



Oracle Maximum
Availability Architecture

Oracle GoldenGate Classic Architecture with Oracle Real Application Clusters Configuration Best Practices

ORACLE TECHNICAL BRIEF | OCTOBER 2022

Introduction	1
Configuration Overview	2
Oracle GoldenGate	2
Oracle Real Application Clusters	3
Oracle Clusterware	4
Oracle Grid Infrastructure Agents	4
Oracle Database File System (DBFS)	4
Oracle Automatic Storage Management Cluster File System (ACFS)	4
Configuration Best Practices	5
Step 1: Create the Database Services	5
Step 2: Create Oracle Net Alias for Oracle GoldenGate Database Connections	5
Step 3: Set Up a File System on Oracle RAC	6
Oracle Database File System (DBFS)	6
Oracle ASM Cluster File System (ACFS)	10
Step 4: Install Oracle GoldenGate	13
Step 5: Configure Oracle GoldenGate and the Database	13
Step 6: Configure GoldenGate Credential Store for Secure Database Login	13
Step 7: Configure Oracle GoldenGate Processes to Use the Shared File System	14
Extract Configuration	14
Data Pump Configuration	15
Replicat Configuration	16
Step 8: Configure Autostart of Extract, Data Pump, and Replicat Processes	17

Step 9: Oracle Clusterware Configuration	18
Summary of Recommendations When Deploying on Oracle RAC	22
References	23
Appendix A: Troubleshooting Oracle GoldenGate on Oracle RAC	24
1. XAG log file	24
2. CRS trace file	24
3. GoldenGate report files	25
Example Configuration Problems	25
1. Incorrect parameter settings in the <code>mount-dbfes.conf</code> file	25
2. GoldenGate is not stopping when requested by XAG	27
3. Problems with file locking on DBFS	29

Introduction

As of Oracle GoldenGate 21c, Oracle GoldenGate Classic Architecture is deprecated and may be desupported and unavailable in a future release. Oracle recommends using the Oracle GoldenGate Microservices Architecture. Refer to the new and improved [Oracle GoldenGate Microservices Architecture with Oracle Real Application Clusters Configuration Best Practices](#) technical brief when using Oracle GoldenGate Microservices with Oracle RAC.

Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware allow Oracle Database to run any packaged or custom application across a set of clustered servers. This capability provides continual database service uptime for node and instance failures, most planned maintenance activities, and for Oracle RAC expansion. If a clustered node fails, the Oracle Database service continues running on the surviving nodes. When more processing power is needed, you can add another node without interrupting user access to the database or data.

Oracle Clusterware is a cluster manager that is designed specifically for the Oracle Database. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle resources (such as database instances and listeners). If a failure occurs, Oracle Clusterware automatically attempts to restart the failed resource. During outages, Oracle Clusterware relocates the processing performed by the inoperative resource to a backup resource. For example, if a node fails, then Oracle Clusterware relocates the database services used by the application to surviving nodes and instances in the cluster if desired.

This technical brief describes best practices for configuring Oracle GoldenGate to work with Oracle RAC, Oracle Clusterware, and Oracle Database File System (DBFS) or Oracle ASM Cluster File System (ACFS). Oracle GoldenGate is instrumental for many reasons, including the following:

- » To migrate to an Oracle Database, incurring minimal downtime
- » As part of an application architecture that requires Oracle RAC, plus the flexible availability and logical replication features provided by Oracle GoldenGate, such as active-active database for data distribution and continuous availability, and zero or minimal downtime during planned outages for system migrations, upgrades, and maintenance
- » To implement a near real-time data warehouse or consolidated database on Oracle RAC, sourced from various, possibly heterogeneous, source databases, populated by Oracle GoldenGate

- 
- » To capture from an OLTP application running on Oracle RAC to support further downstream consumption, such as middleware integration

This paper focuses on configuring Oracle GoldenGate to run on Oracle RAC, which can act as the source database, as the target database, or in some cases, as both source and target databases for Oracle GoldenGate processing.

Configuration Overview

This section introduces Oracle GoldenGate, Oracle RAC, Oracle Clusterware, Oracle Database File System (DBFS) and Oracle ASM Cluster File System (ACFS). For more information about these features, see the [References](#) section at the end of this technical brief.

Oracle GoldenGate

Oracle GoldenGate provides real-time, log-based change data capture and delivery between homogenous and heterogeneous systems. This technology enables you to construct a cost-effective and low-impact real-time data integration and continuous availability solution.

Oracle GoldenGate replicates data from committed transactions with transaction integrity and minimal overhead on your existing infrastructure. The architecture supports multiple data replication topologies such as one-to-many, many-to-many, cascading, and bidirectional. Its wide variety of use cases includes real-time business intelligence; query offloading; zero-downtime upgrades and migrations; and active-active databases for data distribution, data synchronization, and high availability. Figure 1 shows the Oracle GoldenGate classic architecture referenced throughout this technical brief.

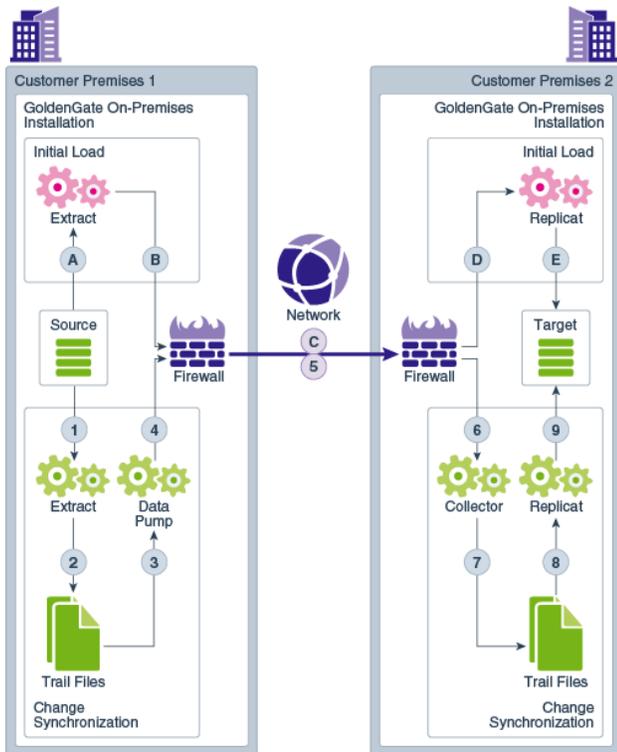


Figure 1: Oracle GoldenGate Classic Architecture

It is recommended that you use Oracle GoldenGate version 21c or later. The latest version of Oracle GoldenGate can be downloaded from:

<http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html>

This paper does not cover configuration for downstream capture mode Extract, or the Oracle GoldenGate Microservices Architecture.

If Oracle Data Guard is also part of the database configuration, refer to the Oracle MAA technical brief “Transparent Role Transitions with Oracle Data Guard and Oracle GoldenGate” at:

<http://www.oracle.com/technetwork/database/features/availability/maa-goldengate-rac-2007111.pdf>

Oracle Real Application Clusters

Oracle Real Application Clusters (Oracle RAC) enables multiple instances that are linked by an interconnect to share access to an Oracle Database. In an Oracle RAC environment, the database runs on two or more systems in a cluster while concurrently accessing a single shared database. The result is a single database that spans multiple hardware systems, enabling Oracle RAC to provide high availability and redundancy during failures in the cluster. Oracle RAC accommodates all system types, from read-only data warehouse systems to update-intensive online transaction processing (OLTP) systems.

Oracle Clusterware

Oracle Clusterware enables servers to communicate with each other, so that they appear to function as a collective unit. This combination of servers is commonly known as a cluster. Although the servers are standalone servers, each server has additional processes that communicate with other servers, which makes the separate servers appear as if they are one system to applications and end users.

Oracle Clusterware provides the infrastructure necessary to run Oracle RAC. Oracle Clusterware also manages resources, such as virtual IP (VIP) addresses, databases, listeners, services, and so on.

Oracle Grid Infrastructure Agents

Oracle Grid Infrastructure Agents (XAG) are Oracle Grid Infrastructure components that provide the high availability (HA) framework to application resources and resource types managed through the agent management interface, AGCTL. This framework provides a complete, ready to use application HA solution that contains pre-defined Oracle Grid Infrastructure resource configurations and agents to integrate applications for complete application HA.

The Oracle Grid Infrastructure Agents provide pre-defined Oracle Clusterware resources for Oracle GoldenGate, Siebel, Oracle PeopleSoft, JD Edwards, and Oracle WebLogic Server, as well as Apache and MySQL applications. Using the agent for Oracle GoldenGate simplifies the creation of dependencies on the source/target database, the application VIP, and the file system (ACFS or DBFS) mount point. The agent command line utility (AGCTL) is used to start and stop Oracle GoldenGate and can also be used to relocate Oracle GoldenGate between the nodes in the cluster.

The latest XAG software should be downloaded from the following location:

<https://www.oracle.com/database/technologies/xag-agents-download.html>

Oracle Database File System (DBFS)

The Oracle Database File System (DBFS) creates a file system interface to files stored in the database. DBFS is similar to NFS in that it provides a shared network file system that looks like a local file system. Because the data is stored in the database, the file system inherits all the high availability and disaster recovery capabilities provided by the Oracle Database.

With DBFS, the server is the Oracle Database. Files are stored as SecureFiles LOBs. PL/SQL procedures implement file system access primitives such as create, open, read, write, and list directory. The implementation of the file system in the database is called the DBFS SecureFiles Store. The DBFS SecureFiles Store allows users to create file systems that can be mounted by clients. Each file system has its own dedicated tables that hold the file system content.

Oracle Automatic Storage Management Cluster File System (ACFS)

The Oracle ASM Cluster File System (ACFS) is a general-purpose industry standard POSIX, X/OPEN, and Windows compliant cluster file system supporting multiple operating systems and server platforms, including Oracle Exadata Database Machine, Oracle Database Appliance (ODA), and traditional server environments. Oracle ACFS

includes advanced features such as file system snapshot, replication, tagging, security, encryption, auditing, and highly available NFS (HANFS) services.

Oracle ACFS makes use of Oracle Automatic Storage Management (ASM) files and inherits ASM features including striping, mirroring, rebalancing, intelligent data placement, preferred read, fast resync, even read, flex ASM, and other features.

Oracle ACFS leverages Oracle Clusterware for cluster membership state transitions and resource-based high availability. Oracle ACFS is bundled into the Oracle Grid Infrastructure (GI) allowing for integrated optimized management of databases, resources, volumes and file systems. Starting with Oracle Grid Infrastructure 12c Release 12.1.0.2, Oracle ACFS is supported on the Oracle Exadata Database Machine running Oracle Linux. Refer to MOS note 1929629.1 for more information.

Configuration Best Practices

Step 1: Create the Database Services

A database service is required so that the Oracle Grid Infrastructure Agent will automatically start the GoldenGate deployment when the database is opened. When DBFS is used for the shared file system, the database service is also used to mount DBFS to the correct RAC instance.

When using a source multitenant database, a separate service is required for the root container database (CDB) and the pluggable database (PDB) that contains the schema being replicated. For a target multitenant database, a single service is required for the PDB.

Create the service using the following command, as the `oracle` user.

```
$ srvctl add service -db <db_name> -service <service name> -preferred <instance_1>
-avaiable <instance_2, instance_3 etc.> -pdb <PDB name>
```

Example:

```
$ srvctl add service -db ggdb -service oggserv_pdb -preferred ggdb1 -available ggdb2
-pdb GGPDB01
```

If not using Oracle Multitenant Database, omit the `-pdb` parameter.

Refer to the Real Application Clusters Administration and Deployment Guide for further details on creating a database service at <https://docs.oracle.com/en/database/oracle/oracle-database/19/racad/server-control-utility-reference.html#GUID-3ED4DBCE-A148-462B-8A79-534A3F0D6E7D>.

Step 2: Create Oracle Net Alias for Oracle GoldenGate Database Connections

To simplify the database connectivity of the Oracle GoldenGate processes when switching between RAC nodes, create a TNS alias on all of the RAC nodes where Oracle GoldenGate may run.

For example:

```
ggdb.local=
  (DESCRIPTION =
    (SDU=2097152)
```

```
(ADDRESS =
  (PROTOCOL=IPC)
  (KEY=LISTENER)
)
(CONNECT_DATA =
  (SERVICE_NAME=oggserv_pdb)
)
)
```

The `SERVICE_NAME` parameter will match the name of the service created in the previous step. If two services were created for a Multitenant Database CDB and PDB, two TNS entries should also be created, one for each service name.

Step 3: Set Up a File System on Oracle RAC

When setting up the configuration, the best practice is to store the Oracle GoldenGate trail files, checkpoint files, bounded recovery (BR) files, and configuration files in DBFS or ACFS to provide the best recoverability and failover capabilities in the event of a system failure. Ensuring the availability of the checkpoint files cluster-wide is essential so that after a failure occurs the Oracle GoldenGate processes can continue running from their last known position. If you are using Oracle Data Guard, DBFS is the recommended file system for use by Oracle GoldenGate due to its seamless integration with the database.

Even though the Oracle GoldenGate installer advises you to install Oracle GoldenGate on shared storage, this is not recommended when configuring a highly available environment, due to forcing downtime during software patching.

Follow the instructions in one of the following sections to configure your chosen file system.

- [Oracle Database File System \(DBFS\)](#)
- [Oracle ASM Cluster File System \(ACFS\)](#)

Oracle Database File System (DBFS)

It is required that you place the DBFS tablespaces inside the database that the Oracle GoldenGate processes are connected to. For example, if an Oracle GoldenGate integrated Extract process is extracting from a database called `ggdb`, the DBFS tablespace would be located in the same `ggdb` database.

Follow instructions in [My Oracle Support note 869822.1](#) to install the required FUSE libraries if they are not already installed. Use the instructions in [My Oracle Support note 1054431.1](#) to configure the database, tablespace, database user, and permissions on source or target GoldenGate environments required for DBFS.

NOTE: When using an Oracle Multitenant Database, the DBFS tablespace MUST be created in a Pluggable Database (PDB). It is recommended that you use the same PDB that the GoldenGate Extract or Replicat processes are connecting to, allowing DBFS to use the same database service, created above in step 1, for its database dependency.

Create a file system for storing the Oracle GoldenGate trail files, checkpoint files, bounded recovery files, discard files, and parameter files. It is recommended that you allocate enough trail file disk space to permit storage of up to 12 hours of trail files, which will provide sufficient space for trail file generation should a problem occur with the

target environment that prevents it from receiving new trail files. The amount of space needed for 12 hours can only be determined by testing trail file generation rates with real production data.

Example file system creation:

```
% cd $ORACLE_HOME/rdbms/admin
% sqlplus dbfs_user/dbfs_password@<database_tns_alias>
SQL> start dbfs_create_filesystem dbfs_gg_tbs goldengate
```

The LOB segment used by DBFS should be configured with the storage options `NOCACHE LOGGING`, which is the default:

```
-- Connect to the DBFS database
SQL> connect system/<passwd>@<database_tns_alias>

-- View current LOB storage:
SQL> SELECT table_name, segment_name, logging, cache
       FROM dba_lobs WHERE tablespace_name='DBFS_GG_TBS';
```

-- More than likely it will be something like this:

```
--
-- TABLE_NAME          SEGMENT_NAME          LOGGING CACHE
-- -----
-- T_GOLDENGATE         LOB_SFSS_FST_73      YES      NO
```

If the LOB segment is not using `NOCACHE LOGGING`, alter it:

```
SQL> ALTER TABLE DBFS.<TABLE_NAME> MODIFY LOB (FILEDATA)
      (NOCACHE LOGGING);
```

-- View the new LOB storage:

```
SQL> SELECT table_name, segment_name, logging, cache
       FROM dba_lobs WHERE tablespace_name='DBFS_GG_TBS';
```

```
-- TABLE_NAME          SEGMENT_NAME          LOGGING CACHE
-- -----
-- T_GOLDENGATE         LOB_SFSS_FST_73      YES      NO
```

Extract uses the `dirtmp` directory to store uncommitted transaction data when its cache (managed by `CACHEMGR`) is full. By placing `dirtmp` on DBFS, you get the additional benefit of larger storage potential. The `dirtmp` directory is only used by the Oracle GoldenGate Extract processes to spill out data once the cache memory area is filled up, so skip the following recommendation if no Extract process is running on this cluster. The contents of `dirtmp` is transient in nature, and therefore not needed by Extract for recovery on startup. If `dirtmp` is used often, and increasing the size of the cache doesn't help, it is recommended that you place `dirtmp` on a separate DBFS file system created with the `NOLOGGING` properties to reduce the I/O created by the temporary file creation.

Create a separate DBFS file system for dirtmp, using a NOCACHE NOLOGGING tablespace. For example:

```
-- Connect to the DBFS database
SQL> connect system/<passwd>@<database_tns_alias>

-- Create the tablespace:
SQL> create bigfile tablespace dbfs_gg_dirtmp_tbs datafile '+DATAC1' size 512g
autoextend on next 8g maxsize 1200g NOLOGGING EXTENT MANAGEMENT LOCAL
AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO;
```

Substitute the size parameters with your required dirtmp storage size.

Create the file system, making sure you are connected to the database as the same user that created the first DBFS file system:

```
% cd $ORACLE_HOME/rdbms/admin
% sqlplus dbfs_user/dbfs_password@<database_tns_alias>
SQL> start dbfs_create_filesystem dbfs_gg_dirtmp_tbs gg_dirtmp
```

The LOB segment used in this file system should be configured with the storage option NOCACHE NOLOGGING, which is the default when the tablespace is created with the NOLOGGING option:

```
-- Connect to the DBFS database
SQL> connect system/<passwd>@<dbfs_tns_alias>

-- View current LOB storage:
SQL> SELECT table_name, segment_name, logging, cache
       FROM dba_lobs WHERE tablespace_name='DBFS_GG_DIRTMP_TBS';

-- More than likely it will be something like this:
--
-- TABLE_NAME          SEGMENT_NAME          LOGGING  CACHE
-- -----
-- T_GG_DIRTMP          LOB_SFSS_FST_73      NO       NO
```

Follow the instructions in [My Oracle Support note 1054431.1](#) for configuring the newly created DBFS file system so that the DBFS instance and mount point resources are automatically started by Cluster Ready Services (CRS) after a node failure, with the following DBFS configuration and script file modifications.

1. Change the mount-dbfs.conf parameters to reflect your database environment.

Modify the MOUNT_OPTIONS parameter to the following:

```
MOUNT_OPTIONS=allow_other,direct_io,failover,nolock
```

The failover option forces all file writes to be committed to the DBFS database in an IMMEDIATE WAIT mode. This prevents data getting lost when it has been written into the dbfs_client cache but not yet written to the database at the time of a database or node failure.

The nolock mount option is required if you are using Oracle Database 18c or later versions, due to a change in the DBFS file locking, which can cause issues for GoldenGate processes after a RAC node failure when a file is currently locked.

If you are using a `dbfs_client` from Oracle Database 12c Release 2 (12.2), make sure you have applied the latest release update that includes the fix for bug 27056711. Once the fix has been applied, the `MOUNT_OPTIONS` should also include the `nolock` option. Refer to [Appendix A, 'Problems with file locking on DBFS'](#) for more information.

2. Modify the `mount-dbfs.sh` script to force unmounting of DBFS when the CRS resource is stopped.

Change two occurrences of:

```
$FUSERMOUNT -u $MOUNT_POINT
```

To the following:

```
$FUSERMOUNT -uz $MOUNT_POINT
```

3. When registering the resource with Oracle Clusterware, be sure to create it as a `cluster_resource` instead of a `local_resource` as specified in the My Oracle Support note. The reason for using `cluster_resource` is so the file system can only be mounted on a single node at one time, preventing mounting of DBFS from concurrent nodes, which creates the potential for concurrent file writes, causing file corruption problems.

Make sure to use the database service name created in a previous step for the DBFS service dependency.

Example:

```
DBNAME=ggdb
DEPNAME=ora.$DBNAME.oggserv_pdb.svc

crsctl add resource $RESNAME \
  -type cluster_resource \
  -attr "ACTION_SCRIPT=$ACTION_SCRIPT, \
        CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, \
START_DEPENDENCIES='hard($DEPNAME)pullup($DEPNAME)', \
        STOP_DEPENDENCIES='hard($DEPNAME)', \
        SCRIPT_TIMEOUT=300"
```

Once the DBFS resource has been created, the file system should be mounted and tested:

```
% crsctl start res dbfs_mount
% crsctl stat res dbfs_mount
```

NOTE: If multiple DBFS file systems, such as a separate `dirtmp` file system, are created by the same user, all of the user owned file systems will be mounted to the `MOUNT_POINT` specified in the `mount-dbfs.conf` file. For example, if a separate file system was created for `dirtmp` (instructions above), after starting the `dbfs_mount` resource the following directories will exist under the same mount point (`MOUNT_POINT=/mnt/dbfs`):

```
/mnt/dbfs/goldengate
/mnt/dbfs/gg_dirtmp
```

After the file system is mounted, create directories for storing the Oracle GoldenGate files:

```
% cd /mnt/dbfs/goldengate
% mkdir dirchk
% mkdir dirprm
% mkdir dirdat
$ mkdir dircrd
$ mkdir dirrpt
% mkdir BR
```

NOTE: Do not place the `dirpcs` directory on the shared file system due to problems that can occur when the process information contained in the process files do not match the host where the process is currently being started after a RAC node failure.

If creating a datastore for use by JAgent of the Oracle GoldenGate Performance Metric Server, the `dirbdb` directory should also be created on local storage.

Create following symbolic links:

```
% cd $OGG_HOME
% ln -s /mnt/dbfs/goldengate/dirprm dirprm
% ln -s /mnt/dbfs/goldengate/dirchk dirchk
% ln -s /mnt/dbfs/goldengate/dirdat dirdat
% ln -s /mnt/dbfs/goldengate/dircrd dircrd
% ln -s /mnt/dbfs/goldengate/dirrpt dirrpt
```

The location of the Extract/Data Pump trail file directory is specified during process creation. For Extract it is also specified in the parameter file with the `EXTTRAIL`.

Oracle ASM Cluster File System (ACFS)

Oracle ACFS is an alternative to DBFS for the shared Oracle GoldenGate files in an Oracle RAC configuration.

Oracle ACFS is supported on the Oracle Exadata Database Machine running Oracle Linux starting with Oracle Grid Infrastructure 12.1.0.2. Refer to [My Oracle Support note 1929629.1](#) for more information about ACFS configuration requirements for Oracle Exadata Database Machine.

In an Oracle Data Guard configuration, it is not recommended that you use ACFS for Oracle GoldenGate due to the lack of integration with the database, and manual intervention required with ACFS replication during a role transition.

Create a single ACFS file system for storing the Oracle GoldenGate trail files, checkpoint files, bounded recovery files, temp files, discard files, and parameter files.

It is recommended that you allocate enough trail file disk space to permit storage of up to 12 hours of trail files, which provides sufficient space for trail file generation should a problem occur with the target environment that prevents it from receiving new trail files. The amount of space needed for 12 hours can only be determined by testing trail file generation rates with real production data.

1. Create the file system using ASMCMD as the Oracle ASM administrator user:

```
ASMCMD [+] > volcreate -G datac1 -s 1200G ACFS_GG
```

NOTE: Modify the file system size according to the determined size requirements.

```
ASMCMD> volinfo -G datac1 acfs_gg
```

```
Diskgroup Name: DATA1
Volume Name: ACFS_GG
Volume Device: /dev/asm/acfs_gg-151
State: ENABLED
Size (MB): 1228800
Resize Unit (MB): 64
Redundancy: MIRROR
Stripe Columns: 8
Stripe Width (K): 1024
Usage:
Mountpath:
```

Make the file system with the following `mkfs` command.

```
$ /sbin/mkfs -t acfs /dev/asm/acfs-gg-151
```

2. Create the CRS resource for the newly created ACFS file system, if not already created.

Check to see if the file system resource was already created:

```
$ srvctl status filesystem -volume ACFS_GG -diskgroup DATA1
ACFS file system /mnt/acfs_gg is mounted on nodes oggadm07, oggadm08
```

If not already created, create the ACFS mount point as the root user:

```
# mkdir -p /mnt/acfs_gg
```

Create the file system resource as the `root` user. Due to the implementation of distributed file locking on ACFS, unlike DBFS, it is acceptable to mount ACFS on more than one RAC node at any one time.

Create the ACFS resource, as the root user:

```
# srvctl add filesystem -device /dev/asm/acfs_gg-151 -volume ACFS_GG
-diskgroup DATA1 -path /mnt/acfs_gg -user oracle -autostart RESTORE
```

To verify the currently configured ACFS file systems, use the following command to view the file system details:

```
$ srvctl config filesystem

Volume device: /dev/asm/acfs_gg-151
Diskgroup name: datac1
Volume name: ACFS_GG
Canonical volume device: /dev/asm/acfs_gg-151
```

```
Accelerator volume devices:
Mountpoint path: /mnt/acfs_gg
Mount point owner: oracle
```

Check the status of the ACFS resource and mount it:

```
$ srvctl status filesystem -volume ACFS_GG -diskgroup DATA1
ACFS file system /mnt/acfs is not mounted
```

```
$ srvctl start filesystem -volume ACFS_GG -diskgroup DATA1
```

The CRS resource that is created will be named using the format

`ora.diskgroup_name.volume_name.acfs`. Using the above file system example, the CRS resource is called `ora.data1.acfs_gg.acfs`.

To see all ACFS file system CRS resources that currently exist, use the following command:

```
$ crsctl stat res -w "TYPE = ora.acfs.type"
```

```
NAME=ora.data1.acfs_gg.acfs
TYPE=ora.acfs.type
TARGET=ONLINE          , ONLINE
STATE=ONLINE on ogadm07, ONLINE on ogadm08
```

3. Create GoldenGate ACFS directories and symbolic links.

After the file system is mounted, create directories for storing the Oracle GoldenGate files:

```
% cd /mnt/acfs_gg
% mkdir dirchk
% mkdir dirprm
% mkdir dirdat
$ mkdir dirtmp
$ mkdir dircrd
$ mkdir dirrpt
% mkdir BR
```

NOTE: Do not place the `dirpcs` directory on the shared file system due to problems that can occur when the process information contained in the process files do not match the host where the process is currently being started. If creating a datastore for use by JAgent of the Oracle GoldenGate Performance Metric Server, the `dirbdb` directory should also be created on local storage.

Create symbolic links for the directories that are required by Oracle GoldenGate to be on shared storage:

```
% cd $GG_HOME
% ln -s /mnt/acfs_gg/dirprm dirprm
% ln -s /mnt/acfs_gg/dirchk dirchk
% ln -s /mnt/acfs_gg/dirdat dirdat
% ln -s /mnt/acfs_gg/dircrd dircrd
```

```
% ln -s /mnt/acfs_gg/dirrpt dirrpt
```

Refer to the Oracle Automatic Storage Management Administrator's Guide for more information about ACFS:

<https://docs.oracle.com/en/database/oracle/oracle-database/19/ostmg/part-acfs-advn.html#GUID-864C6B7E-CB6E-4E82-90CD-6CDD37D0426E>

Step 4: Install Oracle GoldenGate

1. Download the Oracle GoldenGate software from Oracle Technology Network (OTN) at:

<http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html>

2. Install the Oracle GoldenGate software **locally** on all nodes in the Oracle RAC configuration that will be part of the Oracle GoldenGate configuration. Make sure the installation directory is the same on all nodes.

Follow the generic installation instructions detailed in *Installing Oracle GoldenGate*:

<https://docs.oracle.com/en/middleware/goldengate/core/21.3/installing/index.html>

Step 5: Configure Oracle GoldenGate and the Database

The source and target Oracle GoldenGate databases should be configured using the recommendations specified in the MAA technical brief [Oracle GoldenGate Performance Best Practices](#).

Database configuration summary:

- Run the Oracle GoldenGate source database in archive log mode.
- Enable force logging mode if an Extract is running.
- Enable minimal supplemental logging if an Extract is running. Additionally, add schema or table level logging for all replicated objects.
- Configure the streams pool in the System Global Area (SGA).
- Install the Replication Performance Adviser (UTL_RPADV) PL/SQL package to monitor integrated Extract and Replicat performance.

Oracle GoldenGate configuration summary:

- Use integrated capture mode Extract.
- Use Parallel Replicat with `BATCHSQL` enabled to optimize apply rates.
- Implement the GoldenGate heartbeat table to keep track of end-to-end replication lag.

Step 6: Configure GoldenGate Credential Store for Secure Database Login

Oracle recommends setting an explicit connection string in the GoldenGate parameter file using the Oracle GoldenGate credential store or an encrypted password. Oracle does not recommend using plain text password in the parameter files.

The credential alias name `ggconn` is added to the credential store using the following:

```
GGSCI> ALTER CREDENTIALSTORE ADD USER ggadmin@ggdb.local, password
<password>, alias ggconn
```

Example GoldenGate credential store parameter:

```
USERIDALIAS ggconn
```

If using an encrypted password, an example GoldenGate parameter:

```
USERID ggadmin@ggdb.local, password
AACAAAAAAAAAAAAJAUEUGODSCVJEEIUGKJDJTFNDKEJFFFTC AES128, ENCRYPTKEY
securekey1
```

Step 7: Configure Oracle GoldenGate Processes to Use the Shared File System

When creating the Oracle GoldenGate processes, it is important to make sure they are created placing the trail and checkpoint files on the shared storage directories previously created (DBFS or ACFS).

For checkpoint files, using the `dirchk` symbolic link from the Oracle GoldenGate installation directory to DBFS or ACFS ensures that all Oracle GoldenGate processes created will automatically place the files on the shared storage.

For trail files, follow the configuration instructions included below.

Extract Configuration

1. Use the `CACHEMGR` parameter to place the temporary cache files on the shared storage. Although not strictly required on shared storage, doing so often provides more storage capacity to prevent Extract from failing if the `dirtmp` directory fills up:

For DBFS:

```
CACHEMGR CACHEDIRECTORY /mnt/dbfs/goldengate/dirtmp
```

For ACFS:

```
CACHEMGR CACHEDIRECTORY /mnt/acfs_gg/dirtmp
```

2. Configure the Bounded Recovery (BR) files creation on the shared file system such that in an event of a failover, when there are open long running transactions, Extract can use Bounded Recovery to reduce the time to taken to perform recovery. Set the Bounded Recovery file directory parameter:

For DBFS:

```
BR BRDIR /mnt/dbfs/goldengate/BR
```

For ACFS:

```
BR BRDIR /mnt/acfs_gg/BR
```

For more information on Bounded Recovery refer to the *Oracle GoldenGate Windows and UNIX Reference Guide*:

<https://docs.oracle.com/goldengate/c1230/gg-winux/GWURF/br.htm#GWURF410>

3. Set the `EXTTRAIL` Extract parameter (for DBFS):

For DBFS:

```
EXTTRAIL /mnt/dbfs/goldengate/dirdat/aa
```

For ACFS:

```
EXTTRAIL /mnt/acfs_gg/dirdat/aa
```

4. After creating the Extract, use the same `EXTTRAIL` parameter value to add the trail file location.

For DBFS:

```
% ggsci
GGSCI (ggtest.oracle.com) 1> register extract ext_1 database
GGSCI (ggtest.oracle.com) 2> ADD EXTRACT ext_1, INTEGRATED TRANLOG, begin
now
GGSCI (ggtest.oracle.com) 3> ADD EXTTRAIL /mnt/dbfs/goldengate/dirdat/aa,
EXTRACT ext_1, Megabytes 500
```

For ACFS:

```
% ggsci
GGSCI (ggtest.oracle.com) 1> register extract ext_1 with database
GGSCI (ggtest.oracle.com) 2> ADD EXTRACT ext_1, INTEGRATED TRANLOG, begin
now
GGSCI (ggtest.oracle.com) 3> ADD EXTTRAIL /mnt/acfs_gg/dirdat/aa, EXTRACT
ext_1, Megabytes 500
```

Further instructions about creating an Extract process are available in the *Using Oracle GoldenGate with Oracle Database* at <https://docs.oracle.com/en/middleware/goldengate/core/21.3/oracle-db/configuring-extract.html#GUID-74E23910-A441-4939-BDFD-FE0B2EE9783F>.

Data Pump Configuration

1. Set the `RMTHOST` Data Pump parameter to the hostname that will be used for connecting to the target system. Below in Step 8: Oracle Clusterware Configuration, the Application Virtual IP address is created with Cluster Ready Services (CRS) so that a single IP address can be moved between compute nodes. Data Pump can then continue to connect to the target host when it moves between RAC nodes:

```
RMTHOST gg_target_VIP, MGRPORT 8901
```

2. Set the `RMTRAIL` Data Pump parameter to the trail file location on the target host:

For DBFS:

```
RMTTRAIL /mnt/dbfs/goldengate/dirdat/aa
```

For ACFS:

```
RMTTRAIL /mnt/acfs_gg/dirdat/aa
```

3. Create a Data Pump process, specifying the trail file location on the source host:

For DBFS:

```
% ggsci
GGSCI (ggtest.oracle.com) 1> ADD EXTRACT dpump_1, EXTTRAILSOURCE
/mnt/dbfs/goldengate/dirdat/aa
```

For ACFS:

```
% ggsci
GGSCI (ggtest.oracle.com) 1> ADD EXTRACT dpump_1, EXTTRAILSOURCE
/mnt/acfs_gg/dirdat/aa
```

4. Use the ADD RMTTRAIL command to specify the remote trail file location on the target host:

For DBFS:

```
% ggsci
GGSCI (ggtest.oracle.com) 1> ADD RMTTRAIL /mnt/dbfs/goldengate/dirdat/aa
EXTRACT dpump_1, MEGABYTES 500
```

For ACFS:

```
% ggsci
GGSCI (ggtest.oracle.com) 1> ADD RMTTRAIL /mnt/acfs_gg/dirdat/aa EXTRACT
dpump_1, MEGABYTES 500
```

Further instructions about creating the Data Pump process are available in the *Using Oracle GoldenGate with Oracle Database* at <https://docs.oracle.com/en/middleware/goldengate/core/21.3/oracle-db/configuring-extract.html#GUID-48D4EB1B-C059-4D6F-A130-734D41604BA8>.

Replicat Configuration

1. Set the EXTTRAIL Replicat parameter, as follows:

For DBFS:

```
EXTTRAIL /mnt/dbfs/goldengate/dirdat/aa
```

For ACFS:

```
EXTTRAIL /mnt/acfs_gg/dirdat/aa
```

2. When adding the Replicat, use the same EXTTRAIL parameter to specify the trail file location:

For DBFS:

```
% ggsci
```

```
GGSCI (ggtest.oracle.com) 1> ADD REPLICAT rep_1, PARALLEL, EXTTRAIL  
/mnt/dbfs/goldengate/dirdat/aa
```

For ACFS:

```
GGSCI (ggtest.oracle.com) 1> ADD REPLICAT rep_1, PARALLEL, EXTTRAIL  
/mnt/acfs_gg/dirdat/aa
```

3. If there is an Oracle GoldenGate Data Pump sending trail files to this target host, where Replicat is running, you must add the following parameter to the GLOBALS parameter file located in the Oracle GoldenGate installation directory:

For DBFS:

```
ALLOWOUTPUTDIR /mnt/dbfs/goldengate/dirdat
```

For ACFS:

```
ALLOWOUTPUTDIR /mnt/acfs_gg/dirdat
```

The Manager process should be restarted after modifying the GLOBALS parameter file.

Be careful to specify the ALLOWOUTPUTDIR parameter correctly. If the parameter value specifies a directory that does not exist, the Data Pump sending trail files to the correct directory will fail with the following error:

```
WARNING OGG-01221 Connect failed to <target host IP address>:7819, error  
111:Connection refused.
```

Step 8: Configure Autostart of Extract, Data Pump, and Replicat Processes

Configure the Extract, Data Pump, and Replicat processes to automatically start when the Manager process is started. Add the following parameter to the Manager parameter file (mgr.prm) located in the \$OGG_HOME/dirprm directory:

```
AUTOSTART ER *  
AUTORESTART ER *, RETRIES 20, WAITSECONDS 15, RESETMINUTES 15
```

If there are some Oracle GoldenGate processes that are not required for automatic restarting, instead of using the Oracle GoldenGate process name wildcard (*), explicitly name the processes you want to be restarted automatically by the Manager process.

```
AUTOSTART EXTRACT EXT_1  
AUTOSTART EXTRACT DPUMP_1  
AUTOSTART REPLICAT REP_2  
AUTORESTART EXTRACT EXT_1  
AUTORESTART EXTRACT DPUMP_1  
AUTORESTART EXTRACT REP_2
```

Step 9: Oracle Clusterware Configuration

Configure Oracle Clusterware to manage Oracle GoldenGate using the Oracle Grid Infrastructure Standalone Agent (XAG). Using XAG, automates the mounting of the shared file system (DBFS or ACFS) as well as the stopping and starting of the Oracle GoldenGate processes when relocating between Oracle RAC nodes.

1. Install the latest Oracle Grid Infrastructure Standalone Agent

Download the latest XAG software from <https://www.oracle.com/database/technologies/xag-agents-download.html>

It is a requirement to use XAG version 10.2 or newer.

The XAG software **must** be installed in a directory that is outside of the Oracle Grid Infrastructure home directory to avoid conflicts with software patching. XAG needs to be installed in the same directory on all RAC nodes in the cluster where Oracle GoldenGate is installed.

For example, as the `oracle` user:

```
$ ./xagsetup.sh --install --directory /u01/oracle/xag --all_nodes
```

2. Add the location of the newly installed XAG software to the PATH variable.

```
$ cat .bashrc
export PATH=/u01/oracle/xag/bin:$PATH
```

NOTE: It is important to make sure the XAG directory is specified BEFORE the Grid Infrastructure directory to ensure the correct `agctl` binary is found.

3. Prepare for creating Application Virtual IP Address (VIP).

A dedicated application VIP is required on a GoldenGate target host to ensure the Oracle GoldenGate Data Pumps can communicate with it, regardless of which RAC node is hosting Oracle GoldenGate. The VIP is a cluster resource that Oracle Clusterware manages. The VIP is assigned to a cluster node and will be migrated to another node in the event of a node failure. This allows Oracle GoldenGate data pump to continue transferring data to the newly assigned target node.

NOTE: The application VIP is only required on clusters where one or more GoldenGate Replicat processes are running, that receives GoldenGate trail files from a Data Pump.

There are two pieces of information needed before creating the application VIP:

a. The network number can be identified by the following command:

```
$ crsctl status resource -p -attr NAME,USR_ORA_SUBNET -w "TYPE =
ora.network.type" |sort | uniq

NAME=ora.net1.network

USR_ORA_SUBNET=10.133.16.0
```

The `net1` in `NAME=ora.net1.network` indicates this is network 1.

- b. The IP address provided by your system administrator for the new Application VIP. This IP address must be in the same subnet of the cluster environment as determined above.

The VIP will be created in the next step, when configuring the Grid Infrastructure Agent.

4. Configure Oracle Grid Infrastructure Agent (XAG).

Oracle GoldenGate must be registered with XAG so that the processes are started and stopped automatically when the database is started and the file system is mounted.

Below are some examples of registering Oracle GoldenGate with XAG.

- a. Source RAC cluster, using DBFS for the shared file system, and no need for an application VIP:

```
agctl add goldengate GG_SOURCE \  
--gg_home /u01/oracle/goldengate \  
--oracle_home /u01/app/oracle/product/19.0.0.0/dbhome_1 \  
--db_services ora.ggdb.oggserv_cdb.svc \  
--use_local_services --filesystems dbfs_mount
```

Where:

`--db_services` specifies the service name which was created in a previous step. If using Oracle Multitenant Database, specify the PDB database service for Replicat, or the CDB database service for an Extract. If using both Replicat and Extract, specify both services names, separated by a comma.

`--use_local_services` specifies that the Oracle GoldenGate instance must be co-located on the same RAC node where the `db_services` service is running.

`--filesystems` specifies CRS resource name used to mount and unmount the DBFS file system, as detailed above in [step 3](#).

- b. Target RAC cluster, using ACFS, with an application VIP running on a subset of the nodes in the cluster (run as `root` because the Application VIP is created):

```
agctl add goldengate GG_TARGET \  
--gg_home /u01/oracle/goldengate \  
--oracle_home /u01/app/oracle/product/19.1.0.0.0/dbhome_1 \  
--db_services ora.ggdb.oggserv_pdb.svc --use_local_services \  
--filesystems ora.data1.acfs_gg.acfs --network 1 \  
--ip 10.133.16.11 --user oracle --group oinstall \  
--nodes nshb01gg05,nshb01gg06
```

Where:

`--db_services` specifies the service name which was created above in step 4.

`--use_local_services` specifies that the Oracle GoldenGate instance must be co-located on the same RAC node where the `db_services` service is running.

--filesystems specifies CRS resource name used to mount and unmount the ACFS file system, as detailed above in [step 3](#).

--network specifies the network subnet for the VIP, determined above. This parameter is not needed if you are using a previously created VIP.

--ip specified the IP address for the VIP, which was determined above. If you already have a previously created VIP, use the --vip_name <vip_name> parameter instead of --ip.

--user and --group specifies the Oracle GoldenGate user is oracle in the group oinstall. These parameters are required when registering Oracle GoldenGate as the root user.

--nodes specifies which of the RAC nodes this Oracle GoldenGate instance can run on. If GoldenGate is configured to run on any of the RAC nodes in the cluster, this parameter can be omitted.

After registering Oracle GoldenGate with XAG, add the following parameter to the GLOBALS file found in the Oracle GoldenGate installation directory:

```
XAGENABLE
```

A restart is required for the Manager to re-read the change GLOBALS file. Stop the Oracle GoldenGate processes, including Manager, using the Oracle GoldenGate command line tool GGSCI. Once this is complete, starting and stopping all of the Oracle GoldenGate processes can either be carried out using XAG AGCTL command, or GGSCI. If there is a requirement to stop and start a single Oracle GoldenGate process, use GGSCI. The XAG command tool, AGCTL, will only start and stop processes named in the Manager parameter file.

Example GGSCI output:

```
GGSCI (nshb01gg05.us.oracle.com) 1> info all
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt	XAG
MANAGER	STOPPED				MANAGED/xag.GG01.goldengate/
REPLICAT	STOPPED	REP1A	00:00:00	00:25:54	NON-CRITICAL

```
GGSCI (nshb01gg05.us.oracle.com) 2> start mgr
```

```
Transparent Integration with XAG is enabled. Sending the command START MANAGER to XAG...
```

```
2018-03-06 14:55:50 INFO OGG-02407 CRS/XAG status: CRS-2676: Start of 'ora.ogg.db' on ' nshb01gg05' succeeded.
```

```
2018-03-06 14:55:50 INFO OGG-02407 CRS/XAG status: CRS-2676: Start of 'xag.GG01.goldengate' on ' nshb01gg05' succeeded.
```

```
GGSCI (nshb01gg05.us.oracle.com) 3> info all
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt	XAG
MANAGER	RUNNING				MANAGED/xag.GG01.goldengate/
REPLICAT	RUNNING	REP1A	00:00:01	00:00:04	NON-CRITICAL

When using the XAGENABLE parameter, GGSCI INFO ALL shows the current CRS resource name that is created by registering Oracle GoldenGate with XAG. If the GoldenGate manager is not yet registered with XAG, GGSCI will show:

```
GGSCI (nshb01gg05.us.oracle.com) 3> info all
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt	XAG
MANAGER	RUNNING				NOT MANAGED
REPLICAT	RUNNING	REP1A	00:00:01	00:00:07	

Below are some example AGCTL commands that are used to manage Oracle GoldenGate with XAG.

To check the status of Oracle GoldenGate:

```
% agctl status goldengate
Goldengate instance 'GG_Source' is running on dbm01db06
```

To start Oracle GoldenGate manager, and all processes that have autostart enabled:

```
% agctl start goldengate GG_Target --node nshb01gg06
```

To stop all Oracle GoldenGate processes:

```
% agctl stop goldengate GG_Target
```

To manually relocate Oracle GoldenGate to another node:

```
% agctl relocate goldengate GG_Source --node dbm01db05
```

NOTE: The Oracle GoldenGate resource **MUST** be running before relocating it.

To view the configuration parameters for the Oracle GoldenGate resource:

```
% agctl config goldengate GG_Target

Application GoldenGate location is: /u01/oracle/goldengate
Goldengate MicroServices Architecture environment: no
GoldenGate instance type is: dual
EXTRACT groups to monitor:
REPLICAT groups to monitor:
Critical EXTRACT groups:
Critical REPLICAT groups:
Autostart on DataGuard role transition to PRIMARY: no
Autostart JAgent: no
Configured to run on Nodes: nshb01gg05 nshb01gg06
ORACLE_HOME location is: /u01/app/oracle/product/19.0.0.0/dbhome_1
```



Databases needed: ora.ggdb.oggserv_pdb.svc
File System resources needed: ora.datacl.acfs_gg.acfs

To view more detailed clusterware configuration information:

```
% crsctl stat res -w "NAME = xag.GG_Source.goldengate" -p
```

To delete the Oracle GoldenGate resource, stop the processes and then delete the resource:

```
% actl stop goldengate GG_Source  
% agctl remove goldengate GG_Source
```

Further information on the Oracle Grid Infrastructure Bundled Agent:

<https://www.oracle.com/database/technologies/xag-agents-download.html>

Summary of Recommendations When Deploying on Oracle RAC

When configuring Oracle GoldenGate in an Oracle RAC environment, follow these recommendations:

- Install the latest version of Oracle GoldenGate software locally on each RAC node.
- Use the Oracle Database File System (DBFS) or Oracle ASM Cluster File System (ACFS) for the file system where the GoldenGate files are stored (trail, checkpoint, temporary, report, parameter files).
- Do not place the GoldenGate `dirpcs` directory on shared storage.
- Use the same DBFS or ACFS mount point on all of the RAC nodes that may run Oracle GoldenGate.
- Install the latest version of the Grid Infrastructure agent (XAG) on all RAC nodes that will run Oracle GoldenGate (version or newer).
- Configure the GoldenGate Manager process to start and restart all GoldenGate processes.



References

- [Oracle GoldenGate 21c Documentation](#)
- [Oracle Database SecureFiles and Large Object Developer's Guide 19c \(DBFS\)](#)
- [Oracle Automatic Storage Management Cluster File System Administrator's Guide 19c \(ACFS\)](#)
- [Oracle GoldenGate Performance Best Practices MAA Technical Brief](#)
- [Transparent Role Transitions with Oracle Data Guard and Oracle GoldenGate](#)
- Oracle Maximum Availability Architecture Web site
<http://www.otn.oracle.com/goto/maa>

Appendix A: Troubleshooting Oracle GoldenGate on Oracle RAC

There may be occasions when Oracle GoldenGate processes are not successfully started on a RAC node. There are number of files are generated by GoldenGate, XAG, and CRS that should be reviewed to determine the cause of the problem.

Below is a list of important log and trace files, along with their example locations and some example output.

1. XAG log file

Location: <XAG installation directory>/log/<hostname>

Example location: /u01/oracle/xag/log/dbm01db05

File name: agctl_goldengate_oracle.trc

Contains all commands executed with AGCTL along with the output from the commands, including those commands that are executed by CRS.

Example:

```
2017-12-06 10:44:06: agctl start goldengate GG01 --node dbm01db05
2017-12-06 10:44:06: executing cmd: /u01/app/12.2.0/grid/bin/crsctl status res
xag.GG01.goldengate
2017-12-06 10:44:06: executing cmd: /u01/app/12.2.0/grid/bin/crsctl status res
xag.GG01.goldengate -f
2017-12-06 10:44:06: executing cmd: /u01/app/12.2.0/grid/bin/crsctl start
resource xag.GG01.goldengate -f -n dbm01db05
2017-12-06 10:44:16: Command output:
> CRS-2672: Attempting to start 'xag.GG01-vip.vip' on 'dbm01db05'
> CRS-2676: Start of 'xag.GG01-vip.vip' on 'dbm01db05' succeeded
> CRS-2672: Attempting to start 'xag.GG01.goldengate' on 'dbm01db05'
> CRS-2676: Start of 'xag.GG01.goldengate' on 'dbm01db05' succeeded
>End Command output
```

2. CRS trace file

Location: /u01/app/oracle/diag/crs/<hostname>/crs/trace

Example location: /u01/app/oracle/diag/crs/dbm01db05/crs/trace

File name: crsd_scriptagent_oracle.trc

Contains the output created by any CRS resource action scripts, like XAG or dbfs_mount. This trace file is crucial to determining why DBFS or GoldenGate did not start on a RAC node.

Example:

```
2018-03-05 12:59:39.096 : AGFW:714880768: {1:46407:39491} Agent received the
message: RESOURCE_START[xag.GG01.goldengate 1 1] ID 4098:1372229
2018-03-05 12:59:39.096 : AGFW:714880768: {1:46407:39491} Preparing START
command for: xag.GG01.goldengate 1 1
2018-03-05 12:59:39.096 : AGFW:714880768: {1:46407:39491}
xag.GG01.goldengate 1 1 state changed from: UNKNOWN to: STARTING
2018-03-05 12:59:39.096 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] Executing action script:
/u01/oracle/xag/bin/aggoldengateas[start]
```

```

2018-03-05 12:59:39.248 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] GG agent running command 'start'
on xag.GG01.goldengate
2018-03-05 12:59:39.498 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] Checking if mgr is running
2018-03-05 12:59:39.498 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] MGR.pcm is not found - check if
MGR pid is running
2018-03-05 12:59:39.548 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] Cleaning up GG process files
2018-03-05 12:59:41.051 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] Manager started.
2018-03-05 12:59:41.051 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start]
2018-03-05 12:59:41.051 :CLSDYNAM:723285760:
[xag.GG01.goldengate]{1:46407:39491} [start] Waiting for mgr to start
...
2018-03-05 12:59:41.654 : AGFW:714880768: {1:46407:39491} Started implicit
monitor for [xag.GG01.goldengate 1 1] interval=30000 delay=30000
2018-03-05 12:59:41.654 : AGFW:714880768: {1:46407:39491} Agent sending last
reply for: RESOURCE_START[xag.GG01.goldengate 1 1] ID 4098:1372229
2018-03-05 12:59:54.446 :CLSDYNAM:723285760: [dbfs_mount]{1:46407:39408}
[check] Executing action script: /u01/oracle/scripts/mount-dbfs.sh[check]
2018-03-05 12:59:54.497 :CLSDYNAM:723285760: [dbfs_mount]{1:46407:39408}
[check] Checking status now
2018-03-05 12:59:54.497 :CLSDYNAM:723285760: [dbfs_mount]{1:46407:39408}
[check] Check -- ONLINE

```

3. GoldenGate report files

Location: \$OGG_HOME/dirrpt

Example location: /u01/oracle/goldengate/gg123/dirrpt/*.rpt

The GoldenGate report files contain important messages and errors for all of the GoldenGate processes, including the Manager processes. If any of the GoldenGate processes fail to start or abend when they are running, the process report file will contain important information which can be used to determine the cause of the failure.

Example errors from an Extract report file:

```

2018-02-23 13:01:50 ERROR OGG-00446 Unable to lock file
"/u01/oracle/goldengate/gg123/dirchk/EXT_1A.cpe" (error 95, Operation not
supported).
2018-02-23 13:01:50 ERROR OGG-01668 PROCESS ABENDING.

```

Example Configuration Problems

Below are some examples of configuration problems that can be encountered with GoldenGate in a RAC environment along with how to diagnose and resolve them.

1. Incorrect parameter settings in the mount-dbfs.conf file

When XAG fails to mount DBFS the failure will be reported either on the command line (if you are running the manual agctl command) or in the XAG log file:

```
> CRS-2673: Attempting to stop 'xag.GG01.goldengate' on 'nshb01gg05'
```

```

> CRS-2677: Stop of 'xag.GG01.goldengate' on 'nshb01gg05' succeeded
> CRS-2673: Attempting to stop 'dbfs_mount' on 'nshb01gg05'
> CRS-2677: Stop of 'dbfs_mount' on 'nshb01gg05' succeeded
> CRS-2672: Attempting to start 'dbfs_mount' on 'nshb01gg06'
> CRS-2674: Start of 'dbfs_mount' on 'nshb01gg06' failed
> CRS-2679: Attempting to clean 'dbfs_mount' on 'nshb01gg06'
> CRS-2681: Clean of 'dbfs_mount' on 'nshb01gg06' succeeded
> CRS-2564: Failed to relocate resource 'xag.GG01.goldengate'. Will attempt to
restore it on 'nshb01gg05' now.
> CRS-2672: Attempting to start 'dbfs_mount' on 'nshb01gg05'
> CRS-2676: Start of 'dbfs_mount' on 'nshb01gg05' succeeded
> CRS-2672: Attempting to start 'xag.GG01.goldengate' on 'nshb01gg05'
> CRS-2676: Start of 'xag.GG01.goldengate' on 'nshb01gg05' succeeded
> CRS-4000: Command Relocate failed, or completed with errors.

```

Next, check the CRS trace file (crsd_scriptagent_oracle.trc) which shows the reason why DBFS failed to mount. Below are some example errors caused by incorrect parameter settings in the mount-dbfs.conf file.

i. Incorrect DBNAME

```

2018-02-21 19:58:35.656 :CLSDYNAM:4191692544: [dbfs_mount]{2:50354:2216}
[start] ORACLE_SID is
2018-02-21 19:58:35.656 :CLSDYNAM:4191692544: [dbfs_mount]{2:50354:2216}
[start] No running ORACLE_SID available on this host, exiting
2018-02-21 19:58:35.656 : AGFW:4191692544: {2:50354:2216} Command: start for
resource: dbfs_mount 1 1 completed with invalid status: 2

```

ii. Incorrect MOUNT_POINT

```

2018-02-21 20:02:25.970 :CLSDYNAM:3968993024: [dbfs_mount]{2:50354:2264}
[start] fuse: bad mount point '/mnt/dbfs_gg': No such file or directory
2018-02-21 20:02:26.972 :CLSDYNAM:3968993024: [dbfs_mount]{2:50354:2264}
[start] Start -- OFFLINE
2018-02-21 20:02:26.972 : AGFW:3968993024: {2:50354:2264} Command: start for
resource: dbfs_mount 1 1 completed with status: FAIL

```

iii. Incorrect DBFS_USER or DBFS_PASSWD

```

2018-02-21 20:03:42.678 :CLSDYNAM:951891712: [dbfs_mount]{2:50354:2332} [start]
Fail to connect to database server. Error: ORA-01017: invalid
username/password; logon denied
2018-02-21 20:03:42.678 :CLSDYNAM:951891712: [dbfs_mount]{2:50354:2332} [start]
2018-02-21 20:03:43.779 :CLSDYNAM:951891712: [dbfs_mount]{2:50354:2332} [start]
Start -- OFFLINE
2018-02-21 20:03:43.779 : AGFW:951891712: {2:50354:2332} Command: start for
resource: dbfs_mount 1 1 completed with status: FAIL

```

iv. Incorrect ORACLE_HOME

```

2018-02-23 11:46:17.720 :CLSDYNAM:1073739520: [dbfs_mount]{2:50354:45006}
[start] /u01/oracle/scripts/mount-dbfs.sh: line 139:
/u01/app/oracle/product/12.2.0.1/dbhome/bin/srvctl: No such file or directory
2018-02-23 11:46:17.720 :CLSDYNAM:1073739520: [dbfs_mount]{2:50354:45006}
[start] ORACLE_SID is
2018-02-23 11:46:17.720 :CLSDYNAM:1073739520: [dbfs_mount]{2:50354:45006}
[start] No running ORACLE_SID available on this host, exiting

```

```
2018-02-23 11:46:17.720 : AGFW:1073739520: {2:50354:45006} Command: start
for resource: dbfs_mount 1 1 completed with invalid status: 2
```

To resolve these configuration issues, set the correct parameter value in `mount-dbfs.conf`.

2. GoldenGate is not stopping when requested by XAG

When XAG fails to stop GoldenGate it will be reported either on the command line (if you are running the manual `agctl` command) or in the XAG log file:

```
CRS-2673: Attempting to stop 'xag.GG01.goldengate' on 'nshb01gg05'
CRS-2675: Stop of 'xag.GG01.goldengate' on 'nshb01gg05' failed
CRS-2679: Attempting to clean 'xag.GG01.goldengate' on 'nshb01gg05'
CRS-2678: 'xag.GG01.goldengate' on 'nshb01gg05' has experienced an
unrecoverable failure
CRS-2731: Resource 'xag.GG01.goldengate' is already running on server
'nshb01gg05'
CRS-4000: Command Relocate failed, or completed with errors.
```

The CRS trace (`crsd_scriptagent_oracle.trc`) shows:

```
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] Program      Status      Group
Lag at Chkpt   Time Since Chkpt
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] MANAGER      RUNNING
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] EXTRACT      RUNNING      DPUMP_1A
00:00:00      00:33:26
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] EXTRACT      RUNNING      EXT_1A
00:00:07      00:05:07
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] REPLICAT     RUNNING      REP_1
00:00:00      00:05:04
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop]
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] ER still running REPLICAT:REP_1
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] ER still running EXTRACT:EXT_1A
2018-02-22 12:36:10.706 :CLSDYNAM:3919890176:
[xag.GG01.goldengate]{1:46407:20093} [stop] ER still running EXTRACT:DPUMP_1A
2018-02-22 12:36:10.929 : AGENT:3921991424: {1:46407:20093} {1:46407:20093}
Created alert : (:CRSAGF00113:) : Aborting the command: stop for resource:
xag.GG01.goldengate 1 1
2018-02-22 12:36:10.929 :CLSDYNAM:3921991424:
[xag.GG01.goldengate]{1:46407:20093} [stop] Killing action script: stop
2018-02-22 12:36:10.929 : AGFW:3921991424: {1:46407:20093} Command: stop for
resource: xag.GG01.goldengate 1 1 has been aborted, updating resource
state/label
2018-02-22 12:36:10.929 : AGFW:3917788928: {1:46407:20093} Processing
aborted reply : stop for resource: xag.GG01.goldengate 1 1
```

Check the GoldenGate process report files to see if there are errors reported when the STOP is issued by XAG. In this example, none of the report files shows that the 'STOP' was issued for the processes.

When checking with `ggsci`, all the processes report that they are currently running. When a manual stop command is issued an error is reported:

```
GGSCI (nshb01gg05.us.oracle.com) 1> info all
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt	XAG
MANAGER	RUNNING				MANAGED/xag.GG01.goldengate/
EXTRACT	RUNNING	DPUMP_1A	00:00:00	00:45:06	NON-CRITICAL
EXTRACT	RUNNING	EXT_1A	00:00:07	00:16:47	NON-CRITICAL
REPLICAT	RUNNING	REP_1	00:00:00	00:16:44	NON-CRITICAL

```
GGSCI (nshb01gg05.us.oracle.com) 2> stop er *
```

```
Sending STOP request to EXTRACT DPUMP_1A ...
```

```
2018-02-22 12:47:54 ERROR OGG-15161 Could not initialize the connection with EXTRACT DPUMP_1A (Connection refused).
```

```
Sending STOP request to EXTRACT EXT_1A ...
```

```
2018-02-22 12:47:54 ERROR OGG-15161 Could not initialize the connection with EXTRACT EXT_1A (Connection refused).
```

```
Sending STOP request to REPLICAT REP_1 ...
```

```
2018-02-22 12:47:54 ERROR OGG-15161 Could not initialize the connection with REPLICAT REP_1 (Connection refused).
```

This implies the processes might not be running. Checking the processes at the operating system confirms this:

```
$ ps -ef|grep goldengate
oracle 107593      1 0 12:02 ?          00:00:02 ./mgr PARAMFILE
/u01/oracle/goldengate/gg123/dirprm/mgr.prm REPORTFILE
/u01/oracle/goldengate/gg123/dirrpt/MGR.rpt PROCESSID MGR
oracle 181935 391570 0 12:49 pts/2    00:00:00 grep goldengate
```

The only GoldenGate process running is the Manager process.

A common reason for GGSCI reporting a process running when in fact it isn't, is that a process file exists in the `dirpcs` directory without a running process. The `dirpcs` directory contains the following files:

```
$ ls -lrt
total 2
-rw-r----- 1 oracle oinstall 74 Feb 22 12:02 MGR.pcm
-rw-r----- 1 oracle oinstall 67 Feb 22 12:02 DPUMP_1A.pce
-rw-r----- 1 oracle oinstall 65 Feb 22 12:09 REP_1.pcr
-rw-r----- 1 oracle oinstall 65 Feb 22 12:09 EXT_1A.pce
```

If the DBFS resource has been configured correctly, as instructed earlier in this paper, the file cannot be mounted on more than one RAC node at one time. This prevents GoldenGate processes concurrently on multiple nodes which can cause problems. Check all of the RAC nodes to make sure GoldenGate processes that match the process files

in `dirpcs` (`DPUMP_1A`, `EXT_1A` and `REP_1` in this example) are not running on any of the nodes. If they are running on another node, you need to stop the processes and then try restarting on the correct RAC node.

If the GoldenGate processes are not running on another RAC node, remove the process files from `dirpcs`, except the Manager process which is still running:

```
$ rm DPUMP_1A.pce REP_1.pcr EXT_1A.pce
```

Checking the XAG GoldenGate status will now show `INTERMEDIATE` status because only the Manager process is running:

```
$ agctl status goldengate
Goldengate instance 'GG01' INTERMEDIATE on nshb01gg05
```

Now, XAG will be able to stop or relocate GoldenGate without problems:

```
$ agctl relocate goldengate GG01
CRS-2673: Attempting to stop 'xag.GG01.goldengate' on 'nshb01gg05'
CRS-2677: Stop of 'xag.GG01.goldengate' on 'nshb01gg05' succeeded
CRS-2673: Attempting to stop 'dbfs_mount' on 'nshb01gg05'
CRS-2677: Stop of 'dbfs_mount' on 'nshb01gg05' succeeded
CRS-2672: Attempting to start 'dbfs_mount' on 'nshb01gg06'
CRS-2676: Start of 'dbfs_mount' on 'nshb01gg06' succeeded
CRS-2672: Attempting to start 'xag.GG01.goldengate' on 'nshb01gg06'
CRS-2676: Start of 'xag.GG01.goldengate' on 'nshb01gg06' succeeded
```

3. Problems with file locking on DBFS

Not an issue with Oracle 19c databases.

If the Oracle Database 12c Release 2 (12.2) and the `nolock` DBFS mount option is not used, there can be problems with GoldenGate processes trying to lock checkpoint or trail files. The same problem will be encountered if using Oracle Database 11g Release 2 (11.2.0.4) or 12c Release 1 (12.1) with patch for bug 22646150 applied. This patch changes the way in which file locking is handled by DBFS to match Oracle Database 12c Release 2 (12.2). In order to add the `nolock` DBFS mount option, patch for bug 27056711 must be applied to the database. If patch for bug 22646150 has not been applied to the database, the patch for bug 27056711 and the `nolock` mount option is not required.

Below is an example of how to diagnose a GoldenGate locking problem.

After starting GoldenGate the status is `INTERMEDIATE`:

```
$ agctl status goldengate
Goldengate instance 'GG01' INTERMEDIATE on nshb01gg06
```

First, check the CRS trace file (`crsd_scriptagent_oracle.trc`) for errors:

```
2018-02-23 12:08:04.138 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] Manager is running (IP port
nshb01gg06.us.oracle.com.7809, Process ID 216140).
2018-02-23 12:08:04.138 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check]
2018-02-23 12:08:04.338 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check]
```

```

2018-02-23 12:08:04.338 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] Program      Status      Group
Lag at Chkpt Time Since Chkpt XAG
2018-02-23 12:08:04.338 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] MANAGER      RUNNING
NOT MANAGED
2018-02-23 12:08:04.338 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] EXTRACT      STOPPED     DPUMP_1A
00:00:00      00:22:23
2018-02-23 12:08:04.338 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] EXTRACT      STOPPED     EXT_1A
00:00:00      00:22:22
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] REPLICAT     STOPPED     REP_1
00:00:00      00:22:13
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check]
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] EXTRACT:DPUMP_1A not running
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] monitored ER DPUMP_1A not running
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] EXTRACT:EXT_1A not running
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] monitored ER EXT_1A not running
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] REPLICAT:REP_1 not running
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] monitored ER REP_1 not running
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] ER(s) not running :
DPUMP_1A,EXT_1A,REP_1
2018-02-23 12:08:04.339 :CLSDYNAM:4080445184:
[xag.GG01.goldengate]{2:50354:45079} [check] CRS_STATE_DETAILS:ER(s) not
running : DPUMP_1A,EXT_1A,REP_1

```

Next, check the GoldenGate report files of the stopped processes. It is possible that the GoldenGate report files don't show the startup error. If this is the case, use `ggsci` to try restarting the stopped processes, and then recheck the report files.

For example, the report file for Extract `EXT_1A` shows:

```

2018-02-23 13:01:50 ERROR   OGG-00446  Unable to lock file
"/u01/oracle/goldengate/gg123/dirchk/EXT_1A.cpe" (error 95, Operation not
supported).
2018-02-23 13:01:50 ERROR   OGG-01668  PROCESS ABENDING.

```

Check the DBFS mount options:

```
$ ps -ef|grep dbfs_client
```

```

oracle  216037      1  0 12:04 ?          00:00:05
/u01/app/oracle/product/12.2.0.1/dbhome_1/bin/dbfs_client dbfs@ -o
allow_other,direct_io,failover /mnt/dbfs

```

It is clear the `nolock` mount option was not used, which leads to the `Operation not supported` error.



Change the `mount-dbfs.conf` file to add the `nolock` mount option:

```
MOUNT_OPTIONS=allow_other,direct_io,failover,nolock
```

Stop GoldenGate, unmount DBFS and restart GoldenGate:

```
$ agctl stop goldengate GG01  
$ crsctl stop res dbfs_mount  
$ agctl start goldengate GG01
```

Finally, check the status of GoldenGate:

```
$ agctl status goldengate  
Goldengate instance 'GG01' is running on nshb01gg06
```



Oracle GoldenGate Classic Architecture with Oracle
Real Application Clusters Configuration
October 2022
Author: Stephan Haisley

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  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

Integrated Cloud Applications & Platform Services

Copyright © 2018, 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. 0116

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