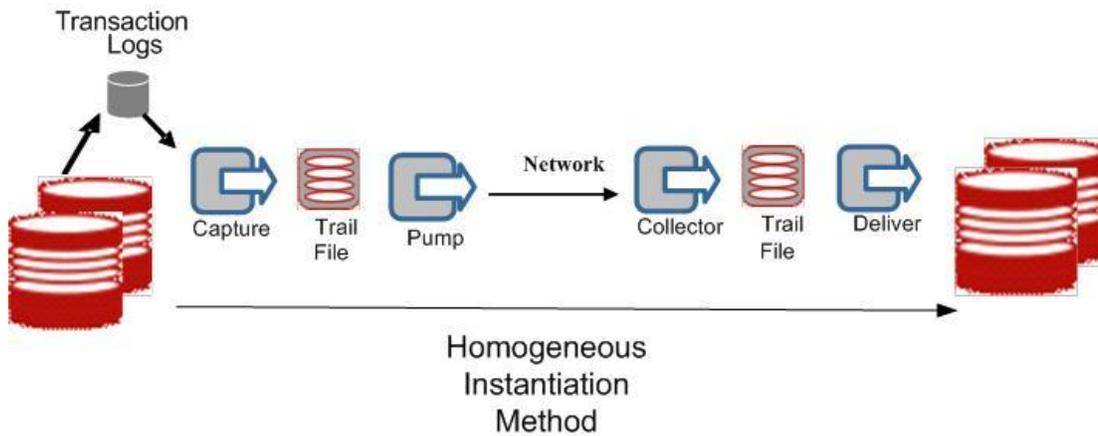


Figure1: Source and Target Systems are the same version of the database and operating system

Cloning with Oracle RMAN

In this scenario, the database is cloned using RMAN. This is a simpler method, but it still requires files to be copied to the target system. The GoldenGate CSN/Oracle Database SCN that should be used to start Replicat will be written to the Oracle alert.log file on the target after the database is recovered.

Below are sample steps that you would use to instantiate an Oracle homogeneous target system using RMAN.

Sample Steps - Cloning with Oracle RMAN

This sample scenario will only work in like to like environments. In this document we are assuming that the destination has the exact same directory structure as the source. The source system needs a listener configured with entries in the tnsnames.ora for both the source and target.

Source Setup

[See 12c Installation Guide](#)

1. Install OGG Software
2. Set Library paths, Oracle Environmental Variables
3. Create subdirectories for OGG

```
GGSCI> create subdirs
```
4. Enable minimal supplemental logging in the database

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```
5. Increase UNDO to allowed maximum (recommend 24 hours)
6. Turn on FORCE LOGGING in the tablespaces where tables are created with NOLOGGING

Create OGG user (oraggs)

[See 12c Installation Guide](#)

```
SQL> create user oraggs identified by xxxxxx;  
SQL> alter user oraggs default tablespace <OGG Tablespace Name> temporary  
tablespace TEMP quota unlimited on <OGG Tablespace Name>;
```

Grant oraggs user permissions

[See 12c Installation Guide](#)

```
SQL> grant resource, dba to oraggs;  
SQL> exec dbms_goldengate_auth.grant_admin_privilege('ORAGGS' );
```

Enable Supplemental logging at the schema level or table level

[See 12c Installation Guide](#)

```
----- Source System  
GGSCI> DBLOGIN USERID oraggs, PASSWORD xxxx  
GGSCI> ADD SCHEMATRANDATA APPS  
OR  
GGSCI> ADD TRANDATA APPS.<TABLENAME>
```

Source MGR Parameter File

```
PORT 7801  
DYNAMICPORTLIST 7802-7803  
PURGEOLDEXTRACTS ./dirdat/et* , USECHECKPOINTS, MINKEEPHOURS 72  
AUTORESTART ER *, RETRIES 3, WAITMINUTES 10, RESETMINUTES 60  
DOWNREPORTMINUTES 15  
DOWNCRITICAL  
LAGCRITICALSECONDS 10  
LAGINFOMINUTES 0  
LAGREPORTMINUTES 15
```

```
ACCESSRULE, PROG *, IPADDR <Source IP Address>, ALLOW
ACCESSRULE, PROG *, IPADDR *, DENY
```

Note: The local system IPADDR may be the first entry of the loopback (127.0.0.1) or localhost in the /etc/hosts file. The MGR.rpt file will list what IP Address is trying to initiate the command.

Extract Parameter File

```
EXTRACT EAPPS
--- User login
USERID oraggs, PASSWORD xxxx

DISCARDFILE ./dirrpt/eapps.dsc, APPEND
DISCARDROLLOVER AT 01:00 ON SUNDAY

EXTTRAIL ./dirdat/et

STATOPTIONS REPORTFETCH
REPORTCOUNT every 10 minutes, RATE
REPORTROLLOVER AT 01:00 ON SUNDAY

--- DDL Parameters
DDL INCLUDE MAPPED
DDLOPTIONS REPORT

TABLE APPS.* ;
```

Extract Pump Parameter File

```
EXTRACT PAPPS
RMTHOST <Target ip address>, MGRPORT 7801
PASSTHRU
RMTRAIL ./dirdat/rt
TABLE APPS.* ;
```

Creating OGG Processes and Trail Files on Source Server

```
GGSCI> DBLOGIN USERID oraggs, PASSWORD xxxx
GGSCI> REGISTER EXTRACT EAPPS, DATABASE
GGSCI> ADD EXTRACT EAPPS, INTEGRATED TRANLOG, BEGIN NOW
GGSCI> ADD EXTTRAIL ./dirdat/et, EXTRACT EAPPS, MEGABYTES 500
GGSCI> ADD EXTRACT PAPPS, EXTTRAILSOURCE ./dirdat/et
GGSCI> ADD RMTRAIL ./dirdat/rt, EXTRACT PAPPS, MEGABYTES 500
```

Target Setup

[See 12c Installation Guide](#)

1. Install OGG Software
2. Set Library paths, Oracle Environmental Variables
3. Create subdirectories for OGG

```
GGSCI> create subdirs
```

Create OGG user (oraggs)

[See 12c Installation Guide](#)

```
SQL> create user oraggs identified by xxxxxx;
SQL> alter user oraggs default tablespace <OGG Tablespace Name> temporary
tablespace TEMP quota unlimited on <OGG Tablespace Name>;
```

Grant oraggs user permissions

[See 12c Installation Guide](#)

```
SQL> grant resource, dba to oraggs;
SQL> exec dbms_goldengate_auth.grant_admin_privilege('ORAGGS' );
```

Target MGR Parameter File

```
PORT 7801
DYNAMICPORTLIST 7802-7803
PURGEOLDEXTRACTS ./dirdat/rt* , USECHECKPOINTS, MINKEEPHOURS 72
AUTORESTART ER *, RETRIES 3, WAITMINUTES 10, RESETMINUTES 60
DOWNREPORTMINUTES 15
DOWNCRITICAL
LAGCRITICALSECONDS 10
LAGINFOMINUTES 0
LAGREPORTMINUTES 15
ACCESSRULE, PROG *, IPADDR <Target IP Address>, ALLOW
ACCESSRULE, PROG *, IPADDR <Source IP Address>, ALLOW
ACCESSRULE, PROG *, IPADDR *, DENY
```

Note: The local system IPADDR may be the first entry of the loopback (127.0.0.1) or localhost in the /etc/hosts file. The MGR.rpt file will list what IP Address is trying to initiate the command.

Target Replicat Parameter File

```
REPLICAT RAPPS
--- ASSUMETARGETDEFS is ignored in OGG 12.2
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rapps.dsc, APPEND

--- User login
USERID oraggs, PASSWORD xxxx

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

--- DDL Parameters
DDL INCLUDE ALL
DDLOPTIONS REPORT

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

Syntax to Create Target Groups and Trail Files

```
GGSCI> DBLOGIN USERID oraggs, PASSWORD xxxx
GGSCI> ADD CHECKPOINTTABLE chktbl
GGSCI> ADD REPLICAT RAPPS, INTEGRATED, EXTTRAIL ./dirdat/rt, CHECKPOINTTABLE
chktbl
```

Start of OGG Processes Before Target Instantiation

The manager process on all servers should be started.

```
GGSCI> start MGR
```

Start Extract and Extract Pump Process on Source Environment

```
GGSCI(Source Server)> START EXTRACT EAPPS
GGSCI(Source Server)> START EXTRACT PAPPS
```

RMAN Tasks on Source

- » Create init.ora file for the Target system.
- » Run the following SQL (if you currently use an spfile; if you already use a pfile then you can use the pfile and skip this step)

```
SQLPLUS> create pfile='<path>/init<sid>.ora' from spfile ;
```

- » Copy init<sid>.ora file to target (to \$ORACLE_HOME/dbs directory on destination)

```
sftp> put init<sid>.ora
```

- » Backup source database

```
OS> RMAN target /
```

```
RMAN> backup database plus archivelog ;
```

- » Copy backup files to destination located in the same location as they were written on the source (default is \$ORACLE_HOME/dbs but it may have been changed using the CONFIGURE command in RMAN).

```
sftp> <backup files>
```

- » Once backup completes, capture SCN to clear datafile fuzziness

```
RMAN> restore database preview summary;
```

```
Media recovery start SCN is 34821758
```

```
Recovery must be done beyond SCN 34822578 to clear datafile fuzziness
```

Note for future use the second SCN (34822578) returned from the command, it appears at the end of the output.

RMAN Tasks On the Target (RMAN refers to as auxiliary)

- » Add a tnsnames.ora entry (in \$ORACLE_HOME/network/admin) for the source database.

- » Create the password file (in the \$ORACLE_HOME/dbs directory):

```
$ORACLE_HOME/dbs> orapwd file=orapw<SID> password=<password>
```

- » Start up auxiliary instance (make sure the \$ORACLE_SID is set correctly and the init<SID>.ora is in \$ORACLE_HOME/dbs)

```
SQLPLUS> startup nomount
```

- » Create duplicate database. This step will open the database

```
OS> RMAN target sys/<sys password for source db>@<source database from
tnsnames.ora >
RMAN> connect auxiliary /
RMAN> duplicate target database to <databasename> nofilenamecheck until scn <value
returned from RMAN restore database preview summary command> ;
```

Note: you can also run the duplicate database command by running RMAN on the source. In that case you use connect target / on the source, and connect auxiliary sys@<target db>. In either case you must copy the backup files to the destination server. Also for this scenario you must include a tnsnames entry for the target database which would be added to the tnsnames.ora file on the host with the auxiliary. In addition, on the source server, you must have listener configured to accept requests for the destination database in a nomount state. This is done by explicitly including the service/SID in the listener.ora configuration file.

- » 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 34822578
```

At **34822578** in this sample is the SCN where we want Replicat to start processing

- » Create spfile (if you want to use an spfile; if not you can skip this step and the next step below):

```
SQLPLUS> create spfile from pfile='$ORACLE_HOME/dbs/init<sid>.ora' ;
```

- » Stop and Restart Database to pick up spfile:

```
SQLPLUS> shutdown immediate
SQLPLUS> startup
```

Start Replicat on Target System

- » START REPLICAT <REPLICAT NAME>, ATCSN <SCN>

```
GGSCI > START REPLICAT RAPPS, ATCSN 34822578
```

Notes for Oracle RAC and ASM

The steps to duplicate an Oracle RAC database, or to go from a single instance to an Oracle RAC database, or vice versa, are not very different from the approach to duplicate a single instance database. In order to restore and recover a RAC database you have to have a dedicated connection to the database, which implies you have to set the `cluster_database` initialization parameter to false. After the restore and recovery you can reset this parameter to true and bring up the other instances. Similar to the single instance recovery using RMAN the alert.log for the instance that was used to perform the recovery contains the SCN that was recovered until, which must be used to start replicat.

If you recover a database into an ASM instance then it is probably easiest to use Oracle Managed files in ASM. Use the `db_create_file_dest`, `db_create_online_log_dest_1` and `db_recovery_file_dest` parameters to indicate which ASM diskgroup(s) you want to use for the data and temp files, redo log files and archive log files.

See Also: Oracle Database Backup and Recovery Advanced User's Guide for more specific details.

Additional Information on RMAN approach

There are many more options to the RMAN duplicate command which are beyond the scope of this paper. This 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 article id 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.

See Also: Oracle Database Backup and Recovery Advanced User's Guide for more specific details.

Heterogeneous Instantiation Methods

A heterogeneous environment is a configuration where the source and target are different. The difference could be as simple as a database version or 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.

In a heterogeneous environment, a clone of the source system would need to be created using one of the Homogeneous Instantiation Methods described in the previous sections of this paper. This clone would then become the source for the initial load of the target system. The CSN determined when building the clone will become the CSN where Replicat will begin processing.

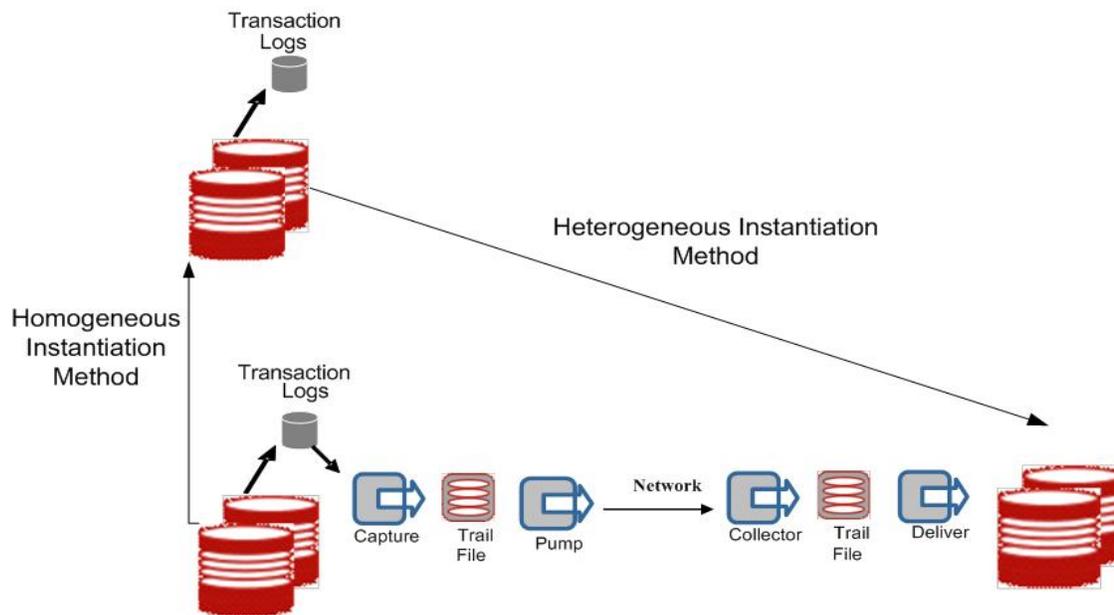


Figure2: Source and Target Systems have different Operating System, Database or both.

Some of the methods that can be used to instantiate the target from the clone are OracleTransportable Tablespaces (Oracle to Oracle only), Oracle expdp/impdp (Oracle to Oracle only), or GoldenGate Initial Load extract/replicat.

The one heterogeneous instantiation method that does not always require a clone is Oracle EXPDP/IMPDP. You can use these utilities to pull the data from the clone or to pull the data directly from the source database. The decision to use a clone versus using the product source usually comes down reducing the overhead on the source system for pulling this data.



Transportable Tablespaces

Oracle Transportable Tablespaces allows users to quickly move user tablespaces across Oracle databases. Oracle Transportable Tablespaces became available in Oracle8i Database of the database. However, this feature did not have cross platform support until version 10 of the database.

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 affect on the highly available production database.

See Also: Oracle Database Administrator's Guide for more specific details on Transportable Tablespaces.

GoldenGate Initial Load (Oracle to Any)

GoldenGate's Initial Load process allows users to easily move data cross database and cross platform. For example, data can be moved from an Oracle Database 10g/HP-UX environment to a Microsoft SQL Server/Windows environment.

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

Data Pump Export (expdp) and Data Pump Import (impdp) were introduced in Oracle 10g. Data Pump Export (expdp) functionality is similar to that of the original Export utility (exp), they are completely separate utilities and their files are not compatible. Data Pump Import (impdp) functionality is similar to that of the original Import utility (imp), but it can only read files created by Data Pump Export. The original Export utility (exp) is not be supported in Oracle Database 11g for general use. Import (imp) will be supported to enable import of export files created with older versions of Oracle. Visit <http://metalink.oracle.com> for details on its support for exp, imp, expdp and impdp.



Prior to Oracle GoldenGate 12.2, the export would be performed directly on the production system by using the export with the FLASHBACK_SCN option. Then the FLASHBACK_SCN used for the export would then be the CSN value used for the Replicat. Note that you have to take the entire export using the same value for FLASHBACK_SCN for your entire export, even if you use multiple export files (e.g. you run multiple sessions in parallel, or in the case of Data Pump Export, you use Oracle's parallelism).

For OGG Versions 12.2 and above, the FLASHBACK_SCN is no longer required when using the Replicat Parameter, DBOPTIONS ENABLE_INSTANTIATION_FILTERING, is enabled. 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 case, make sure that there will be sufficient undo space available to run through the export. If the production system is very busy and there is no disk space available to add any undo space but you can configure a duplicate system, then you can use that duplicate system to run a transactionally consistent export (assuming the duplicate system will be idle there is no need to use FLASHBACK_SCN in that case).

See Also: Oracle Database Administrator's Guide for more specific details on Data Pump Export, Data Pump Import, original export and import utilities.

Example 1: Full Export Using Oracle Data Pump For OGG Versions Prior to 12.2

OGG Configuration on the Source System

- » Create an OGG User
- » Add supplemental log data to all schemas being replicated for update operations.
- » Create a Manager Parameter file on source system
- » 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 an Extract Pump parameter file on source system.
- » Create Remote Trail File

OGG Configuration on the Target System

- » Create an OGG User
- » Create a Manager parameter file on target system.
- » 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 Manager on both source and target systems
- » Start Extract on source system
- » Start Extract Pump 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

- » Create a database directory:

```
SQLPLUS> create directory dumpdir as '<some directory>' ;
```
- » Get the current SCN on the source database :

```
SQLPLUS> select current_scn from v$database ;
CURRENT_SCN
-----
28318029
```
- » Run the Data Pump Export using the flashback SCN you obtained in the previous step. The following example shows running the expdp utility at a Degree Of Parallelism (DOP) of 4. If you have sufficient system resources (CPU, memory and IO) then running at a higher DOP will decrease the amount of time it takes to take the export (up to 4x for a DOP of 4). Note that expdp uses Oracle Database parallel execution settings (e.g. parallel_max_servers) which have to be set appropriately in order to take advantage of parallelism.



Other processes running in parallel may be competing for those resources. See the Oracle Documentation for more details.

```
>expdp directory=dumpdir full=y parallel=4 dumpfile=ora102_%u.dmp  
flashback_scn=28318029
```

```
>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 DIRECTORY=dumpdir DUMPFILE=ora102_%u.dmp
```

Start Replicat on Target System

- » Start Replicat after import is complete:

```
GGSCI> START REPLICAT RAPP, AFTERCSN 28318029
```

Example 2: Full Export Using Oracle Data Pump For OGG Versions 12.2 and above

OGG Configuration on the Source System

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

```
GGSCI> add schematrandata apps
2017-07-17 13:30:27 INFO OGG-01788 SCHEMATRANDATA has been added on schema
apps.
2017-07-17 13:30:27 INFO OGG-01976 SCHEMATRANDATA for scheduling columns
has been added on schema apps.
```

```
GGSCI> info schematrandata apps
2017-07-17 13:30:57 INFO OGG-06480 Schema level supplemental logging,
excluding non-validated keys, is enabled on schema APPS.
2017-07-17 13:30:57 INFO OGG-01980 Schema level supplemental logging is
enabled on schema APPS for all scheduling columns.
2017-07-17 13:30:57 INFO OGG-10462 Schema APPS have 2 prepared tables for
instantiation.
```

- » Create a Manager Parameter file on source system
- » 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 an Extract Pump parameter file on source system.
- » Create Remote Trail File

OGG Configuration on the Target System

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

- Must have Parameter DBOPTIONS ENABLE_INSTANTIATION_FILTERING

```
REPLICAT RAPPS
--- ASSUMETARGETDEFS is ignored in OGG 12.2
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rapps.dsc, APPEND
```

DBOPTIONS ENABLE_INSTANTIATION_FILTERING

```
--- User login
USERID oraggs, PASSWORD xxxx
```

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

```
--- DDL Parameters
DDL INCLUDE ALL
```

DDLOPTIONS REPORT

MAP APPS.*, TARGET APPS.*;

- » Set up an initial Replicat checkpoint on target system.

Start OGG Processes

- » Start Manager on both source and target systems
- » Start Extract on source system
- » Start Extract Pump 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

- » Create a database directory:

```
SQLPLUS> create directory dumpdir as '<some directory>' ;
```

- » Run the Data Pump Export. The following example shows running the expdp utility at a Degree Of Parallelism (DOP) of 4. If you have sufficient system resources (CPU, memory and IO) then running at a higher DOP will decrease the amount of time it takes to take the export (up to 4x for a DOP of 4). Note that expdp uses Oracle Database parallel execution settings (e.g. parallel_max_servers) which have to be set appropriately in order to take advantage of parallelism. Other processes running in parallel may be competing for those resources. See the Oracle Documentation for more details.

```
>expdp directory=dumpdir full=y parallel=4 dumpfile=ora102_%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 DIRECTORY=dumpdir DUMPFILE=ora102_%u.dmp
```

Start Replicat on Target System

- » Start Replicat after import is complete:

```
GGSCI> START REPLICAT RAPP
```

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

```
2017-07-17 15:02:51 INFO OGG-10155 Instantiation CSN filtering is enabled  
on table APPS.TCUSTMER at CSN 829,723,224.
```

```
2017-07-17 15:02:51 INFO OGG-10155 Instantiation CSN filtering is enabled  
on table APPS.TCUSTORD at CSN 829,723,223.
```

Example 3: Schema Level Export Using Oracle Data Pump For OGG Versions Prior to 12.2

OGG Configuration on the Source System

- » Create an OGG User
- » Add supplemental log data to all schemas being replicated for update operations.
- » Create a Manager Parameter file on source system
- » 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 an Extract Pump parameter file on source system.
- » Create Remote Trail File

OGG Configuration on the Target System

- » Create an OGG User
- » Create a Manager parameter file on target system.
- » 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 Manager on both source and target systems
- » Start Extract on source system
- » Start Extract Pump 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 (Schema) on Source System

- » Create a database directory:

```
SQLPLUS> create directory dumpdir as '<some directory>' ;
```
- » Get the current SCN on the source database :

```
SQLPLUS> select current_scn from v$database ;
CURRENT_SCN
-----
28318029
```
- » Run the Data Pump Export using the flashback SCN you obtained in the previous step. The following example shows running the expdp utility at a Degree Of Parallelism (DOP) of 4. If you have sufficient system resources (CPU, memory and IO) then running at a higher DOP will decrease the amount of time it takes to take the export (up to 4x for a DOP of 4). Note that expdp uses Oracle Database parallel execution settings



(e.g. `parallel_max_servers`) which have to be set appropriately in order to take advantage of parallelism. Other processes running in parallel may be competing for those resources. See the Oracle Documentation for more details.

```
>expdp directory=dumpdir schemas=apps parallel=4 dumpfile=ora102_%u.dmp  
flashback_scn=28318029
```

```
>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 DIRECTORY=dumpdir DUMPFILE=ora102_%u.dmp`

Start Replicat on Target System

- » Start Replicat after import is complete:
`GGSCI> START REPLICAT RAPP, AFTERCSCN 28318029`

Example 4: Schema Level Export Using Oracle Data Pump For OGG Versions 12.2 and above

OGG Configuration on the Source System

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

```
GGSCI> add schematrandata apps
2017-07-17 13:30:27 INFO OGG-01788 SCHEMATRANDATA has been added on schema
apps.
2017-07-17 13:30:27 INFO OGG-01976 SCHEMATRANDATA for scheduling columns
has been added on schema apps.
```

```
GGSCI> info schematrandata apps
2017-07-17 13:30:57 INFO OGG-06480 Schema level supplemental logging,
excluding non-validated keys, is enabled on schema APPS.
2017-07-17 13:30:57 INFO OGG-01980 Schema level supplemental logging is
enabled on schema APPS for all scheduling columns.
2017-07-17 13:30:57 INFO OGG-10462 Schema APPS have 2 prepared tables for
instantiation.
```

- » Create a Manager Parameter file on source system
- » 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 an Extract Pump parameter file on source system.
- » Create Remote Trail File

OGG Configuration on the Target System

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

- Must have Parameter DBOPTIONS ENABLE_INSTANTIATION_FILTERING

```
REPLICAT RAPPS
--- ASSUMETARGETDEFS is ignored in OGG 12.2
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rapps.dsc, APPEND
```

DBOPTIONS ENABLE_INSTANTIATION_FILTERING

```
--- User login
USERID oraggs, PASSWORD xxxx
```

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

```
--- DDL Parameters
DDL INCLUDE ALL
```

DDLOPTIONS REPORT

MAP APPS.*, TARGET APPS.*;

- » Set up an initial Replicat checkpoint on target system.

Start OGG Processes

- » Start Manager on both source and target systems
- » Start Extract on source system
- » Start Extract Pump 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 (Schema) on Source System

- » Create a database directory:

```
SQLPLUS> create directory dumpdir as '<some directory>' ;
```

- » Run the Data Pump Export. The following example shows running the expdp utility at a Degree Of Parallelism (DOP) of 4. If you have sufficient system resources (CPU, memory and IO) then running at a higher DOP will decrease the amount of time it takes to take the export (up to 4x for a DOP of 4). Note that expdp uses Oracle Database parallel execution settings (e.g. parallel_max_servers) which have to be set appropriately in order to take advantage of parallelism. Other processes running in parallel may be competing for those resources. See the Oracle Documentation for more details.

```
>expdp directory=dumpdir schemas=apps parallel=4 dumpfile=ora102_%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 DIRECTORY=dumpdir DUMPFILE=ora102_%u.dmp
```

Start Replicat on Target System

- » Start Replicat after import is complete:

```
GGSCI> START REPLICAT RAPP
```

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

```
2017-07-17 15:02:51 INFO OGG-10155 Instantiation CSN filtering is enabled  
on table APPS.TCUSTMER at CSN 829,723,224.
```

```
2017-07-17 15:02:51 INFO OGG-10155 Instantiation CSN filtering is enabled  
on table APPS.TCUSTORD at CSN 829,723,223.
```



Where to Go for More Information

Hopefully, this white paper has provided a quick overview of what options you can utilize to instantiate your target environment. Undoubtedly, you will eventually fine-tune this process in your own environment.

Reference the [Oracle Database 12.1 Documentation](#) for additional information on the Oracle 12.1 RDBMS.

Reference the [Oracle GoldenGate 12c Reference Guide](#) and the [Oracle GoldenGate 12c Administration Guide](#) for additional information on:

- Extract Parameters for Windows and Unix
- Replicat Parameters for Windows and Unix
- Extract Management Considerations
- Replicat Management Considerations



CONNECT WITH US

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

 oracle.com

[A-Team Chronicles](https://atteam-oracle.com) atteam-oracle.com

Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

Hardware and Software, Engineered to Work Together

Copyright © 2014, 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. 0717

OGG 12c Tutorial for Oracle to Oracle (12c Multitenant Container Database)

July 2017

Author: Tracy West

Contributing Authors:



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