

An Oracle White Paper  
September 2012

## **Oracle GoldenGate Best Practices: Configuring Oracle GoldenGate with Oracle Grid Infrastructure Bundled Agents (XAG)**

<b>EXECUTIVE OVERVIEW .....</b>	<b>4</b>
<b>INTRODUCTION.....</b>	<b>4</b>
<b>PREREQUISITES FOR IMPLEMENTING OGG WITH XAG:.....</b>	<b>6</b>
<b>STEPS FOR IMPLEMENTING OGG WITH XAG .....</b>	<b>7</b>
<b>Install Oracle Grid Infrastructure.....</b>	<b>7</b>
<b>Create a Shared File System for OGG Recovery Related Files .....</b>	<b>8</b>
Oracle ASM Cluster File System (ACFS) .....	8
Database File System (DBFS) .....	9
Oracle Cluster File System (OCFS2).....	10
Example of Creating an Oracle DBFS File System: .....	10
Example of Creating an Oracle ACFS File System: .....	10
<b>Register File System with Clusterware .....</b>	<b>12</b>
Example of Registering DBFS File System with Clusterware: .....	12
Example of Registering ACFS File System with Clusterware: .....	13
<b>Create an OGG administrative user account and OGG database user account .....</b>	<b>14</b>
<b>Install OGG Software .....</b>	<b>14</b>
<b>Create GoldenGate VIP and GoldenGate Agent.....</b>	<b>15</b>
Configuring the GoldenGate VIP and GoldenGate Agent Separately as the Oracle Clusterware Administrator .....	15
Configure VIP and GoldenGate Agent in one command.....	17
<b>Create the OGG Processes .....</b>	<b>18</b>
<b>Start OGG Agent .....</b>	<b>20</b>
<b>Status OGG Agent and VIPS .....</b>	<b>20</b>
<b>Status Cluster Resources .....</b>	<b>20</b>
<b>Test OGG Agent Failover.....</b>	<b>21</b>
<b>Cleanup Commands.....</b>	<b>21</b>

**CONCLUSION ..... 22**

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

## Executive Overview

This document is an introduction to Oracle GoldenGate's best practices and guidelines for configuring Oracle GoldenGate (OGG) with Oracle Grid Infrastructure Bundled Agents (XAG) for Oracle GoldenGate. This document is intended for Oracle Database Administrators (DBAs), Oracle Developers with some basic knowledge of Oracle GoldenGate software product and Oracle GoldenGate administrators. The document is intended to be an introductory supplement to the existing series of documentation available from Oracle.

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

- The reader has basic knowledge of Oracle GoldenGate products and concepts
- Referencing Oracle GoldenGate Version 11.1.1+ or 11.2.1+
- Referencing <http://oracle.com/goto/clusterware> for more details and links to specific Oracle Clusterware documentation
- Referencing Oracle Version 10.2 and above
- Referencing OS: All Oracle GoldenGate supported platforms for Oracle

## Introduction

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

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

The Oracle Clusterware Bundled Agents for OGG are now part of the Oracle Grid Infrastructure. They automate the failover and recovery of OGG processes in an Oracle RAC environment. This paper will outline the steps to configure OGG and the Oracle Clusterware Bundled Agents (XAG) for complete OGG HA. For unplanned outages the Oracle Clusterware Bundled agents for OGG will ensure that Oracle GoldenGate can tolerate server failures by moving processing to another available server in the cluster. For planned outages, the Oracle Clusterware Bundled Agents for OGG allows for online relocation without interruption of operation. As a result Oracle GoldenGate processing is dependent only on database high availability rather than server availability thereby ensuring minimal latency. The combination of the highly available GoldenGate processes managed by the Oracle

Clusterware Bundled Agents for OGG and the highly available Real Application Clusters (RAC) database is the complete HA and resource management solution.

## Prerequisites for Implementing OGG with XAG:

- The following combination of Oracle Grid Infrastructure/GoldenGate/Database releases is supported.

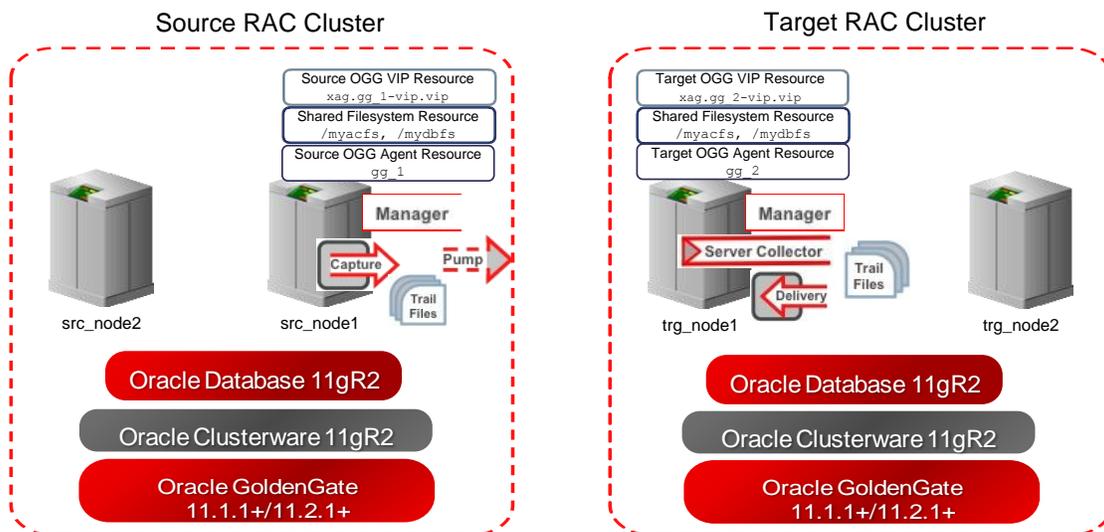
	Grid Infrastructure (GI)	GoldenGate	Database
	11.2.0.3.+	11.2.1.0.0/11.1.1.+	10.2.0.5.0, 11.1.+ , 11.2.+

Oracle Grid Infrastructure (GI) is a suite of products which include Oracle Clusterware, Automatic Storage Management (ASM), ASM Cluster File System (ACFS) and Oracle Database Quality of Service Management (QoS). The GI 11.2.0.3.+ release implies the same Oracle Automatic Storage Manager (ASM) release.

- A shared filesystem for OGG recovery information such as checkpoint files, bounded recovery files and trail files.
- An operational installation of the Oracle Grid Infrastructure version 11.2.0.3 or later on all nodes where OGG is targeted to run
- An OGG administrator account in the common UNIX primary group shared by the Oracle Grid Infrastructure owner

## Steps for Implementing OGG with XAG

These steps are required for all environments (Source and/or Target) that will be implementing OGG for High Availability with XAG. The various steps will have specific commands as examples. The examples will reference the resources and processes as outlined in the diagram below. This diagram depicts two Oracle RAC clusters representing Oracle GoldenGate replication where both the source and target databases are highly available as well as the associated Oracle GoldenGate processes. The XAG environment (XAG directory, and execution paths) must be installed on both nodes in the Source and Target clusters. The environments must be identical on both nodes in the cluster so that Oracle GoldenGate and Oracle Clusterware execution, log and configuration files are available on all nodes.



### Install Oracle Grid Infrastructure

The Oracle Grid Infrastructure bundled agents require an operational installation of the Oracle Grid Infrastructure version 11.2.0.3 or later on all nodes where the application is targeted to run. The bundled agents for 11.2.0.3 are available for download on the <http://oracle.com/goto/clusterware> downloads tab. Oracle Grid Infrastructure releases after 11.2.0.3 will include bundled agents as part of the standard distribution and installed by the Oracle Universal Installer in the Grid Infrastructure home directory.

The bundled application agents and the AGCTL management interface are packaged for release 11.2.0.3 in the zip file xagpack.zip. The \$XAG\_HOME is created locally when the xagpack.zip file is unzipped in a local directory. The \$XAG\_HOME and sub-directories must be owned by Oracle Grid Infrastructure install owner. The setup script, xagsetup.sh, must be run as the Oracle Grid Infrastructure install owner. The xagsetup.sh script offers the option to install local only (the default) or to install on all nodes in the cluster or, a subset of nodes in the cluster where the application is targeted to run. When deploying the bundled agents to multiple nodes in the cluster, the xagsetup.sh is executed on only one node of the cluster, and the xagsetup.sh script will attempt to create \$XAG\_HOME directory on all remote nodes, with an identical absolute path and the correct ownership and permissions. Please review the readme.txt file in the \$XAG\_HOME directory for details. Output for xagsetup.sh execution is logged in the directory \$XAG\_HOME/log/xagsetup.log.

For Example:

```
# unzip xagpack.zip
# cd xag

**Make sure you pre-create the installation directory in the participating
nodes beforehand otherwise the install will fail.

# xagsetup.sh -install --nodes <node1,node2,node3...>
OR
# xagsetup.sh --install -all_nodes
```

## Create a Shared File System for OGG Recovery Related Files

A shared files system is required for OGG recovery information such as checkpoint files, trail files, and BR files. The file system needs a shared file system because OGG will need to startup on different nodes in the event of planned or unplanned outages . The following options are available from Oracle at no additional cost:

- Oracle Automatic Storage Management (ASM) Cluster File System (ACFS). Oracle ACFS was introduced with Oracle Database 11g Release 2. See My Oracle Support note 948187.1, “ACFS Technical Overview and Deployment Guide”, for configuring ACFS.
- Oracle DataBase File System (DBFS). Oracle DBFS was introduced with Oracle Database 11g Release 2.
- Oracle Cluster File System (OCFS2) – available only on Linux.

## Oracle ASM Cluster File System (ACFS)<sup>1</sup>

Oracle Database 11g Release 2 introduces the Oracle Automatic Storage Management Cluster File System (ASM Cluster File System, ACFS). ACFS is a general purpose single-node (standalone) or

---

<sup>1</sup> With the initial release of Oracle Database 11.2.0.1 ACFS is not available on all platforms. Please check the release notes to ensure your platform is supported if you want to use ACFS.

multi-node cluster file system on top of ASM volume management (ADVM)). ACFS can be accessed using industry-standard Network Attached Storage (NAS) file access protocols: Network File System (NFS), iSCSI and Common Internet File System (CIFS).

The Oracle Grid Infrastructure ships with the Automatic Storage Management Clustered File System (ACFS). An ACFS file system will be mounted on all user defined nodes and file system availability is managed entirely by the Oracle Grid Infrastructure. As a result ACFS can be used to store Oracle GoldenGate processing files required to failover between nodes in case of a failure. To enhance and automate application availability, applications can define start/stop dependencies on the ACFS configured file system resource, ora.data.<vol>.acfs, so that the application is started, stopped or relocated as the ACFS resource comes online or goes offline. ACFS is the recommended HA file system for applications configured for Oracle Grid Infrastructure.

**Note:** ACFS can be used for Oracle Golden Gate trail files with no restrictions.

Oracle GoldenGate binaries can be stored on an ACFS filesystem and you can also store the recovery-related files in a cluster configuration in ACFS to make them accessible to all nodes; however, ACFS mounted on multiple servers with concurrent file access does not currently support file locking, thus, for this purpose, you would need to mount ACFS on only one server. If ACFS is mounted on one server at a time then file locking is supported. This single mounted cluster filesystem has fully integrated HA capabilities. NOTE: this limitation will not exist in the 11.2.0.4+ or 12.1 ACFS releases.

For more information about the Oracle Database 11g Release 2 ACFS, please refer to the Oracle Database Storage Administrator's Guide as part of the Oracle Database 11g Release 2 documentation set ([http://download.oracle.com/docs/cd/E11882\\_01/server.112/e10500/toc.htm](http://download.oracle.com/docs/cd/E11882_01/server.112/e10500/toc.htm)).

## Database File System (DBFS)

Oracle Database 11g Release 2 also introduces a Database File System (DBFS). In DBFS files are stored as secure files which are internally stored as LOB data values in the Oracle Database. In-database storage provides high availability, security and encryption capabilities that may not be otherwise available on general purpose file systems. In a cluster configuration the DBFS can be accessed from multiple nodes, and hence it can act as a cluster file system.

Files in DBFS can be managed through a set of PL/SQL APIs. In order to mount a DBFS as an OS file system another component, the DBFS client (dbfs\_client) is required. For Oracle Database 11.2.0.1 you can only mount a DBFS file system on Linux.

DBFS can be used for Oracle Golden Gate trail files with no restrictions. Oracle GoldenGate installation can be done on DBFS and you can also store the recovery-related files in a cluster configuration in DBFS to make them accessible to all nodes; however, DBFS mounted on multiple servers concurrently does not currently support file locking thus you would need to mount DBFS on only one server at a time.

GoldenGate bounded recovery files are supported on DBFS only from GoldenGate version 11.2.1 and Oracle Database 11g Release 2 (or higher).

For more information about DBFS, its restrictions as well as how to configure a DBFS, please refer to the Oracle Database SecureFile and Large Objects Developer's Guide as part of the Oracle Database 11g Release 2 documentation set

([http://download.oracle.com/docs/cd/E11882\\_01/appdev.112/e10645/toc.htm](http://download.oracle.com/docs/cd/E11882_01/appdev.112/e10645/toc.htm)).

## Oracle Cluster File System (OCFS2)

Cluster File System (OCFS2) is included in recent Linux distributions (included in the Linux kernel in some distributions). OCFS2 is an open source general purpose cluster file system. Refer to the OCFS2 website for more information: <http://oss.oracle.com/projects/ocfs2/>.

## Example of Creating an Oracle DBFS File System:

See My Oracle Support note [1054431.1](#) for configuring DBFS in an Oracle Clusterware environment. The support note includes the required action script.

## Example of Creating an Oracle ACFS File System:

For more details see [Oracle® Database Storage Administrator's Guide](#)

To create and verify an ACFS file system, perform the following steps:

1. Create an Oracle ASM volume in a mounted disk group with the ASMCMD `volcreate` command.

The compatibility parameters `COMPATIBLE.ASM` and `COMPATIBLE.ADVM` must be set to 11.2 or higher for the disk group. See "[Disk Group Compatibility Attributes](#)".

Start ASMCMD connected to the Oracle ASM instance. You must be a user in the OSASM operating system group. See "[About Privileges for Oracle ASM](#)".

When configuring Oracle ADVM volume devices within a disk group, Oracle recommends assigning the Oracle grid infrastructure user and Oracle ASM administrator roles to users who have root privileges.

To create a volume:

```
ASMCMD [+] > volcreate -G data -s 10G volume1
```

The volume name must be less than or equal to eleven alphanumeric characters, starting with an alphabetic character.

When creating an Oracle ASM volume, a volume device name is created that includes a unique Oracle ADVM persistent disk group number. The volume device file functions in the same manner as any other disk or logical volume to mount file systems or for applications to use directly.

For information about the `volcreate` command, see "[volcreate](#)".

2. Determine the device name of the volume that was created.

You can determine the volume device name with the ASMCMD `volinfo` command or from the `VOLUME_DEVICE` column in the `V$ASM_VOLUME` view.

For example:

```
ASMCMD [+] > volinfo -G data volume1
Diskgroup Name: DATA

      Volume Name: VOLUME1
      Volume Device: /dev/asm/volume1-123
      State: ENABLED
      ...

SQL> SELECT volume_name, volume_device FROM V$ASM_VOLUME
       WHERE volume_name = 'VOLUME1';

VOLUME_NAME          VOLUME_DEVICE
-----
VOLUME1              /dev/asm/volume1-123
```

For information about the `volinfo` command, see "[volinfo](#)".

#### See Also:

[Oracle Database Reference](#) for information about the `V$ASM_VOLUME` view

3. Create a file system with the Oracle ACFS `mkfs` command.

Create a file system using an existing volume device.

For example:

```
$ /sbin/mkfs -t acfs /dev/asm/volume1-123
mkfs.acfs: version                = 11.2.0.1.0.0
mkfs.acfs: on-disk version        = 39.0
mkfs.acfs: volume                 = /dev/asm/volume1-123
mkfs.acfs: volume size            = 10737418240
mkfs.acfs: Format complete.
```

See ["mkfs"](#) (Linux or UNIX) or ["acfsformat"](#) (Windows). The `root` privilege is not required. The ownership of the volume device file dictates who can run this command.

4. Mount the file system with the Oracle ACFS `mount` command.

For example:

```
# /bin/mount -t acfs /dev/asm/volume1-123 /u01/app/acfsmounts/myacfs
```

See ["mount"](#) (Linux or UNIX) or ["acfsmountvol"](#) (Windows). The `root` privilege is required to run the `mount` command and the Windows Administrator privilege is required to run the `acfsmountvol` command.

After the file system has been mounted, ensure that the permissions are set to allow access to the file system for the appropriate users. For example:

```
# chown -R oracle:dba /u01/app/acfsmounts/myacfs
```

## Register File System with Clusterware

The filesystem created in the previous step will need to be mounted by clusterware for OGG on a failover or node switchover. In order for this to happen the filesystem must be registered with clusterware. The clusterware resource name for the file system will be referenced later when configuring the GoldenGate Clusterware Agents.

### Example of Registering DBFS File System with Clusterware:

This is a sample from Support note [1054431.1](#) for configuring DBFS in a Clusterware environment. Please reference the above note for more details on how to create the DBFS file system and the action script.

Register the Clusterware resource by executing the following sample script as the RDBMS owner of the DBFS repository database (typically `oracle`) user. The `ORACLE_HOME` and `DBNAME` should reference your Grid Infrastructure `ORACLE_HOME` directory and your DBFS

repository database name, respectively. If mounting multiple file systems, you may also need to modify the ACTION\_SCRIPT and RESNAME. You will be using the resource name (RESNAME) value for the “filesystems” parameter when creating the GoldenGate Clusterware Agent with the “agctl add” command.

```
##### start script add-dbfs-resource.sh
#!/bin/bash
ACTION_SCRIPT=/u01/app/11.2.0/grid/crs/script/mount-dbfs.sh
RESNAME=mydbfs
DBNAME=orcl
DBNAMEL=`echo $DBNAME | tr A-Z a-z`
ORACLE_HOME=/u01/app/11.2.0/grid
PATH=$ORACLE_HOME/bin:$PATH
export PATH ORACLE_HOME
crsctl add resource $RESNAME \
  -type local_resource \
  -attr "ACTION_SCRIPT=$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"
##### end script add-dbfs-resource.sh
```

Then run this as the Grid Infrastructure owner (typically `oracle`) on one database server only:

```
$ sh ./add-dbfs-resource.sh
```

## Example of Registering ACFS File System with Clusterware:

Register the file system with the `acfsutil registry` command.

For example:

```
# /sbin/acfsutil registry -a /dev/asm/volume1-123 /u01/app/acfsmounts/myacfs
acfsutil registry: mount point /u01/app/acfsmounts/myacfs successfully added
to Oracle Registry
```

See ["acfsutil registry"](#). The `root` or `asmadmin` privileges are required to modify the registry.

The Oracle Clusterware resource for the file system will be created automatically with this command. The resource name will be required for the “filesystems” parameter later when creating the GoldenGate Clusterware Agent with the “agctl add” command. In this case the resource name would be “ora.data.myacfs.acfs”. You can determine that resource name automatically created by using the following command:

```
$ $GRID_HOME/bin/crsctl stat res -t | more
```

Sample output:

NAME	TARGET	STATE	SERVER	STATE_DETAILS
-----				
Local Resources				
-----				
ora.DATA.dg	ONLINE	ONLINE	src node1	
	ONLINE	ONLINE	src node2	
ora.LISTENER.lsnr	ONLINE	ONLINE	src_node1	
	ONLINE	ONLINE	src_node2	
<b>ora.data.myacfs.acfs</b>				
	ONLINE	ONLINE	src_node1	mounted on
/u01/app/acfsmounts/myacfs				
	ONLINE	ONLINE	src_node2	mounted on
/u01/app/acfsmounts/myacfs				

## Create an OGG administrative user account and OGG database user account

The OGG Instance will require an OS user account and a database user account. Please follow the instructions in the Oracle GoldenGate Installation Guide for complete details.

## Install OGG Software

Oracle GoldenGate binaries must be available on every server in the same location (e.g. /u01/app/ggate). Current best practices are to do a local installation on every server, but you can do a single installation on a shared file system. You will need shared storage for the recovery-related files. On a Unix/Linux platform you can use a symbolic link to a central location for the shared directories.

The environments must be identical on all nodes in the cluster where OGG expected to run so that OGG and Oracle Clusterware execution, log and configuration files are available on all nodes.

Please follow the steps as described in the Oracle GoldenGate Installation Guide for the specific software installation steps.

## Create GoldenGate VIP and GoldenGate Agent

The application VIP can be created in advance by the Grid Admin as root using the `appvipcfg` command. If the application VIP is created in advance only the `--vip_name` needs to be defined in the `agctl add` command and the command can be executed as the GoldenGate admin.. If you are creating the application VIP with `agctl`, the command must be run as root, the `--vip_name` should not be defined and the `--network`, `--ip` and `--user` flags must be defined.

## Configuring the GoldenGate VIP and GoldenGate Agent Separately as the Oracle Clusterware Administrator

This procedure requires that a VIP be created as a separate step by the Oracle Clusterware Administrator using the `appvipcfg` command.

As the Oracle Clusterware Administrator, use the following command to determine the network number:

```
$ $GRID_HOME/bin/crsctl stat res -p |grep -ie .network -ie subnet |grep -ie name -ie subnet
```

Sample output is:

```
NAME=ora.net1.network
USR_ORA_SUBNET=X.X.X.0
NAME=ora.net2.network
USR_ORA_SUBNET=X.X.X.0
```

**NOTE:** There may be multiple networks defined in the cluster and it is at the discretion of the Oracle Clusterware Administrator and the Oracle GoldenGate Administrator to choose the correct network based on the required interface and subnet.

To create the application VIP, login as root and run:

```
# $GRID_HOME/bin/appvipcfg create
-network=1 \
-ip=X.X.X.149 \
-vipname= xag.gg_1-vip.vip \
-user=oracle \
```

`network` – The network number returned from the previous command

`ip` – The ip address assigned to the GoldenGate VIP

`vipname` – The name assigned to the GoldenGate VIP

**user** – The user who owns the VIP

As **root**, allow the Oracle Grid infrastructure software owner (e.g. **oracle**) to start the VIP.

```
# $GRID_HOME/bin/crsctl setperm resource xag.gg_1-vip.vip -u user:oracle:r-x
```

Then, as **oracle**, start the VIP:

```
$ $GRID_HOME/bin/crsctl start resource xag.gg_1-vip.vip
```

To validate whether the VIP is running and on which node it is running, execute:

```
$ $GRID_HOME/bin/crsctl status resource xag.gg_1-vip.vip
```

Sample output:

```
NAME= xag.gg_1-vip.vip
TYPE=app.appvip.type
TARGET=ONLINE
STATE=ONLINE on src_node1
```

**OGG Administrator**, **oracle**, creates **OGG Agent** as shown in the following example (This example would be executed on the source system):

```
$ $XAG_HOME/bin/agctl add goldengate gg_1 \
--gg_home /myacfs \
--instance_type source \
--nodes src_node1, src_node2 \
--vip_name xag.gg_1-vip.vip \
--filesystems ora.data.myacfs.acfs \
--databases orcl \
--oracle_home /u01/app/oracle/product/11.2.0/dbhome_1 \
--monitor_extracts ext1,pmp1
```

**gg\_home** – Location of OGG binaries

**instance\_type** – OGG source or OGG target (source, target)

**nodes** – The hostname of the nodes in the cluster

**vip\_name** – The application vip created in the previous step.

**filesystems** – The filesystem resource of the OGG shared filesystem (ACFS, DBFS, etc)

**database** – The database name

**oracle\_home** – The \$ORACLE\_HOME location

**monitor\_extracts** – The extracts to monitor

`monitor_replicats` – The replicats to monitor

This example will output the status of the agent created above:

```
$ $XAG_HOME/bin/agctl status goldengate gg_1
Goldengate instance 'gg_1' is not running
```

## Configure VIP and GoldenGate Agent in one command

This procedure requires that the command be ran as root since it will be creating the VIP at the same time that the OGG resources will be created. It will create the VIP and then start it. This approach is not possible with a DBFS filesystem due to the fact that root is not aware of the DBFS filesystem.

Root user Creates GoldenGate Agent and VIP in one command as shown in the following example:

```
$ $XAG_HOME/bin/agctl add goldengate gg_1
--gg_home /myacfs \
--instance_type source \
--nodes src_node1,src_node2 \
--network 1 \
--ip X.X.X.160 \
--filesystems ora.data.myacfs.acfs \
--user oracle \
--group oinstall \
--databases orcl \
--oracle_home /u01/app/oracle/product/11.2.0/dbhome_1 \
--monitor_extracts ext1,pmp1
```

`gg_home` – Location of OGG binaries

`instance_type` – OGG source or OGG target

`nodes` – The hostname of the nodes in the cluster

`network` – The network number as noted earlier in the `crsctl` output.

`ip` - The ip address assigned to the GoldenGate VIP

`filesystems` – The filesystem resource of the OGG shared filesystem (ACFS, DBFS, etc.)

`user` – The user who will control this agent

`group` – The group of the user who will control this agent

`databases` – The database name

`oracle_home` – The `$ORACLE_HOME` location

`monitor_extracts` – The extracts to monitor

Check the status of the running resource using:

```
$ $XAG_HOME/bin/agctl status goldengate gg_1
Goldengate instance 'gg_1' is not running
```

This example will output the status of the vip created above:

```
$ $GRID_HOME/bin/crsctl stat res xag.gg_1-vip.vip

NAME=xag.gg_1-vip.vip
TYPE=app.appvip_net1.type
TARGET=OFFLINE
STATE=OFFLINE
```

Refer to <http://oracle.com/goto/clusterware> for more details and links to specific Oracle Clusterware documentation on specific commands.

## Create the OGG Processes

The following steps are examples of how to setup a basic configuration of OGG.

1. Enable Supplemental logging on source database tables. Make sure the ORACLE\_SID is set in the environment and minimal supplemental logging enabled in the database.

```
GGSCI> DBLOGIN USERID ggadm1 PASSWORD ggs123

GGSCI> ADD TRANDATA <schema>.*

Or

GGSCI> ADD SCHEMATRANDATA <schema>
```

2. Create Manager Parameter file on Source and Target

The manager process requires that the following lines exist in the mgr.prm file in order for extract and/or replicat processes to be restarted by the GoldenGate Agents on a failover or started on an GoldenGate Agent startup.

```
AUTORESTART ER *, RETRIES 5, WAITMINUTES 1, RESETMINUTES 60
AUTOSTART
```

3. Create Extract Process on Source

- a. Register Extract with database for Integrated Extract. Make sure the ORACLE\_SID is set in the environment.

```
GGSCI> DBLOGIN USERID ggadm1 PASSWORD ggs123
GGSCI> REGISTER EXTRACT ext1 DATABASE
```

- b. Create Extract

```
For Integrated Extract
GGSCI> ADD EXTRACT ext1, INTEGRATED TRANLOG, BEGIN NOW
```

```
For Classic Extract
GGSCI> ADD EXTRACT ext1, TRANLOG, THREADS 4, BEGIN NOW
```

```
Create Local Trail File
GGSCI> ADD EXTTRAIL ./dirdat/et, EXTRACT etest, MEGABYTES 100
```

#### 4. Create Extract Pump on Source

```
GGSCI> ADD EXTRACT pmp1, exttrailsources ./dirdat/et
```

```
Create Remote Trail File
GGSCI> ADD RMTRAIL ./dirdat/rt, EXTRACT pmp1, MEGABYTES 100
```

5. Edit Extract Pump parameter file. The RMTHOST parameter must reference the resource VIP IP of the target for automatic failover in an OGG XAG Configuration. The PARAMS – UL option was added to insure that the pump will restart even if a trail file was momentarily locked by the old server collector process on a failover. The timeout is set to 120 to insure that an old server collector process will shut down after 120 seconds if it loses communication.

Get VIP IP from Target Cluster:

Where *xag.gg\_2-vip.vip* is the vipname on target

```
$ GRID_HOME/bin/crsctl stat res xag.gg_2-vip.vip -p|grep USR_ORA_VIP|grep -v "GEN"
```

```
USR_ORA_VIP=X.X.X.160
```

```
RMTHOST Parameter in Extract Pump:
RMTHOST X.X.X.160, MGRPORT 9998, TIMEOUT 120, PARAMS -UL
```

6. Create Replicat on Target . The checkpoint table is critical to insure that data is not replayed in the event of a failover.

```
GGSCI> DBLOGIN USERID ggadmin, PASSWORD ggs123
```

```
GGSCI> ADD REPLICAT repl, exttrailsource ./dirdat/rt, CHECKPOINTTABLE
ggckpt
```

## Start OGG Agent

Since the manager must be up on the target for the pump extract to start, the target OGG Agent should be started first. In this example, `gg_1` is the source agent and `gg_2` is the target agent.

node – node where agent should be started

On Target Cluster execute:

```
$ $XAG_HOME/bin/agctl start goldengate gg_2 --node trg_node1
```

On Source Cluster execute:

```
$ $XAG_HOME/bin/agctl start goldengate gg_1 --node src_node1
```

Where `trg_node1` is the node in the target cluster where you want the replicat to run and `src_node1` is the node in the source cluster where you want the extract to run.

## Status OGG Agent and VIPs

Check the Status of the Agent and VIPs.

Source Cluster:

Agent Status

```
$ $XAG_HOME/bin/agctl status goldengate gg_1
```

VIP Status

```
$ GRID_HOME/bin/crsctl stat res xag.gg_1.goldengate
```

Target Cluster:

Agent Status

```
$ $XAG_HOME/bin/agctl status goldengate gg_2
```

VIP Status

```
$ $GRID_HOME/bin/crsctl stat res xag.gg_2.goldengate
```

## Status Cluster Resources

The following command shows the status of all cluster resources.

```
$ $GRID_HOME/bin/crsctl status resource -t
```

## Test OGG Agent Failover

Relocate the OGG Processes to another node to test failover (login as oracle):

node – node where agent should be relocated

Target:

```
$ $XAG_HOME/bin/agctl relocate goldengate gg_2 --node trg_node2
```

Source:

```
$ $XAG_HOME/bin/agctl relocate goldengate gg_1 --node src_node2
```

## Cleanup Commands

If you want to stop Oracle Clusterware Agent from managing the Oracle GoldenGate processes, and you want to cleanup the changes you made, then:

Stop Oracle GoldenGate XAG(login as oracle):

```
$ $XAG_HOME/bin/agctl stop resource gg_1
```

Stop the VIP (as oracle):

```
$ $GRID_HOME/bin/crsctl stop resource xag.gg_1-vip.vip
```

Delete the application gg\_1 as the application owner (oracle) or root:

This command will delete the agent and vip if the agent and vip were created in one command and this command is executed as root. Otherwise, this command will only delete the agent.

```
$ $XAG_HOME/bin/agctl delete resource gg_1
```

Delete the VIP (login as root):

```
# $GRID_HOME/bin/appvipcfg delete -vipname=xag.gg_1-vip.vip
```

## Conclusion

This paper has presented many different Oracle technologies working together to provide a comprehensive HA solution for Oracle GoldenGate. Oracle Clusterware is a full feature cluster management solution. Starting with Oracle Database 10g Oracle mandates its users to use Oracle's Clusterware for Oracle RAC cluster management. Starting with Oracle Database 10g Release 2 there are commands to register and manage other applications through Oracle Clusterware.

This best practice document discusses how to address Oracle GoldenGate failover during server failures by using a step-by-step example to manage Oracle GoldenGate through Oracle Clusterware XAG. Oracle GoldenGate high availability through Oracle Clusterware will help keep Oracle GoldenGate available through operating system and hardware crashes as long as the database is accessible from a surviving node. It should be noted however that there may still be other reasons for Oracle GoldenGate processes to become unavailable and you should continue to use notification procedures to ensure ultimate high availability for Oracle GoldenGate.

Before the launch of the GoldenGate Clusterware Agents, MOS Note 1313703.1 was the recommended approach for integrating Oracle GoldenGate with Clusterware. Using the Oracle GoldenGate Clusterware Agent is now the Best Practice for integrating Oracle GoldenGate with Clusterware in an Oracle 10.2.0.5+ environment. The MOS Note 1313703.1 should only be used in environments not supported by the Oracle GoldenGate Clusterware Agents.



Oracle GoldenGate Best Practices: Configuring  
Oracle GoldenGate with Oracle Grid  
Infrastructure Bundled Agents (XAG)  
September 2012  
Author:s Tracy West, Sourav Bhattacharya  
Contributing Authors: John McHugh

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com



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

Copyright © 2012, 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.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. 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. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

**SOFTWARE. HARDWARE. COMPLETE.**