

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

ORACLE WHITE PAPER | AUGUST 2018

| | |
|---|----|
| Introduction | 1 |
| Configuration Overview | 2 |
| Oracle GoldenGate | 2 |
| Oracle Real Application Clusters | 4 |
| Oracle Clusterware | 4 |
| Oracle Grid Infrastructure Agents | 4 |
| Oracle Database File System (DBFS) | 4 |
| Oracle Automatic Storage Management Cluster File System (ACFS) | 5 |
| Configuration Best Practices | 5 |
| Step 1: Set Up a File System on Oracle RAC | 5 |
| Oracle Database File System (DBFS) | 6 |
| Oracle ASM Cluster File System (ACFS) | 9 |
| Step 2: Install Oracle GoldenGate | 11 |
| Step 3: Configure Oracle GoldenGate and the Database | 11 |
| Step 4: Create an Oracle GoldenGate Deployment | 12 |
| Step 5: Create Oracle Net Alias for Oracle GoldenGate Database Connections | 14 |
| Step 6: Oracle Clusterware Configuration | 15 |
| Step 7: Configure Oracle GoldenGate Processes to Use the Shared File System | 21 |
| Extract Configuration | 21 |
| Replicat Configuration | 22 |
| Distribution Path Configuration | 24 |
| Step 8: Configure Autostart of Extract and Replicat Processes | 26 |
| Create Auto Start Tasks for Oracle GoldenGate Processes | 26 |



| | |
|--|----|
| Create Auto Restart Tasks for Oracle GoldenGate Processes | 27 |
| Summary of Recommendations when Deploying on Oracle RAC | 28 |
| References | 29 |
| Appendix A: Troubleshooting Oracle GoldenGate on Oracle RAC | 30 |
| 1. XAG log file | 30 |
| 2. XAG GoldenGate instance trace file | 30 |
| 3. CRS trace file | 31 |
| 4. GoldenGate deployment log files | 32 |
| 5. GoldenGate report files | 32 |
| Example Configuration Problems | 33 |
| 1. Incorrect parameter settings in the <code>mount-dbfs.conf</code> file | 33 |
| 2. Problems with file locking on DBFS | 35 |



Introduction

Oracle Real Application Clusters (RAC) and Oracle Clusterware allow the 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 an Oracle RAC clustered node or instance fails, the Oracle Database service continues running on the surviving nodes and instances. 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 RAC nodes and instances in the RAC cluster.

This white paper describes best practices for configuring Oracle GoldenGate Microservices Architecture, introduced with Oracle GoldenGate 12.3, 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 data from an OLTP application running on Oracle RAC to support further downstream consumption, such as middleware integration



This paper focuses on configuring Oracle GoldenGate Microservices Architecture 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 Microservices Architecture, 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 white paper.

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.

Introduced in Oracle GoldenGate 12.3, Oracle GoldenGate Microservices Architecture is a new administration architecture that provides REST-enabled services as part of the Oracle GoldenGate environment. The REST-enabled services provide remote configuration, administration, and monitoring through HTML5 web pages, command line interfaces, and APIs. Figure 1 shows the Oracle GoldenGate MicroServices Architecture referenced throughout this white paper.

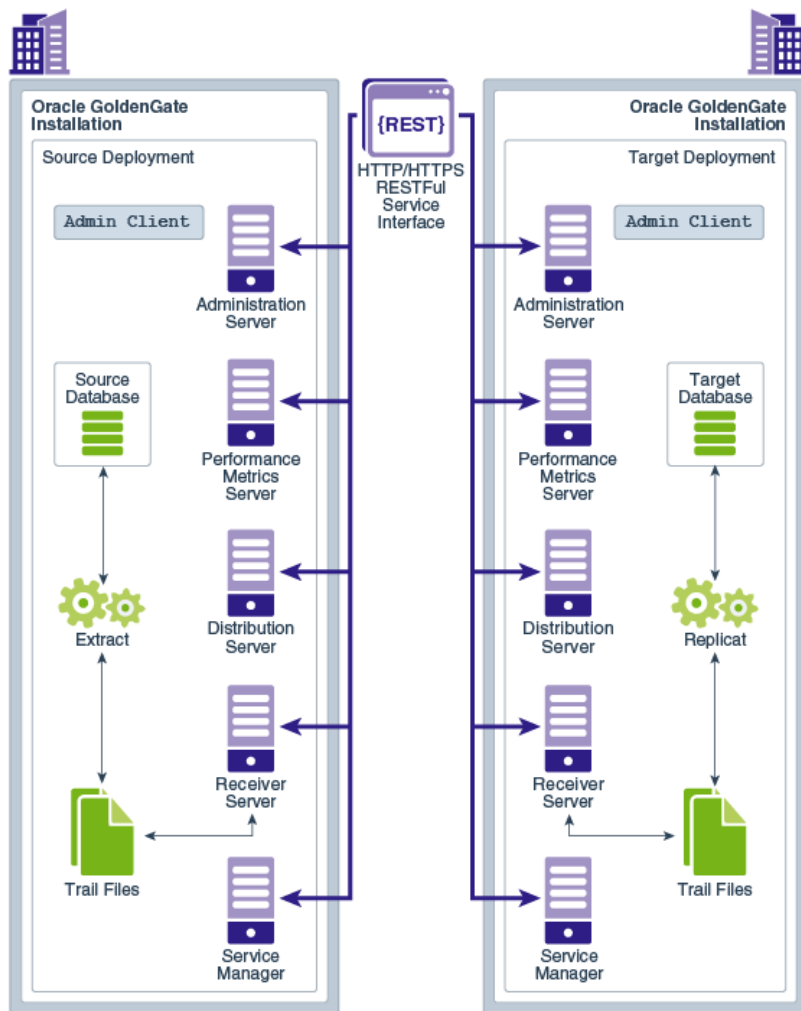


Figure 1: Oracle GoldenGate Microservices Architecture

More information about Oracle GoldenGate Microservices Architecture can be found in the Oracle GoldenGate documentation at:

<https://docs.oracle.com/goldengate/c1230/gg-winux/GGCON/toc.htm>

You must use Oracle GoldenGate version 12.3.0.1.4 or later to use the Oracle GoldenGate Microservices Architecture. The latest version of Oracle GoldenGate can be downloaded from:

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

This paper does not include information about configuration for downstream capture mode Extract.

For information about Oracle GoldenGate Classic Architecture, see the “Oracle GoldenGate with Oracle Real Application Clusters Configuration” white paper:

<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 the 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.


When using Oracle GoldenGate Microservices Architecture you **MUST** use XAG version 9 or later.

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

<http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/xag-agents-downloads-3636484.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 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 1 (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: Set Up a File System on Oracle RAC

Oracle GoldenGate Microservices Architecture is designed with a simplified installation and deployment directory structure. The installation directory should be placed on local storage on each Oracle RAC node to provide minimized downtime during software patching. The deployment directory, which is created when creating a deployment using the Oracle GoldenGate Configuration Assistant (oggca.sh), must be placed on a shared file system. The deployment directory contains configuration, security, log, parameter, trail, and checkpoint files. Placing the deployment in DBFS or ACFS provides 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 GoldenGate processes can continue running from their last known position.

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 below to configure the chosen file system.

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.

Create a file system for storing the Oracle GoldenGate deployment files. It is recommended that you allocate enough trail file disk space to permit storage of up to 12 hours of trail files. Doing this will give 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 option `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

```

The deployment `temp` directory is used by Extract to store uncommitted transaction data when its cache (managed by `CACHEMGR`) has filled up. By placing `temp` on DBFS, you get the additional benefit of larger storage potential. The `temp` directory is only used by the Oracle GoldenGate Extract processes to spill out data to disk once the cache memory area has filled up, so skip the following recommendation if no Extract process is running on this cluster. The contents of `temp` is transient in nature, and therefore not needed by Extract for recovery on startup. If `temp` is used often, and increasing the size of the cache doesn't help, it is recommended that you place `temp` 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, as shown in this 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 temporary file storage size.

Create the file system, making sure you are connected as the same database 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 in this file system should be configured with the storage options `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 modifications.

1. 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 the accidental mounting of DBFS from concurrent nodes creating the potential of concurrent file writes, causing file corruption problems.

Example command to register the DBFS resource:

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

2. Modify the `MOUNT_OPTIONS` parameter in the `mount-dbfs.conf` file to the following:

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

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.

If you are using a `dbfs_client` from Oracle Database 12c Release 2 (12.2), due to a change in the file locking implementation a fix is required for bug 27056711. Once the fix has been applied, the `MOUNT_OPTIONS` should be modified to include the `nolock` option:

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

The `nolock` mount option is also required if you are using Oracle Database 18c.

If you are using Oracle Database 11g Release 11.2.0.4 or Oracle Database 12c Release 1, run the following command to check if a database patch has been applied to fix bug 22646150, which changes the way in which file locking is handled by DBFS:

```
$ opatch lsinventory | grep 22646150
```

If this bug has been patched, refer to [Appendix A, 'Problems with file locking on DBFS'](#) section for further information.

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 the directory for storing the Oracle GoldenGate files.

```
% cd /mnt/dbfs/goldengate
% mkdir deployments
```

NOTE: Leave the shared file system mounted. It is required for creating the Oracle GoldenGate deployment in a later step.

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 deployment files.

It is recommended that you allocate enough trail file disk space to permit storage of up to 12 hours of trail files. Doing this will give 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 `ASMCMDB` as the Oracle ASM administrator user.

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

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

```
ASMCMDB> 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:
```

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.

```
# 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.

```
# srvctl add filesystem -device /dev/asm/acfs_gg-151 -volume ACFS_GG
-diskgroup DATA1 -path /mnt/acfs_gg -user oracle -autostart NEVER
```

NOTE: It is important to include the `-autostart NEVER` parameter to prevent ACFS from being automatically mounted when the RAC node reboots. The Grid Infrastructure Agent (XAG) controls when the file system is mounted and unmounted.

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: data1
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 is 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.datacl.acfs_gg.acfs
TYPE=ora.acfs.type
TARGET=ONLINE          , ONLINE
STATE=ONLINE on oggadm07, ONLINE on oggadm08
```

3. Create GoldenGate ACFS directory.

After the file system is mounted, create the directory for storing the Oracle GoldenGate deployments.

```
% cd /mnt/acfs_gg
% mkdir deployments
```

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

<https://docs.oracle.com/database/122/OSTMG/overview-acfs-advm.htm#OSTMG31000>

NOTE: Leave the shared file system mounted. It is required for creating the Oracle GoldenGate deployment in a later step.

Step 2: 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 GoldenGate configuration. Make sure the installation directory is the same on all nodes.

NOTE: Even though the Oracle GoldenGate installer may advise 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 generic installation instructions detailed in *Installing Oracle GoldenGate*:

<https://docs.oracle.com/goldengate/c1230/gg-winux/OGGIN/toc.htm>

Step 3: Configure the Oracle GoldenGate Database

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

Database configuration summary:

- » Run the Oracle GoldenGate source database in `ARCHIVELOG` 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 redo log reading performance parameters for Extract.
- » Configure the streams pool in the System Global Area (SGA).
- » Install the Streams Performance Adviser (UTL_SPADV) PL/SQL package to monitor integrated Extract and integrated Replicat performance.

Step 4: Create an Oracle GoldenGate Deployment

Once the Oracle GoldenGate software has been installed, the next step is to create a deployment using the Oracle GoldenGate Configuration Assistant (oggca).

Recommendations for creating the Oracle GoldenGate deployment are listed below.

1. Specify the following for the creation of a new Service Manager.
 - a. Service Manager Deployment Home location on the shared DBFS or ACFS file system.
 - b. Specify `localhost` as the Listening hostname/address field. Using `localhost` allows the deployment to be started on all of the RAC nodes without the need for a Virtual IP address (VIP).
 - c. Select to 'Integrate with XAG'.

Below is a screenshot of the recommended configuration.

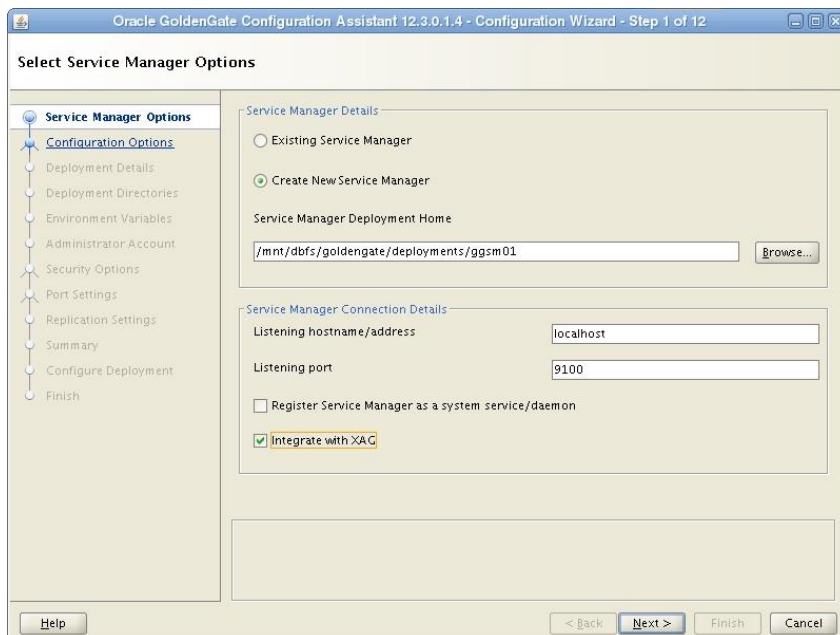


Figure 2: Oracle GoldenGate Microservices Service Manager configuration

2. Specify the Deployment home directory on the shared DBFS or ACFS file system:

Oracle GoldenGate Configuration Assistant 12.3.0.1.4 - Configuration Wizard - Step 4 of 12

Specify Deployment Directories

Service Manager Options
Configuration Options
Deployment Details
Deployment Directories
Environment Variables
Administrator Account
Security Options
Port Settings
Replication Settings
Summary
Configure Deployment
Finish

Deployment home:
/mnt/dbfs/goldengate/deployments/gg01 Browse...

☐ Customize directories

Configuration files:
/mnt/dbfs/goldengate/deployments/gg01/etc Browse...

Configuration bootstrap files, process parameter files:
/mnt/dbfs/goldengate/deployments/gg01/etc/conf Browse...

Security configuration files:
/mnt/dbfs/goldengate/deployments/gg01/etc/ssl Browse...

Runtime and support files:
/mnt/dbfs/goldengate/deployments/gg01/var Browse...

Trail files:
/mnt/dbfs/goldengate/deployments/gg01/var/lib/data Browse...

Help < Back Next > Finish Cancel

Figure 3: Deployment home directory configuration

3. If the Management Pack for Oracle GoldenGate has been licensed, enable monitoring to use the performance metric server using either Berkeley Database (BDB) or Lightning Memory Database (LMDb).

Oracle GoldenGate Configuration Assistant 12.3.0.1.4 - Configuration Wizard - Step 8 of 12

Specify Port Settings

Service Manager Options
Configuration Options
Deployment Details
Deployment Directories
Environment Variables
Administrator Account
Security Options
Port Settings
Replication Settings
Summary
Configure Deployment
Finish

Service Manager listening hostname/address: 127.0.0.1

Service Manager port: 9100

Administration Server port: 9101

Distribution Server port: 9102

Receiver Server port: 9103

☒ Enable Monitoring

Metrics Server port: 9104

Metrics Server UDP port (data): 9105

Metrics Server DataStore type: BDB (selected)
BDB
LMDb

Help < Back Next > Finish Cancel

Figure 4: Selecting the Metric Server Datastore type

Continue through the Oracle GoldenGate Configuration Assistant until the deployment is created.

If LMDB was selected for the Metrics Server DataStore type and DBFS is the chosen shared storage type, the physical datastore location on disk must be moved to local storage after the deployment creation. This is due to a limitation of accessing memory mapped files on DBFS that is mounted with the `direct_io` mount option.

To modify the datastore location use the following instructions.

1. Set environment variable OGG_VAR_HOME to the location of the deployment var directory.

```
$ export OGG_VAR_HOME=/mnt/dbfs/goldengate/deployments/gg01/var
```

2. Create a directory on local storage for the datastore.

```
$ mkdir -p /u01/goldengate/datastore
```

3. Modify the datastore location.

```
$ pmsrvr --prop=/config/RepoDatastorePath --value=/u01/goldengate/datastore  
--exit
```

NOTE: The status of the Oracle GoldenGate deployment servers remains persistent across deployment stop and starts. For example, if the Administration Server is shut down when the deployment is stopped (will also shut down the Extract and Replicat processes), it will remain shut down next time the deployment is started.

Step 5: Create Oracle Net Alias for Oracle GoldenGate Database Connections

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

For example:

```
ggdb.local=
(DESCRIPTION =
  (SDU=2097152)
  (ADDRESS =
    (PROTOCOL=BEQ)
    (PROGRAM=/u01/app/oracle/product/12.2.0/dbhome_1/bin/oracle)
    (ARGV0=oracleggdb1)
    (ARGS='(DESCRIPTION= (SDU=2097152) (LOCAL=YES) (ADDRESS= (PROTOCOL=BEQ)) )')
  )
  (ENVS='ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1,ORACLE_SID=ggdb1')
)
(CONNECT_DATA= (SID=ggdb1))
)
```

The `ORACLE_SID` parameter will match the SID running on the Oracle RAC node where the `tnsnames.ora` file is located. This way, the same TNS alias can be used by the Oracle GoldenGate processes, no matter which node they are running on.

For further information on setting the Oracle Net SDU parameter refer to the *Oracle Net Services Administrator's Guide*:

<https://docs.oracle.com/en/database/oracle/oracle-database/12.2/netag/net-services-administrators-guide.pdf>

With the Oracle GoldenGate deployment already created, use the Administration Server home page to create a database credential using the above TNS alias. Because the deployment is stored on shared storage, and the TNS alias is created on all of the required RAC nodes used by Oracle GoldenGate, this database credential only needs to be created on one RAC node where the shared file system is currently mounted.

See figure 5 below for an example of the database credential creation using the TNS alias.

The screenshot shows the Oracle GoldenGate Administration Server 12.3.0.1.4 for Oracle 12c (gg01) interface. The 'Database' tab is selected, and the 'Credentials' section is active. A table with columns 'Domain', 'Alias', 'User ID', and 'Action' is shown, but it is empty with the message 'No data to display.' Below the table, there is a section for creating a new credential. The 'Credential Domain' is set to 'GoldenGate'. The 'Credential Alias' is set to 'gg_database'. The 'User ID' field is highlighted with a red box and contains the value 'ggadmin@ggdb.local'. The 'Password' and 'Verify Password' fields are masked with dots. 'Cancel' and 'Submit' buttons are at the bottom.

Figure 5: Creating database credential using a TNS alias

Step 6: Oracle Clusterware Configuration

The following procedure shows how to 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 deployment when relocating between Oracle RAC nodes.

1. Install the Oracle Grid Infrastructure Standalone Agent.

Download the latest XAG software from <http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/xag-agents-downloads-3636484.html>.

When using Oracle GoldenGate Microservices Architecture, it is a requirement to use XAG version 9 or later.

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 must be installed in the same directory on all RAC nodes in the cluster where 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 so that the location of `agctl` is known when the user logs on to the machine.

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

NOTE: It is important to make sure that the XAG bin directory is specified BEFORE the Grid Infrastructure bin directory to ensure the correct `agctl` binary is found. This should be set in the oracle user environment to take effect at time of logging on, such as in the `.bashrc` file when the Bash shell is in use.

3. Prepare to create the Application Virtual IP Address (VIP).

A dedicated application VIP is required on a GoldenGate target host to ensure the Oracle GoldenGate Distribution Server can communicate with the Distribution Receiver running on 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 is migrated to another node in the event of a node failure. This allows Oracle GoldenGate Distribution Server 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 is also a target for a distribution path.

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

- a. The network number, which can be identified using 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 for the new Application VIP, provided by your system administrator. 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 Oracle Grid Infrastructure Agent.

4. Create an application service.

An application service is required so that the Oracle Grid Infrastructure Agent will automatically start the Oracle GoldenGate deployment when the database is opened.

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

```
$ srvctl add service -db <db_name> -service oggserv -preferred <instance_1>
-avaliable <instance_2, instance_3 etc.>
```

Example:

```
$ srvctl add service -db ggdb -service oggserv -role PRIMARY -preferred ggdb1
-avaliable ggdb2
```

5. Configure Oracle Grid Infrastructure Agent (XAG).

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

To register Oracle GoldenGate Microservices Architecture with XAG use the following command format.

```
agctl add goldengate <instance_name>
  --gg_home <GoldenGate_Home>
  --service_manager
  --config_home <GoldenGate_SvcMgr_Config>
  --var_home <GoldenGate_SvcMgr_VarDir>
  --port <port number>
  --adminuser <OGG admin user>
  --user <GG instance user>
  --group <GG instance group>
  --network <network_number>
  --ip <ip_address>
  --filesystems <CRS_resource_name>
  --db_services <service_name>
  --use_local_services
  --nodes <node1, node2, ... ,nodeN>
```

Where:

- gg_home specifies the location of the Oracle GoldenGate software.
- service_manager indicates this is an Oracle GoldenGate Microservices instance.
- config_home specifies the Oracle GoldenGate deployment configuration home directory.
- var_home specifies the Oracle GoldenGate deployment variable home directory.
- port specifies the deployment Service Manager port number.
- adminuser specifies the Oracle GoldenGate Microservices administrator account name.
- user specifies the name of the operating system user that owns the Oracle GoldenGate deployment.
- group specifies the name of the operating system group that owns the Oracle GoldenGate deployment.
- network specifies the network subnet for the VIP, determined on page 17 (optional).
- ip specifies the IP address for the VIP, which was determined on page 17 (optional). If you have already created a VIP, then specify it using the --vip_name <vip_name> parameter in place of --network and --ip.
- filesystems specifies the CRS file system resource that must be mounted before the deployment is started.

--db_services specifies the ora.<database>.<service_name>.svc service name that was created in the previous step.

--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.

--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 (optional).

Notes:

1. The GoldenGate instance registration with XAG **MUST** be run as the root user, even if no VIP is created.
2. The user and group parameters are mandatory because the GoldenGate registration with XAG is run as the root user.

Below are some examples of registering Oracle GoldenGate with XAG.

- a. Source RAC cluster, no need for an application VIP using DBFS.

```
# agctl add goldengate GG_SOURCE \  
--gg_home /u01/oracle/goldengate/gg123_MA \  
--service_manager \  
--config_home /mnt/dbfs/goldengate/deployments/ggsm01/etc/conf \  
--var_home /mnt/dbfs/goldengate/deployments/ggsm01/var \  
--port 9100 \  
--adminuser admin \  
--user oracle \  
--group oinstall \  
--filesystems dbfs_mount \  
--db_services ora.ggdb.oggserv.svc \  
--use_local_services
```

Where:

- the GoldenGate instance is GG_SOURCE
- the GoldenGate home directory is /u01/oracle/goldengate/gg123_MA
- this is an Oracle GoldenGate Microservices Architecture instance (--service_manager)
- the Oracle GoldenGate deployment configuration home directory is /mnt/dbfs/goldengate/deployments/ggsm01/etc/conf
- the Oracle GoldenGate deployment variable home directory is /mnt/dbfs/goldengate/deployments/ggsm01/var
- the deployment Service Manager port number is 9100
- the Oracle GoldenGate Microservices administrator account name is admin
- the Oracle GoldenGate user is oracle in the group oinstall
- the CRS resource name for the file system the deployment depends on is dbfs_mount
- this GoldenGate instance will be started on the same RAC node as the CRS service called ora.ggdb.oraserv.svc will be co-located on the same RAC node as this GoldenGate instance.

- b. Target RAC cluster, using ACFS, with an application VIP running on a subset of the nodes in the cluster.

```
# agctl add goldengate GG_TARGET \  
--gg_home /u01/oracle/goldengate/gg123_MA \  
--service_manager \  
--config_home /mnt/acfs/goldengate/deployments/ggsm02/etc/conf \  
--var_home /mnt/acfs/goldengate/deployments/ggsm02/var \  
--port 9100 \  
--adminuser admin \  
--user oracle \  
--group oinstall \  
--network 1 --ip 10.13.11.203 \  
--filesystems ora.datacl.acfs_gg.acfs \  
--db_services ora.ggdb.oggserv.svc \  
--use_local_services
```

Where:

- the GoldenGate instance is GG_TARGET
- the GoldenGate home directory is /u01/oracle/goldengate/gg123_MA
- this is an Oracle GoldenGate Microservices Architecture instance (--service_manager)
- the Oracle GoldenGate deployment configuration home directory is /mnt/acfs/goldengate/deployments/ggsm02/etc/conf
- the Oracle GoldenGate deployment variable home directory is /mnt/acfs/goldengate/deployments/ggsm02/var
- the deployment Service Manager port number is 9100
- the Oracle GoldenGate Microservices administrator account name is admin
- the Oracle GoldenGate user is oracle in the group oinstall
- the network is the default ora.net1.network and the VIP is 10.13.11.203
- the CRS resource name for the file system the deployment depends on is ora.datacl.acfs_gg.acfs
- this GoldenGate instance will be started on the same RAC node as the CRS service called ora.ggdb.oraserv.svc will be co-located on the same RAC node as this GoldenGate instance.

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

To check the status of Oracle GoldenGate:

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

To start the Oracle GoldenGate deployment, and all Extract/Replicat processes that have autostart enabled:

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



To stop the Oracle GoldenGate deployment:

```
% agctl stop goldengate GG_Target
```

To manually relocate the Oracle GoldenGate deployment 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

Instance name: GG_TARGET
Application GoldenGate location is: /u01/oracle/goldengate/gg123_MA
Goldengate MicroServices Architecture environment : yes
Goldengate Service Manager configuration directory :
/mnt/acfs/goldengate/deployments/ggsm01/etc/conf
Goldengate Service Manager var directory :
/mnt/acfs/goldengate/deployments/ggsm01/var
Service Manager Port : 9100
Goldengate Administration User : admin
Autostart on DataGuard role transition to PRIMARY: no
Database Services needed: ora.ggdb.oggserv.svc [use_local_services]
File System resources needed: ora.dataac1.acfs_gg.acfs
Network: 1, IP: 10.13.11.203, User:oracle, Group:oinstall
```

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:

<http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/xag-agents-downloads-3636484.html>

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

When creating Extract, Distribution Paths, and Replicat processes with Oracle GoldenGate Microservices Architecture, all files that need to be shared between Oracle RAC nodes are already shared with the deployment being stored on a shared file system.

Extract Configuration

1. When creating an Extract using the Oracle GoldenGate Administration Server GUI interface, leave the **Trail SubDirectory** parameter blank. Leaving it blank will ensure that the trail files are created in the deployment directories stored on the shared file system.

The screenshot displays the Oracle GoldenGate Administration Server GUI for version 12.3.0.1.4 for Oracle 12c. The breadcrumb navigation shows 'Overview > Add Extract'. The left sidebar contains a user profile 'admin' with 'Security' status, and a menu with 'Overview', 'Configuration', 'Diagnosis', and 'Administrator'. The main content area is titled 'Add Extract' and features a progress bar with three steps: 'Extract Type' (selected), 'Extract Options', and 'Parameter File'. Under the 'Basic Information' section, the following fields are visible: 'Process Name' (EXT_1), 'Description' (empty), 'Intent' (Unidirectional), 'Credential Domain' (GoldenGate), 'Credential Alias' (gg_database), 'Begin' (Now), 'Trail Name' (aa), 'Trail SubDirectory' (empty and highlighted with a red box), 'Trail Size (MB)' (500), 'Trail Sequence' (0), 'Trail Offset' (0), and 'Remote' (toggle switch).

Figure 6: Creating an Oracle GoldenGate Extract process with the Administration Server.

2. 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 temporary files directory fills up.

For DBFS:

```
CACHEMGR CACHEDIRECTORY /mnt/dbfs/gg_dirtmp
```

For ACFS:

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

Here is an example of the parameters specified for an integrated Extract with the Oracle GoldenGate Administration Server GUI.

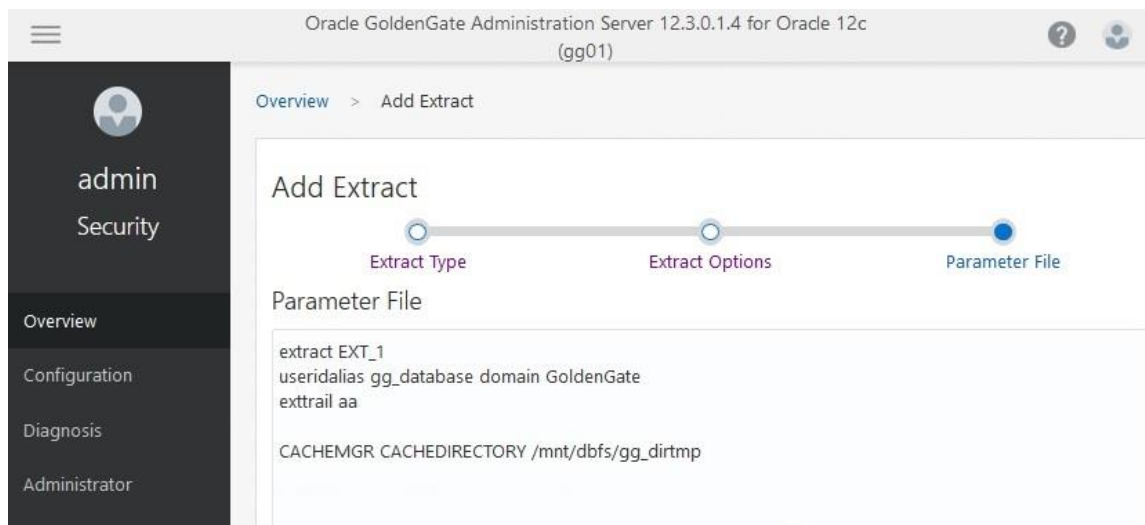


Figure 7: Extract parameters for defining the temporary cache and bounded recovery files.

More instructions about creating an Extract process are available in the *Using Oracle GoldenGate Microservices Architecture* guide at <https://docs.oracle.com/goldengate/c1230/gg-winux/GGSAU/toc.htm>.

Replicat Configuration

1. If creating one or more non-integrated Replicat processes, make sure that a checkpoint table has been created in the GGADMIN schema. The checkpoint table can be created using the Oracle GoldenGate Administration Server GUI, clicking on the '+' button and entering the checkpoint table name in the format of `schema.tablename`.

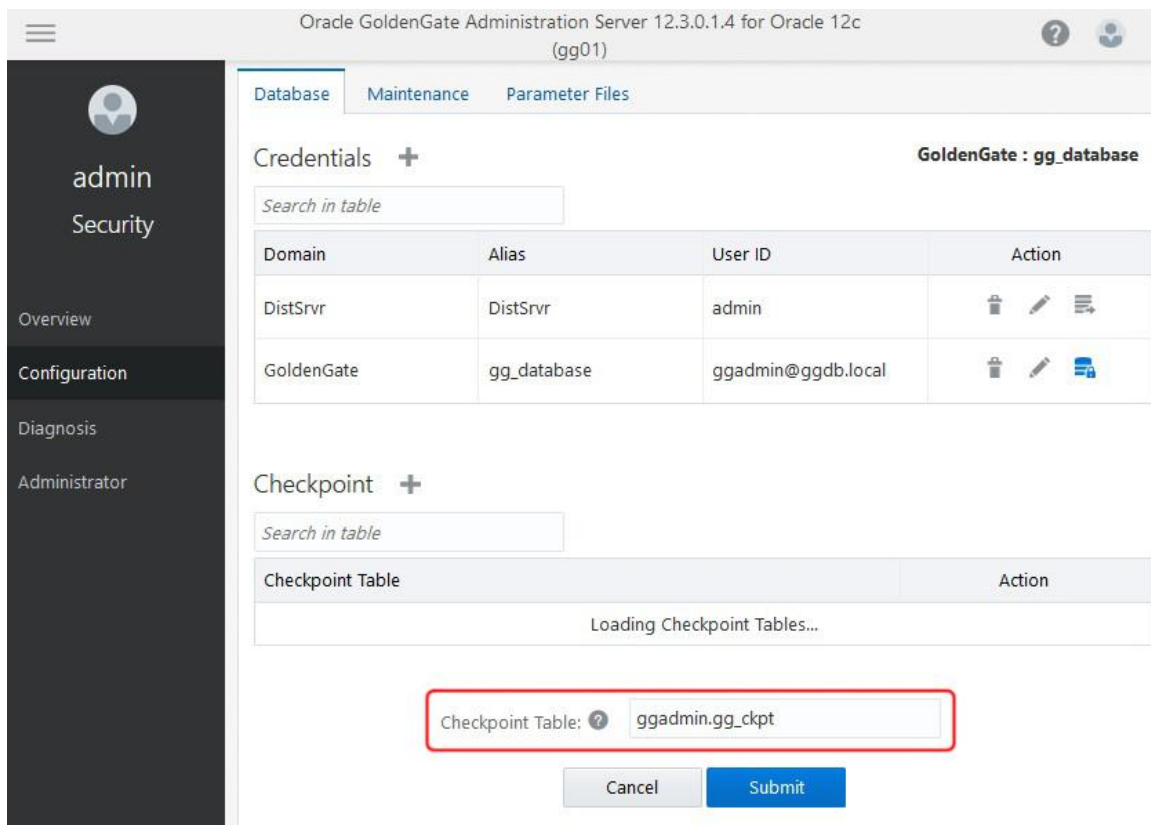


Figure 8: Creating the checkpoint table for non-integrated Replicat processes

- When creating a Replicat using the Oracle GoldenGate Administration Server GUI interface, leave the **Trail SubDirectory** parameter blank. Leaving it blank will ensure the trail files are created in the deployment directories stored on the shared file system.
- If a checkpoint table was created previously, select the table name from the **Checkpoint Table** pulldown list.

Below in figure 9, is a screenshot that shows the **Trail SubDirectory** and **Checkpoint Table** prompts.

Oracle GoldenGate Administration Server 12.3.0.1.4 for Oracle 12c (gg01)

Overview > Add Replicat

Add Replicat

Replicat Type Replicat Options Parameter File

* Process Name: REP_1

Description:

Intent: Unidirectional

Create new credential

Credential Domain: GoldenGate

Credential Alias: gg_database

* Begin: Position in Log

* Transaction Log Sequence Number: 0

* Transaction Log RBA Offset: 0

* Source: Trail

* Trail Name: aa

Trail SubDirectory:

Checkpoint Table: GGADMIN.GG_CKPT

Figure 9: Replicat creation with Trail SubDirectory and Checkpoint Table

- After the Replicat has been created, add the `CHECKPOINTTABLE` parameter to the `GLOBALS` file, located in `/<deployment_directory>/etc/conf/ogg`. Make sure the same checkpoint table name, with schema name, is specified.

For example:

```
CHECKPOINTTABLE GGADMIN.GG_CKPT
```

Distribution Path Configuration

With Oracle GoldenGate Microservices Architecture, Data Pump processes have been replaced with Distribution Paths that provide a way to distribute one or more trails to one or more destinations using lightweight filtering (no transformations). Paths are created using the Oracle GoldenGate Distribution Server GUI.

Below is an example of creating a distribution path, with configuration recommendations highlighted.

Oracle GoldenGate Distribution Server 12.3.0.1.4 (gg01)

Overview > Add Path

Add Path

* Path Name:

Description:

* Source:

1. Generated Source URI:

Reverse proxy enabled? ☐

2. * Target:

Use Basic Authentication: ☐

Generated Target URI:

Sequence Length:

Trail Size (MB):

Configure Trail Format: ☐

Begin:

Source Sequence Number:

Source RBA Offset:

3. Critical: ☐

Auto Restart: ☒

Auto Restart Options

Retries:

Delay:

Figure 10: Creating a distribution path using the Distribution Server GUI

1. Change the **Generated Source URI** specifying `localhost` for the server name. This allows the distribution path to be started on any of the Oracle RAC nodes.
2. Set the **Target** host name to the host name that will be used for connecting to the target system along with the port number of the Receiver Server running on the target host. If the Oracle GoldenGate target host is an Oracle RAC cluster, specify the Application Virtual IP host name created with Cluster Ready Services (CRS), so that a single IP address can be moved between Oracle RAC nodes. Instructions for creating a VIP for use by Oracle GoldenGate is provided above in step 6.

In the event of a GoldenGate target RAC node failover, the distribution path will continue to transfer trail files to the Receiver Server when it moves between RAC nodes.

3. Set the distribution path to automatically restart when the Distribution Server starts. This is required so that manual intervention is not required after a RAC node relocation of the Distribution Server. It is recommended to set the number of **Retries** to 10. Set the **Delay**, which is the amount of time in minutes to pause between restart attempts, to 1.

More instructions about creating distribution paths are available in *Using Oracle GoldenGate Microservices Architecture* at <https://docs.oracle.com/goldengate/c1230/gg-winux/GGSAU/toc.htm>.

Step 8: Configure Autostart of Extract and Replicat Processes

Configure the Extract and Replicat processes to automatically start when the Oracle GoldenGate Administration Server is started, and then to restart if any Extract or Replicat processes abend.

Create Auto Start Tasks for Oracle GoldenGate Processes

Using the Oracle GoldenGate Administration Server GUI, create new auto start tasks for each of the Oracle GoldenGate processes. The following are the recommended settings:

- **Critical:** leave the default setting of disabled. When **Critical** is enabled for the auto start task for an Oracle GoldenGate process, if it fails to start, the entire deployment will fail over to another Oracle RAC node. This is generally not recommended, because the reason for the process failing to start may also cause it to fail when starting on another node. The Oracle RAC node relocation of the deployment will cause an outage for all other Oracle GoldenGate processes.
- **Process Type:** leave the default type of ER, indicating either an Extract or Replicat process task.

Figure 11 shows an example of creating a new auto start task for an Oracle GoldenGate Extract process.

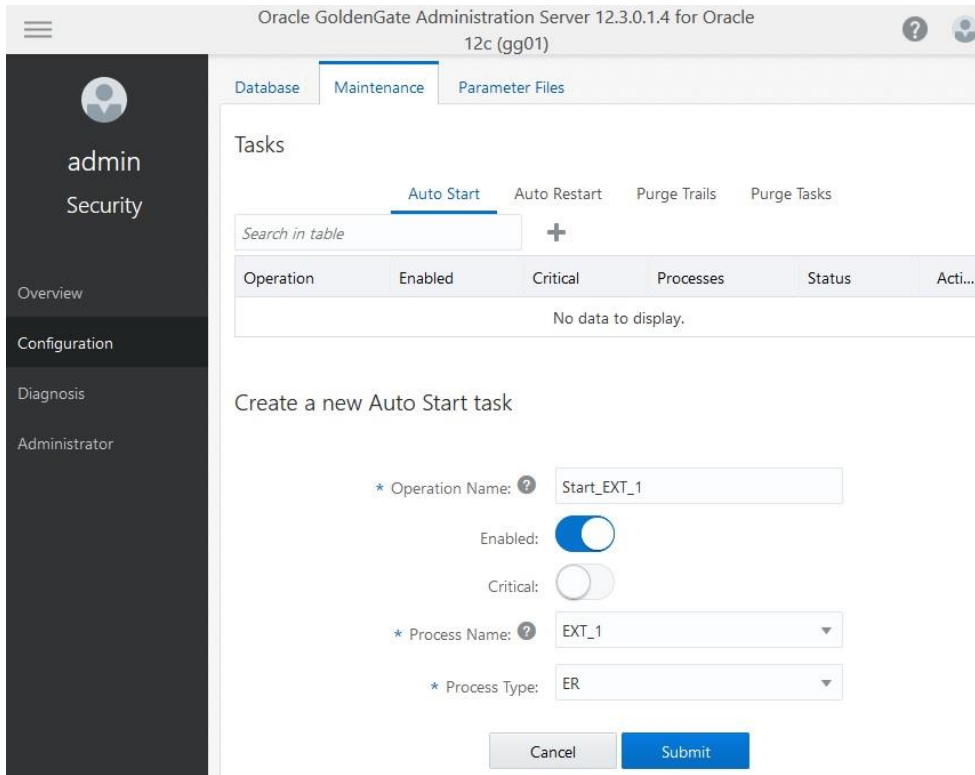


Figure 11: Creating a GoldenGate process auto start task using the Oracle GoldenGate Administration Server GUI.

Create Auto Restart Tasks for Oracle GoldenGate Processes

Using the Oracle GoldenGate Administration Server GUI, create auto restart tasks for each of the Oracle GoldenGate processes. The following are the recommended settings.

- **Process Type:** leave the default type of ER, indicating either an Extract or Replicat process task.
- **Delay:** set to 1 minute, so that the Oracle GoldenGate process will try restarting with a one minute interval.
- **Max Retries:** set to 5, so that the Oracle GoldenGate process will be restarted 5 times before aborting restart efforts.
- **Window:** set to 20 minutes, so that the number of retries are reset to 0 after 20 minutes.

Figure 12 shows an example of creating a new auto restart task for an Oracle GoldenGate Extract process.

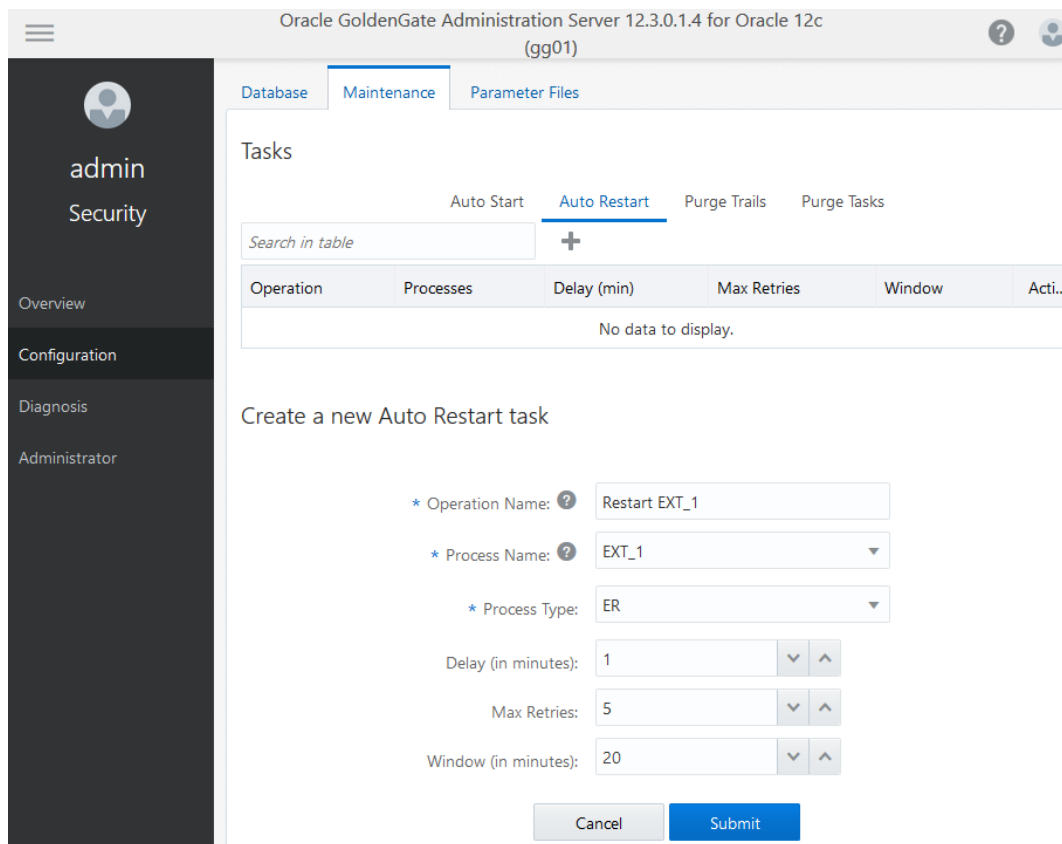


Figure 12: Creating a GoldenGate process auto restart task using the Oracle GoldenGate Administration Server GUI.

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 Oracle RAC node, making sure the software location is the same on all Oracle RAC nodes.
- » 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, and parameter files).
- » Use the same DBFS or ACFS mount point on all of the Oracle RAC nodes that may run Oracle GoldenGate.
- » When creating the Oracle GoldenGate deployment, specify either DBFS or ACFS for the deployment location.
- » Install Grid Infrastructure agent (XAG) version 9 or later on all Oracle RAC nodes that will run Oracle GoldenGate.
- » Configure the Oracle GoldenGate process to start and restart when the deployment is started.



References

- » [Installing Oracle GoldenGate 12c \(12.3.0.1\)](#)
- » [Using the Oracle GoldenGate Microservices Architecture \(12.3.0.1\)](#)
- » [Reference for Oracle GoldenGate \(12.3.0.1\)](#)
- » [Oracle Database SecureFiles and Large Object Developer's Guide \(DBFS\)](#)
- » [Oracle Automatic Storage Management Administrator's Guide \(ACFS\)](#)
- » [Oracle Clusterware Administration and Deployment Guide](#)
- » [Oracle GoldenGate Performance Best Practices MAA White Paper](#)
- » [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 an Oracle RAC node. There are number of files generated by Oracle 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:

```
2018-06-18 11:52:21: stop resource success
2018-06-18 11:52:38: agctl start goldengate GGT
2018-06-18 11:52:38: executing cmd: /u01/app/12.2.0.1/grid/bin/crsctl status
res xag.GGT.goldengate
2018-06-18 11:52:38: executing cmd: /u01/app/12.2.0.1/grid/bin/crsctl status
res xag.GGT.goldengate -f
2018-06-18 11:52:38: executing cmd: /u01/app/12.2.0.1/grid/bin/crsctl start
resource xag.GGT.goldengate -f
2018-06-18 11:52:45: Command output:
> CRS-2672: Attempting to start 'xag.GGT.goldengate' on 'dbm01db05'
> CRS-2676: Start of 'xag.GGT.goldengate' on 'dbm01db05' succeeded
>End Command output
2018-06-18 11:52:45: start resource success
```

2. XAG GoldenGate instance trace file

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

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

File name: <GoldenGate_instance_name>_agent_goldengate.trc

Contains the output from the commands executed by agctl along with the environment variables used, and any debug output enabled for the underlying commands.

Example:

```
2018-06-18 12:14:46: Exported ORACLE_SID ggdg1
2018-06-18 12:14:46: Exported GGS_HOME /u01/oracle/goldengate/gg123_MA
2018-06-18 12:14:46: Exported OGG_CONF_HOME
/mnt/dbfs/goldengate/deployments/ggsm01/etc/conf
2018-06-18 12:14:46: Exported LD_LIBRARY_PATH
/u01/oracle/goldengate/gg123_MA:/u01/app/12.2.0.1/grid/lib:/etc/ORCLcluster/lib
2018-06-18 12:14:46: Exported LD_LIBRARY_PATH_64
/u01/oracle/goldengate/gg123_MA
2018-06-18 12:14:46: Exported LIBPATH /u01/oracle/goldengate/gg123_MA
```

```

2018-06-18 12:14:46: ogg input =
{"oggHome":"/u01/oracle/goldengate/gg123_MA","serviceManager":{"oggConfHome":"/mnt/dbfs/goldengate/deployments/ggsm01/etc/conf","portNumber":9100},"username":"admin","credential":"xyz"}
2018-06-18 12:14:46: About to exec /u01/oracle/goldengate/gg123_MA/bin/XAGTaskHealthCheck
2018-06-18 12:14:47: XAGTask retcode = 0

```

3. 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:

```

2018-06-18 11:52:38.634 : AGFW:549631744: {1:30281:59063} Agent received the
message: RESOURCE_START[xag.GGT.goldengate 1 1] ID 4098:4125749
2018-06-18 11:52:38.634 : AGFW:549631744: {1:30281:59063} Preparing START
command for: xag.GGT.goldengate 1 1
2018-06-18 11:52:38.634 : AGFW:549631744: {1:30281:59063} xag.GGT.goldengate
1 1 state changed from: OFFLINE to: STARTING
2018-06-18 11:52:38.634 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] Executing action script:
/u01/oracle/XAG_MA/bin/aggoldengatescaas[start]
2018-06-18 11:52:38.786 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] GG agent running command 'start' on
xag.GGT.goldengate
2018-06-18 11:52:42.140 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] ServiceManager fork pid = 265747
2018-06-18 11:52:42.140 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] Waiting for
/mnt/dbfs/goldengate/deployments/ggsm01/var/run/ServiceManager.pid
2018-06-18 11:52:42.140 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] Waiting for SM to start
2018-06-18 11:52:42.140 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] ServiceManager PID = 265749
2018-06-18 11:52:43.643 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] XAGTask retcode = 0
2018-06-18 11:52:43.643 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [start] XAG HealthCheck after start
returned 0
2018-06-18 11:52:43.643 : AGFW:558036736: {1:30281:59063} Command: start for
resource: xag.GGT.goldengate 1 1 completed with status: SUCCESS
2018-06-18 11:52:43.643 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [check] Executing action script:
/u01/oracle/XAG_MA/bin/aggoldengatescaas[check]
2018-06-18 11:52:43.644 : AGFW:549631744: {1:30281:59063} Agent sending
reply for: RESOURCE_START[xag.GGT.goldengate 1 1] ID 4098:4125749
2018-06-18 11:52:43.795 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [check] GG agent running command 'check' on
xag.GGT.goldengate
2018-06-18 11:52:45.548 : CLSDYNAM:558036736:
[xag.GGT.goldengate]{1:30281:59063} [check] XAGTask retcode = 0

```

```
2018-06-18 11:52:45.548 : AGFW:549631744: {1:30281:59063} xag.GGT.goldengate
1 1 state changed from: STARTING to: ONLINE
```

4. GoldenGate deployment log files

Location: <Goldengate_deployment_directory>/<deployment_name>/var/log

Example location: /mnt/dbfs/goldengate/deployments/gg01/var/log

File names: adminsrvr.log, recvsrvr.log, pmsrvr.log, distsrvr.log

Contains the output of start, stop and status checks of the Oracle GoldenGate deployment processes (Administration Server, Distribution Server, Receiver Server and the Performance Metrics Server).

Example:

```
2018-06-18T11:52:42.645-0400 INFO | Setting deploymentName to 'gg01'. (main)
2018-06-18T11:52:42.665-0400 INFO | Read SharedContext from store for length 19
of file '/mnt/dbfs/goldengate/deployments/gg01/var/lib/conf/adminsrvr-
resources.dat'. (main)
2018-06-18T11:52:42.723-0400 INFO | XAG Integration enabled (main)
2018-06-18T11:52:42.723-0400 INFO | Configuring security. (main)
2018-06-18T11:52:42.723-0400 INFO | Configuring user authorization secure store
path as
'/mnt/dbfs/goldengate/deployments/gg01/var/lib/credential/secureStore/'. (main)
2018-06-18T11:52:42.731-0400 INFO | Configuring user authorization as ENABLED.
(main)
2018-06-18T11:52:42.749-0400 INFO | Set network configuration. (main)
2018-06-18T11:52:42.749-0400 INFO | Asynchronous operations are enabled with
default synchronous wait time of 30 seconds (main)
2018-06-18T11:52:42.749-0400 INFO | HttpServer configuration complete. (main)
2018-06-18T11:52:42.805-0400 INFO | SIGHUP handler installed. (main)
2018-06-18T11:52:42.813-0400 INFO | SIGINT handler installed. (main)
2018-06-18T11:52:42.815-0400 INFO | SIGTERM handler installed. (main)
2018-06-18T11:52:42.817-0400 WARN | Security is configured as 'disabled'.
(main)
2018-06-18T11:52:42.818-0400 INFO | Starting service listener... (main)
2018-06-18T11:52:42.819-0400 INFO | Mapped 'ALL' interface to address
'ANY:9101' with default IPV4/IPV6 options identified by
'nshb01adm05.us.oracle.com'. (main)
2018-06-18T11:52:42.821-0400 INFO | Captured 1 interface host names:
'dbm01db05.us.oracle.com' (main)
2018-06-18T11:52:42.824-0400 INFO | The Network ipACL specification is empty.
Accepting ANY address on ALL interfaces. (main)
2018-06-18T11:52:42.826-0400 INFO | Server started at 2018-06-18T11:52:42.827-
05:00 (2018-06-18T15:52:42.827Z GMT) (main)
```

5. GoldenGate report files

Location: <Goldengate_deployment_directory>/<deployment_name>/var/lib/report

Example location: /mnt/dbfs/goldengate/deployments/gg01/var/lib/report

The GoldenGate report files contain important information, warning 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-dbfis.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:

```
$ agctl start goldengate GGT --node nshb01gg05

CRS-2672: Attempting to start 'dbfs_mount' on ' nshb01gg05'
CRS-2674: Start of 'dbfs_mount' on ' nshb01gg05' failed
CRS-2679: Attempting to clean 'dbfs_mount' on ' nshb01gg05'
CRS-2681: Clean of 'dbfs_mount' on ' nshb01gg05' succeeded
CRS-4000: Command Start failed, or completed with errors.
```

The XAG log file (`agctl_goldengate_oracle.trc`) has the advantage that it shows timestamps that can be used when looking at other log or trace files:

```
2018-06-19 15:32:16: executing cmd: /u01/app/12.2.0.1/grid/bin/crsctl start
resource xag.GGT.goldengate -f -n nshb01gg05
2018-06-19 15:32:19: Command output:
> CRS-2672: Attempting to start 'dbfs_mount' on ' nshb01gg05'
> CRS-2674: Start of 'dbfs_mount' on ' nshb01gg05' failed
> CRS-2679: Attempting to clean 'dbfs_mount' on ' nshb01gg05'
> CRS-2681: Clean of 'dbfs_mount' on ' nshb01gg05' succeeded
> CRS-4000: Command Start failed, or completed with errors.
>End Command output
2018-06-19 15:32:19: start resource failed rc=1
```

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-dbfis.conf` file.

i. Incorrect DBNAME

```
2018-06-19 15:32:16.679 : AGFW:1190405888: {1:30281:17383} dbfs_mount 1 1
state changed from: UNKNOWN to: STARTING
2018-06-19 15:32:16.680 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383}
[start] Executing action script: /u01/oracle/scripts/mount-dbfis.sh[start]
2018-06-19 15:32:16.732 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383}
[start] mount-dbfis.sh mounting DBFS at /mnt/dbfs from database ggdg
2018-06-19 15:32:17.883 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383}
[start] ORACLE_SID is
2018-06-19 15:32:17.883 :CLSDYNAM:1192507136: [dbfs_mount]{1:30281:17383}
[start] No running ORACLE_SID available on this host, exiting
```

2018-06-19 15:32:17.883 : AGFW:1192507136: {1:30281:17383} Command: start
for resource: dbfs_mount 1 1 completed with invalid status: 2

ii. Incorrect MOUNT_POINT

2018-06-19 16:45:14.534 : AGFW:1734321920: {1:30281:17604} dbfs_mount 1 1
state changed from: UNKNOWN to: STARTING
2018-06-19 16:45:14.535 : CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604}
[start] Executing action script: /u01/oracle/scripts/mount-dbfs.sh[start]
2018-06-19 16:45:14.586 : CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604}
[start] mount-dbfs.sh mounting DBFS at /mnt/dbfs_gg from database ggdgs
2018-06-19 16:45:15.638 : CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604}
[start] ORACLE_SID is ggdg1
2018-06-19 16:45:15.738 : CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604}
[start] spawning dbfs_client command using SID ggdg1
2018-06-19 16:45:20.745 : CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604}
[start] **fuse: bad mount point `/mnt/dbfs_gg': No such file or directory**
2018-06-19 16:45:21.747 : CLSDYNAM:1736423168: [dbfs_mount]{1:30281:17604}
[start] Start -- OFFLINE
2018-06-19 16:45:21.747 : AGFW:1736423168: {1:30281:17604} Command: start
for resource: dbfs_mount 1 1 completed with status: FAIL

iii. Incorrect DBFS_USER or DBFS_PASSWD

2018-06-19 16:47:47.855 : AGFW:1384478464: {1:30281:17671} dbfs_mount 1 1
state changed from: UNKNOWN to: STARTING
2018-06-19 16:47:47.856 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start] Executing action script: /u01/oracle/scripts/mount-dbfs.sh[start]
2018-06-19 16:47:47.908 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start] mount-dbfs.sh mounting DBFS at /mnt/dbfs from database ggdgs
2018-06-19 16:47:48.959 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start] ORACLE_SID is ggdg1
2018-06-19 16:47:49.010 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start] spawning dbfs_client command using SID ggdg1
2018-06-19 16:47:55.118 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start] **Fail to connect to database server. Error: ORA-01017: invalid
username/password; logon denied**
2018-06-19 16:47:55.118 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start]
2018-06-19 16:47:56.219 : CLSDYNAM:1386579712: [dbfs_mount]{1:30281:17671}
[start] Start -- OFFLINE
2018-06-19 16:47:56.220 : AGFW:1386579712: {1:30281:17671} Command: start
for resource: dbfs_mount 1 1 completed with status: FAIL

iv. Incorrect ORACLE_HOME

2018-06-19 16:50:38.952 : AGFW:567502592: {1:30281:17739} dbfs_mount 1 1
state changed from: UNKNOWN to: STARTING
2018-06-19 16:50:38.953 : CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739}
[start] Executing action script: /u01/oracle/scripts/mount-dbfs.sh[start]
2018-06-19 16:50:39.004 : CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739}
[start] mount-dbfs.sh mounting DBFS at /mnt/dbfs from database ggdgs
2018-06-19 16:50:39.004 : CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739}
[start] /u01/oracle/scripts/mount-dbfs.sh: line 136:
/u01/app/oracle/product/12.2.0.1/bin/srvctl: No such file or directory
2018-06-19 16:50:39.004 : CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739}
[start] /u01/oracle/scripts/mount-dbfs.sh: line 139:
/u01/app/oracle/product/12.2.0.1/bin/srvctl: No such file or directory

```

2018-06-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739}
[start] ORACLE_SID is
2018-06-19 16:50:39.004 :CLSDYNAM:569603840: [dbfs_mount]{1:30281:17739}
[start] No running ORACLE_SID available on this host, exiting
2018-06-19 16:50:39.004 : AGFW:569603840: {1:30281:17739} Command: start for
resource: dbfs_mount 1 1 completed with invalid status: 2

```

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

2. Problems with file locking on DBFS

If using 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 Microservices Architecture locking problem.

When starting a deployment with XAG, one or more processes may not start due to detecting a locking conflict on one or more files. This will often occur after a RAC node failover where the deployment did not get a chance to shut down cleanly.

When one of the deployment server processes fails to start (Administration Server, Performance Metrics Server, Distribution Server, Receiver Server or Service Manager) check the log file for the particular server located in the deployment `var/log` directory.

For example, the log file `/mnt/dbfs/goldengate/deployments/gg01/var/log/pmsrvr.log` shows the following error on startup:


```

2018-07-11T12:41:57.619-0700 ERROR| SecureStore failed on open after retrying
due to extended file lock. (main)
2018-07-11T12:41:57.619-0700 ERROR| SecureStore failed to close (28771). (main)
2018-07-11T12:41:57.619-0700 INFO | Set network configuration. (main)
2018-07-11T12:41:57.619-0700 INFO | Asynchronous operations are enabled with
default synchronous wait time of 30 seconds (main)
2018-07-11T12:41:57.619-0700 INFO | HttpServer configuration complete. (main)
2018-07-11T12:42:07.674-0700 ERROR| Unable to lock process file, Error is [1454]
- OGG-01454 (main)
2018-07-11T12:42:07.675-0700 ERROR| Another Instance of PM Server is Already
Running (main)

```

An Extract process will report start up failures in the `ER-events.log` logfile located in the deployment log file directory.

For example, `/mnt/dbfs/goldengate/deployments/gg01/var/log/ER-events.log` shows the following error:



```
2018-07-11T00:14:56.845-0700 ERROR OGG-01454 Oracle GoldenGate Capture for
Oracle, EXT1.prm: Unable to lock file
"/mnt/dbfs/goldengate/deployments/gg01/var/run/EXT1.pce" (error 11, Resource
temporarily unavailable). Lock currently held by process id (PID) 237495.

2018-07-11T00:14:56.861-0700 ERROR OGG-01668 Oracle GoldenGate Capture for
Oracle, EXT1.prm: PROCESS ABENDING.
```

Next, check to make sure the process failing to start up, are not running on any of the RAC nodes.

Example:

```
$ ps -ef|grep EXT1|grep -v grep
```

Once it has been determined that the process is not running, the deployment needs to be shutdown cleanly, the file system unmounted and the correct DBFS patch applied.

Example:

```
$ agctl stop goldengate GGT
$ crsctl stop resource dbfs_mount
```

Check the DBFS mount options:

```
$ ps -ef|grep dbfs_client

oracle    204017      1  0 14:37 ?          00:00:00
/u01/app/oracle/product/12.2.0.1/dbhome_1/bin/dbfs_client dbfs@dbfs.local -o
allow_other,failover,direct_io /mnt/dbfs
```

It is clear the `nolock` mount option was not used, which leads to the locking errors.

Use the guidelines above on page 36, to determine if a DBFS patch is required. After which add the `nolock` mount option to the `mount-dbfs.conf` file on all RAC nodes that are part of the deployment.

Example:

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

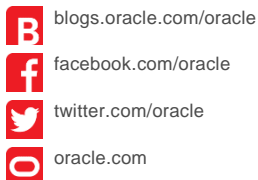
Finally, restart the deployment:

```
$ agctl start goldengate GGT
```



Oracle GoldenGate Microservices Architecture with
Oracle Real Application Clusters Configuration
August 2018
Author: Stephan Haisley

CONNECT WITH US



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

