Transitioning E-Business Suite to the Maximum Availability Architecture on HP Systems: E-Business Suite 11i.10.2 and Database 10gR2

An Oracle White Paper
May 2007

Maximum Availability Architecture

Oracle Best Practices For High Availability

Transitioning E-Business Suite to the Maximum Availability Architecture on HP Systems

Transitioning E-Business Suite to the Maximum Availability Architecture on HP Systems

EXECUTIVE SUMMARY

Oracle Maximum Availability Architecture (MAA) is Oracle's best practice blueprint based on proven Oracle high-availability technologies and recommendations. The goal of MAA is to remove the complexity in designing the optimal high-availability architecture.

Published as part of the MAA series of white papers, this paper focuses on transitioning an E-Business Suite application to an MAA configuration while keeping application downtime to a minimum. The most current version of this document is located at Oracle Maximum Availability Architecture (MAA).

The starting point for our process is Oracle E-Business Suite 11.5.10.2 on a single instance (non-RAC) Oracle 10gR2 database. Our final configuration is Oracle E-Business Suite 11.5.10.2 running on a two node 10gR2 RAC cluster with Flashback, ASM, Oracle Clusterware, HP Serviceguard Extension for RAC (release A.11.17) and a disaster recover site leveraging Oracle Data Guard.

During our tests we were able to limit E-Business Suite application downtime to 5 minutes for the transition to MAA. We also formulated an approach for the switchover and failover to the disaster site that limited E-Business Suite application downtime to 5 minutes.

This project was a joint Oracle and HP case study and the steps that are presented here were tested on HP systems. The goal was to minimize application downtime by utilizing the latest products from HP and Oracle.

INTRODUCTION

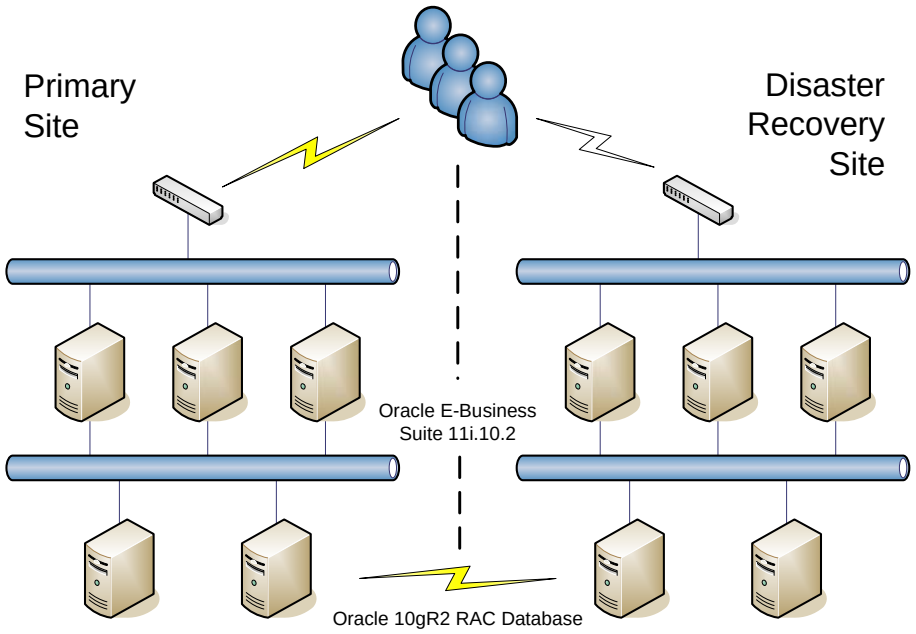
In this section we will introduce the technologies we use to achieve MAA.

- Oracle Real Application Clusters
- Oracle Data Guard
- Oracle Flashback
- Oracle Automatic Storage Management
- HP Serviceguard
- HP Serviceguard Extension for RAC
- Shared Logical Volumes
- Multi-Node Packages

We will also explain our approach to transitioning to MAA and provide more details on the minimal application downtime that was achieved.

Finally, we will describe our example environment and document conventions.

Oracle Maximum Availability Architecture (MAA)



Oracle Real Application Clusters

Oracle Real Application Clusters (RAC) allows the Oracle database to run any packaged or custom application unchanged across a set of clustered nodes. This capability provides the highest levels of availability and the most flexible scalability. If a clustered node fails, the Oracle database will continue running on the surviving

nodes. When more processing power is needed, another node can be added without interrupting user access to data.

Oracle Data Guard

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive failures, disasters, errors, and data corruption. Data Guard maintains these standby databases as transactionally consistent copies of the production database. If the production database becomes unavailable due to a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus greatly reducing the application downtime caused by the outage. Data Guard can be used with traditional backup, restore, and clustering solutions to provide a high level of data protection and data availability.

Oracle Flashback

Flashback Database quickly rewinds an Oracle database to a previous time, to correct any problems caused by logical data corruptions or user errors. Flashback Database is like a 'rewind button' for your database. It provides database point in time recovery without requiring a backup of the database to first be restored. When you eliminate the time it takes to restore a database backup from tape, database point in time recovery is fast.

In this paper we document how Flashback can be used to return the production database to standby mode after failover without the need for a lengthy database restore, thus accelerating the return to normal production operation after a disaster.

Oracle Automatic Storage Management

Automatic Storage Management (ASM) provides a vertically integrated file system and volume manager directly in the Oracle kernel, resulting in:

- Significantly less work to provision database storage
- Higher levels of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Unique capabilities for database applications

For optimal performance, ASM spreads files across all available storage. To protect against data loss, ASM extends the concept of SAME (stripe and mirror everything) and adds more flexibility in that it can mirror at the database file level rather than the entire disk level[1].

[1] It can be said that ASM is the implementation and automation of SAME.

Oracle Clusterware

High availability configurations have redundant hardware and software that maintain operations by avoiding single points of failure. To accomplish this, the Oracle Clusterware is installed as part of the RAC installation process. Oracle Clusterware is a portable solution that is integrated and designed specifically for the Oracle database[2]. In a RAC environment, Oracle Clusterware monitors all Oracle resources (such as instances and listeners). If a failure occurs, Oracle Clusterware will automatically attempt 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 in the cluster fails, Oracle Clusterware will relocate services being used by the application and notify client processes to reconnect to a surviving node.

HP Serviceguard

HP has been delivering high-availability and disaster tolerant solutions for more than 25 years. HP Serviceguard, with more than 150,000 licenses sold worldwide, is designed to protect mission-critical applications from a wide variety of hardware and software failures.

HP Serviceguard builds upon the concept of virtualization by grouping multiple servers into a cluster to provide highly available application services that enhance data integrity. Within the cluster, HP Serviceguard closely monitors the health and status of software and hardware components and uses cluster management tools to efficiently manage systems within the cluster.

HP Serviceguard Extension for RAC (SGeRAC)

HP Serviceguard and HP SGeRAC are tightly integrated to exploit the best features of HP enterprise clusters and Oracle relational database servers. SGeRAC provides all of the functionality needed to support an Oracle RAC environment on HP 9000 and HP Integrity servers. The major components of Serviceguard Extension for RAC are listed here.

- Cluster Manager: provides the lower level cluster services for Oracle Clusterware by establishing and monitoring the cluster members and monitoring various components within each node.
- Cluster Membership: informs Oracle Clusterware about system failures to facilitate fast database recovery.
- Network Manager: detects and recovers from network card and cable failures with the use of redundant network components, transparent to the Oracle software.

[2] Although it is optimized for the Oracle Database, Oracle Clusterware can support third party applications as well.

- Shared Logical Volume Manager: provides the basic functionality to share physical disks and buses between the nodes
- Package Manager: monitors and controls SGeRAC packages containing highly available applications.

HP Shared Logical Volumes

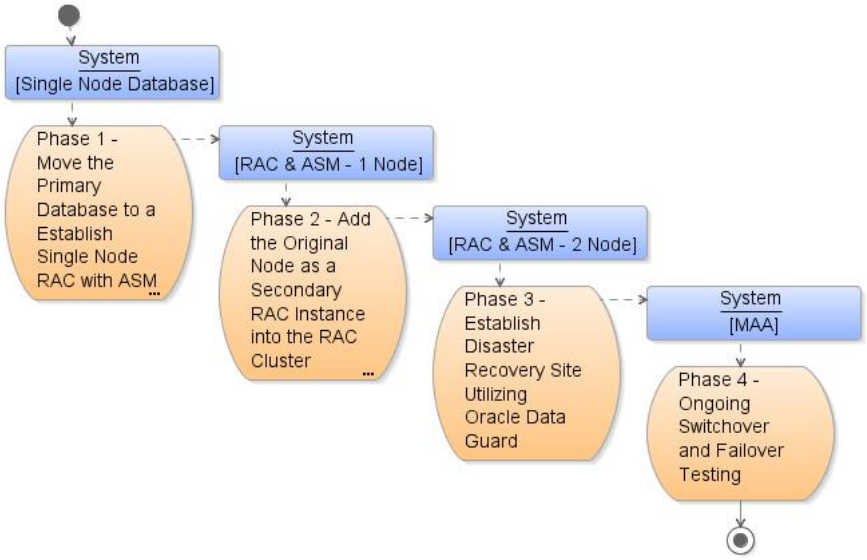
In SGeRAC A.11.17, ASM disk groups utilize raw logical volumes managed by Shared Logical Volumes Manager (SLVM). SLVM ensures that the shared disk groups have the same names on all nodes and protects ASM data against inadvertent overwrites from nodes inside, as well as outside the cluster. HP-UX also provides multiple physical paths for logical volumes, using either the PV Links feature from SLVM, or separate products such as HP StorageWorks Secure Path.

HP SGeRAC Multi-Node Packages (MNP)

HP SGeRAC A.11.17 and Oracle 10gR2 support the creation of an Oracle Clusterware package. These packages are called Multi-Node Packages and do not failover or failback between nodes. The MNP reside on each RAC node to monitor and support the startup and shutdown of the Oracle Clusterware service.

Approach

To achieve the transition with minimal application downtime we took a four-phased approach:



Phase 1 - Move the Primary Database to a Establish Single Node RAC with ASM

In the first phase we create a single node 10gR2 RAC database instance on a new database server utilizing HP Serviceguard Extension for RAC, Oracle Clusterware and ASM. We then use a backup of the production database and convert this RAC

database instance into a local Data Guard physical standby for the production database on the new server. Finally, we switchover to the new database server and enable Flashback Database, and it becomes our production environment.

Phase 2 - Add the Original Node as a Secondary RAC Instance into the RAC Cluster

Once the application has stabilized on the new database server it is possible to take the original database server out of service. We then install the cluster related hardware and establish HP Serviceguard Extension for RAC, Oracle Clusterware, ASM and RAC on the old production server and add it to the cluster, creating a two node RAC cluster.

Phase 3 - Establish Disaster Recovery Site Utilizing Oracle Data Guard

To complete the MAA it is also necessary to establish a disaster recovery site. To do this we establish a two node 10gR2 RAC database cluster on servers at the DR site utilizing HP Serviceguard Extension for RAC, Oracle Clusterware and ASM. We instantiate the Data Guard physical standby using backups taken from the production database. We also clone the E-Business Suite application tier software from the primary site to the disaster site. The standby system will apply changes received from the primary site so that it stays up to date and will be ready to take over in the event of an emergency.

Phase 4 - Ongoing Switchover and Failover Testing

The Disaster Recovery site can be used via switchover to provide application services while the production platform or site is undergoing planned maintenance or via failover if the production site is rendered unusable by a severe unplanned outage. In this phase we test the switchover and failover procedures. We recommend that switchover and failover procedures are tested regularly to validate the MAA configuration.

Application Downtime

An important goal of the project was to minimize the application downtime during the transition so that this would not become an obstacle to establishing MAA.

During our tests we were able to limit E-Business Suite Application downtime to 5 minutes for the switch over to the RAC database. Here are details of the downtime steps:

Downtime Steps to Switch to RAC Database	Time (m:ss)
Step 1.9.3 - Switchover to Standby Database	0:43
Step 1.9.4 - Enable Flashback	0:01
Step 1.9.5 - Open the Database	0:05
Step 1.9.6 - Remove the Old Application Topology	0:02

Downtime Steps to Switch to RAC Database	Time (m:ss)
Step 1.9.7 - Configure Database Tier	1:34
Step 1.9.8 - Restart Listeners	0:02
Step 1.9.9 - Configure Application Tiers	2:50
Total	**5:17**

During our disaster simulation test we were able to limit E-Business Suite Application downtime to 5 minutes for the failover to a disaster site. Here are details of the downtime steps:

Downtime Steps to Failover to Disaster Recovery Site	Time (m:ss)
Step 4.3.1 - Failover to DR Database	0:09
Step 4.3.2 - Enable Flashback	0:01
Step 4.3.3 - Open the DR Database	0:12
Step 4.3.5 - Remove the Old Application Topology	0:02
Step 4.3.6 - Configure DR Database Tier	1:34
Step 4.3.7 - Restart Listeners	0:02
Step 4.3.8 - Configure Application Tiers	2:50
Total	**4:50**

Note, the time to shutdown and startup the application services was not included in our downtime estimates as this can vary considerably in each environment. In our test environment the application shutdown and startup took approximately 3 minutes.

Test Environment

The primary site consisted of three HP 9000 rp3440 servers for the application tier, two HP Integrity rx4640 servers for the database tier, and a single HP StorageWorks EVA4000 for the shared SAN storage.

The disaster recovery site consisted of one HP 9000 rp3440 server for the application tier, two HP Integrity rx4640 servers for the database tier, and a single HP StorageWorks EVA4000 for shared SAN storage. All servers were running HP-UX 11i v2.

E-Business Suite 11.5.10.2 on Oracle Database 10.2.0.5 running the Vision Demo database was established as a starting point for testing.

Customers with an HP PA-RISC homogeneous environment for their E-Business Suite 11i stack will still be able to use this paper as a reference guide. Another option will be to migrate the database to an HP Integrity server prior to starting the transition. See Oracle*Metalink* note: 304489.1.

The examples we show in this document are derived from the test environment. The following table shows the node names and configuration information that was used at each stage of the transition.

	Primary Site			Disaster Site
	Single Node Database	**Single Node RAC and ASM**	**Two Node RAC and ASM**	
Application-Tier Nodes	ha1mth01 ha1mth02 ha1mth03	ha1mth01 ha1mth02 ha1mth03	ha1mth01 ha1mth02 ha1mth03	ha2mth01
APPL_TOP	/u01/appltop	/u01/appltopRAC	/u01/appltopRAC	/u01/appltopRAC
Database-Tier Nodes	ha1dbh	ha1dbh01	ha1dbh01 ha1dbh02	ha2dbh01 ha2dbh02
ORACLE_HOME	/u01/app/oracle/visdb/10.2.0	/u01/app/oracle/visdbRAC/10.2.0	/u01/app/oracle/visdbRAC/10.2.0	/u01/app/oracle/visdbRAC/10.2.0
Instance Names	VIS	VIS1	VIS1 and VIS2	VIS1 and VIS2
Unique DB Name	VIS	VIS_ha1	VIS_ha1	VIS_ha2

Database Instance Configuration

When configuring a database in an Oracle E-Business Suite environment it is essential to make changes in the "ifile" (the parameter "include file") so the AutoConfig utility does not overwrite them. The ifile is located at:

```
$ORACLE_HOME/dbs/<CONTEXT_NAME>_ifile.ora
```

The Oracle Database Reference is an excellent reference for all database parameters.

It will be necessary to configure the original instance and the RAC instances during the transition to RAC, and the primary and standby instances when we establish the DR site. In this document we show the configurations we used for our test environment as examples. We recommend that you prepare similar configuration files and scripts for your environment ahead of time and have them available to use at the appropriate steps.

We have described some of the parameters in more detail below:

parallel_execution_message_size

It is recommended that this parameter be set as follows for optimal redo processing on the standby database.

```
parallel_execution_message_size=8192
```

db_recovery_file_dest

This parameter represents the location of Oracle's Flash Recovery Area. The Flash Recovery Area holds all of Oracle's recovery related files, and more importantly automates the management of those files based on user specified settings. This automation significantly simplifies the administration and configuration of recovery-related files such as backup sets, image copies, archive log files, flashback log files, etc. The Flash Recovery Area can be located on a filesystem or in an ASM diskgroup.

db_recovery_file_dest_size

This parameter represents the amount of space the Flash Recovery Area is allowed to use. The recommended size depends on the application but for the Oracle recommended backup strategy, the Flash Recovery Area is generally 2.5 times larger than the database.

db_file_name_convert and log_file_name_convert

These parameters ensure that database and log file names are matched up properly when the names are not the same in a primary controlfile and a standby controlfile[3].. Note they cannot be set dynamically, so their settings have to be considered carefully to avoid unnecessary downtime.

fal_server and fal_client

These parameters automate the detection and fetching of log sequence gaps between the primary and the standby. For example, gaps can occur if managed recovery is turned off on the standby during a time when many logs are archived on the primary.

For the temporary local standby system set up to convert the production system to RAC and ASM, we used EZConnect syntax in these parameters. This simplified our TNS settings for the interim configuration.

For permanent communications between the production and DR site, we defined and referenced services for these parameters.

db_unique_name

This is a name that differentiates databases running in a Data Guard standby configuration.

Conventions

Convention	Meaning

[3] This is generally the case when using ASM since ASM uses the db_unqiue_name as part of datafile path

Convention	Meaning
Production or primary system	Initial Applications system
applmgr	User that owns the applications file system (APPL_TOP and application tier technology stack)
oracle	User that owns the database (ORACLE_HOME and database files).
CONTEXT_NAME	The CONTEXT_NAME variable refers to the name of the Applications Context file, typically in the format <SID>_<HOSTNAME>.
Mono-spaced Text	Represents command line text. Type this command exactly as shown (except as noted above).
< >	Text enclosed in angle brackets represents a variable. Substitute a value for the variable text. Do not type the angle brackets.
Application Tier	Comprised of all nodes in a single site running the Admin, Concurrent Processing, Forms and Web servers.
APPL_TOP	Folder containing applications software.
Shared APPL_TOP	APPL_TOP located on a shared file system so that multiple nodes can use it.
Database Tier	Comprised of all nodes in a single site running the RDBMS database.
RAC Database	Database running with Oracle RAC.
RAC Instance	One instance of a RAC Database.
ORACLE_HOME	Database / tech stack software location.
ORACLE_SID	Database Service Identifier.
#	OS-level command preceded with # is executed as root.
$	OS-level command preceded with $ is executed as oracle or applmgr.
SQL>	Command preceded with SQL> is executed using SQL*Plus.
RMAN>	Command preceded with RMAN> is executed using the RMAN command tool.

PREREQUISITES

This note is not a substitute for developing a proper understanding of E-Business Suite Deployment, RAC, ASM, Data Guard, SGeRAC and other technologies. To implement and manage ASM, RAC and a physical standby database, you must first read and understand the appropriate documentation. Please see the section entitled References for suggested reading.

This paper assumes that the following prerequisites are in place before beginning the transformation process.

HP Operating Environment

HP-UX 11i v2 0505 Enterprise or Mission Critical Operating Environment or later should be installed.

HP-UX minimum kernel settings

Always retain the higher value between current and minimum value.

Kernel Parameter	Minimum Value
EXECUTABLE_STACK	0
KSI_ALLOC_MAX	32768
MAX_THREAD_PROC	1024
MAXDSIZ	1073741824
MAXDSIZ_64BIT	2147483648
MAXSSIZ	134217728
MAXSSIZ_64BIT	1073741824
MAXUPRC	3687
MSGMAP	4098
MSGMNI	4096
MSGSEG	32767
MSGTQL	4096
NCSIZE	35840
NFILE	63488
NFLOCKS	4096
NINODE	34816
NKTHREAD	7184
NPROC	4096
SEMMNI	4096
SEMMNS	8192
SEMMNU	4092
SEMVMX	32767
SHMMAX	1073741824
SHMMNI	512
SHMSEG	120
VPS_CEILING	64
VX_NINODE (2xNFILES)	126976

HP-UX minimum patching level

Patch	Description
PHKL_33562	reboot(1M) cumulative patch
PHKL_33562	mmap64(2) size limitation, madvise(2) fix
PHKL_33563	VM mmap(2) and madvise(2) fix
PHKL_33820	mmap(2) size limit;mutex unlock PA 64 bit
PHNE_33655	NFS cumulative patch
PHSS_32502	Aries cumulative patch
PHSS_33276	Math Library cumulative patch
PHSS_33277	HP C Compiler (A.06.02)
PHSS_33278	aC++ complier (A.06.02)
PHSS_33279	u2comp/be/plugin library patch
PHSS_34440	linker + fdp cumulative patch
PHSS_34445	milli cumulative patch

MirrorDisk/UX B11.23

HP disk mirroring software was used to setup the internal disks in a mirrored configuration to provide disk redundancy for the HP-UX operating system and local filesystems.

Shared SAN Storage

HP StorageWorks Enterprise Virtual Array (EVA) 4000 and redundant HP StorageWorks 2/16 SAN Switches were installed at each site to provide shared storage for the database nodes. For information concerning HP StorageWorks EVA4000, please refer to the following URL:

http://www.hp.com/go/eva4000

For instructions on setting up and configuring the EVA4000, please refer to the "HP StorageWorks 4000/6000/8000 Enterprise Virtual Array user guide." This document is available on the HP web site at the following URL:

http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00748660/c00748660.pdf

The EVA4000s were configured identically for both sites with 3 Logical Units (LUN) created using HP CommandView for EVA. One 3GB LUN used for CRS and the other two 500GB LUNs were used for DATA and Flash Recovery. Multiple FC HBAs were installed in the database nodes and redundant FC Switches were used to provide multiple physical paths between the database nodes and the shared storage.

Run Cluster Verification

Cluster Verification Utility (CVU) is a single tool that can verify all prerequisites when building a cluster. This includes operating system patches, shared storage accessibility, user equivalence and many more. It is recommended that CVU be run at each stage while building a cluster to ensure prerequisites are met.

The CVU download and FAQ can be found at http://www.oracle.com/technology/products/database/clustering/cvu/cvu_download_homepage.html.

Oracle E-Business Suite

For our testing we installed E-Business Suite 11.5.10.2 with the following patches and patch sets. We recommend that the same release and patch level is attained before beginning the transition to MAA:

- Latest AD.I Minipack - 4712852
- 11i.ATG_PF.H RUP4 or later - 4676589
- TXK Autoconfig Template Rollup Patch M - 4709948
- Post ADX-F Fixes - 5225940
- To improve the performance of the auto configuration script (adautocfg) - 4637088
- 5456078 – CloneContext: Skip DB port check if VALIDATE=NO on application tier
- 2873456 – Enable EditContext on the database tier
- For a list of other required database patches, refer to Interoperability Notes - Oracle Applications 11i with Oracle Database 10g Release 2.

We assume that the application is configured with multiple application tier nodes fronted by a hardware load balancer.

Oracle Database

Our initial production database was a single instance Oracle DB 10.2.0.2. No addition patches are required.

It is assumed that the database is in archive log mode. If this is not the case then this should be scheduled and performed before the transition is started. Note that this will require some database downtime.

It is strongly recommended that the database be configured with mirrored online redo logs and control files.

It is assumed that sufficient network bandwidth to support Oracle Data Guard is available between the primary and disaster recovery site.

PHASE 1 - MOVE THE PRIMARY DATABASE TO A ESTABLISH SINGLE NODE RAC WITH ASM

Task 1.1 - Establish HP Serviceguard and Shared Logical Volumes

Before beginning this process we recommend you read the following documents:

Serviceguard Version A.11.17 Release Notes

Serviceguard Extension for RAC Version A.11.17 Release Notes

Using Serviceguard Extension for RAC

Step 1.1.1 - Apply the SGeRAC Patch Bundle to the New Database Node for Site1

Refer to Serviceguard Extension for RAC Version A.11.17 Release Notes for a list of required SGeRAC patches. It is recommended to group the patches together in a patch bundle in order to reduce the number of reboots.

Step 1.1.2 - Install Serviceguard and SGeRAC on the New Database Node for Site1

Refer to Using Serviceguard Extension for RAC for instructions for installing Serviceguard and SGeRAC A.11.17 software.

Step 1.1.3 - Create the Logical Volumes on the New Database Node for Site1

To create the logical volumes to be managed by SGeRAC and used by CRS and ASM, please refer to section 4 of Sample Configurations with SGeRAC and Oracle RAC 10gR2.

For this project the database tiers for both sites were configured identically. Each used the following shared logical volume configuration:

Volume Group Size	Multiple Device Paths	Logical Volume	LV Size
/dev/vg_crs 3GB	/dev/c6t0d1 /dev/c8t0d1	/dev/vg_crs/rlv_ocr /dev/vg_crs/rlv_vote /dev/vg_crs/rlv_asm	1GB 1GB 1GB
/dev/vg_data 500GB	/dev/c6t0d2 /dev/c8t0d2	/dev/vg_data/rlv_data	500GB
/dev/vg_flash 500GB	/dev/c6t0d3 /dev/c8t0d3	/dev/vg_flash/rlv_flash	500GB

Step 1.1.4 - Configure SGeRAC on the New Database Node for Site1

For instructions on configuring the SGeRAC cluster please refer to section 4.1.2.3 Create Cluster ASCII file in Sample Configurations with SGeRAC and Oracle RAC 10gR2.

The following SGeRAC cluster details were used for a single node cluster for the Site1:

Parameter	Value
CLUSTER_NAME	cluster1
FIRST_CLUSTER_LOCK_VG	/dev/vg_crs
NODE_NAME	ha1dbh
NETWORK_INTERFACE	lan1
STATIONARY_IP	X.X.174.71
NETWORK_INTERFACE	lan2
HEARTBEAT_IP	X.X.174.146
NETWORK_INTERFACE	lan3
FIRST_CLUSTER_LOCK_PV	/dev/dsk/c6t0d1
OPS_VOLUME_GROUP	/dev/vg_crs
OPS_VOLUME_GROUP	/dev/vg_data
OPS_VOLUME_GROUP	/dev/vg_flash

Step 1.1.5 - Create SGeRAC Cluster on the New Database Node for Site1

For instructions on creating the SGeRAC cluster please refer to the section 4.1.2.4 Create Cluster in Sample Configurations with SGeRAC and Oracle RAC 10gR2.

To check the Cluster configuration:

```
# cmcheckconf -v -k -C cluster.conf
```

To create the Cluster:

```
# cmapplyconf -k -v -C cluster.conf
```

To start the Cluster:

```
# cmruncl -v
```

To verify the Cluster:

```
# cmviewcl -v
```

Step 1.1.6 - Activate Volume Groups in Shared Mode on the New Database Node for Site1

To prevent unnecessary application downtime run the vgexport command to create the volume group maps after they are marks as sharable.

In the following steps <sharable_vg> represents a single shared volume group and will need to be run once for each shared volume group.

To deactivate volume groups:

```
# vgchange -a n /dev/<sharable_vg>
```

To mark volume groups sharable:

```
# vgchange -c y -S y /dev/<sharable_vg>
```

To export the volume group maps:

```
# vgexport -p -s -m /tmp/<vg_name>.map /dev/<sharable_vg>
```

To activate sharable volume groups:

```
# vgchange -a s /dev/<sharable_vg>
```

Task 1.2 - Establish Oracle Clusterware on the New Database Node for Site1

Oracle Clusterware provides critical cluster management features including node membership and resource management. When HP SGeRAC is also installed, Oracle Clusterware defers some of its functionality, such as node membership, to Serviceguard via API calls. The following sections explain how to install Oracle Clusterware into an environment where SGeRAC is already present.

Step 1.2.1 - Install Oracle Clusterware 10gR2

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform to install Oracle Clusterware on the new database server. This should be installed in an Oracle home that is separate from the Oracle Application Database and ASM homes. Here are the parameters we specified during the installation process:

Parameter	Value
Oracle Home	/u01/app/oracle/product/10.2.0/crs
Cluster Name	ha1crs
Public Node Name	ha1dbh01.ha.us.oracle.com
Private Node Name	ha1dbh01-vip.ha.us.oracle.com
Virtual Node Name	ha1dbh01-vip.ha.us.oracle.com
Network Usage	Lan2: Private Lan1: Public Lan3: Do Not Use Lan0: Do Not Use Lan4: Do Not Use
OCR	External Redundancy /dev/vg_crs/rlv_ocr
Voting Disk	External Redundancy /dev/vg_crs/rlv_vote

Step 1.2.2 - Apply Patchset 10.2.0.2 or Later

The latest database patch set for your platform can be located on Oracle*MetaLink* and should be applied.

Task 1.3 - Establish ASM on the New Database Node for Site1

Oracle Automatic Storage Management provides volume management and file system capabilities for database files. It automatically stripes data across all disks in a disk group and optionally provides redundancy for cases when customers choose not to have it implemented in the backend storage subsystem. ASM also includes the ability to add and remove storage with no downtime as data is automatically

moved from the disks being removed to the remaining disks. The following sections explain how to install and configure ASM.

The best practice is to have two disk groups, one for database files and one for flash recovery area files. Mirrored copies of your redo logs and control files should be stored in both the flash recovery and the database file disk groups. Detailed ASM administration concepts and commands can be found in the 10gR2 Oracle Database Administrator's Guide.

Step 1.3.1 - Install 10gR2 Database for ASM

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform (see http://www.oracle.com/pls/db102/homepage) to create a clustered ASM installation on the primary and disaster site. This should be installed in an Oracle home that is separate from the Oracle Application Database and Oracle Clusterware homes. Choose the "software only" installation because we want to apply the patch before creating the instance.

Step 1.3.2 - Apply Patchset 10.2.0.2 or Later

The latest database patch set for your platform can be located on Oracle*MetaLink* and should be applied.

Step 1.3.3 - Create ASM Instance and Disk Groups

Use the DBCA utility to create an ASM instance and configure the disk groups. You will need to create data and flash recovery disk groups. This is also documented in the guide.

Disk Group	Logical Volume
DATA	/dev/vg_data/rlv_data
FLASH	/dev/vg_flash/rlv_flash

Step 1.3.4 - Remove the Listener Created by DBCA

While configuring ASM, DBCA creates a listener and registers it to CRS. This will conflict with the listener used by E-Business Suite, and needs to be deregistered with netca..

Task 1.4 - Prepare Existing Database for RAC

Step 1.4.1 - Create Additional Redo Threads

Each RAC instance requires its own redo thread. A single instance database has only one thread of redo, so additional threads must be added and enabled. Here is an example of how this can be done:

```
SQL> alter database add logfile thread 2
   group 4 ('/oradb/oradata/visdata/log2_1.dbf') size 50M,
```

```
    group 5 ('/oradb/oradata/visdata/log2_2.dbf') size 50M,
    group 6 ('/oradb/oradata/visdata/log2_3.dbf') size 50M;
SQL> alter database enable public thread 2;
```

Step 1.4.2 - Create Additional Undo Tablespaces

Each RAC instance requires its own undo tablespace. A single instance database has only one undo tablespace, so additional tablespaces must be added. Here is an example of how this can be done:

```
SQL> create undo tablespace "APPS_UNDOTS2" datafile
    '/oradb/oradata/visdata/undots201.dbf' size 1048576000,
    '/oradb/oradata/visdata/undots202.dbf' size 1048576000,
    '/oradb/oradata/visdata/undots203.dbf' size 1048576000,
    '/oradb/oradata/visdata/undots204.dbf' size 1048576000
    blocksize 8192 extent management local autoallocate;
```

Step 1.4.3 - Execute CATCLUST.SQL

The Oracle provided catclust.sql script must be executed to prepare a database for RAC operations:

```
SQL> @$ORACLE_HOME/rdbms/admin/catclust.sql
```

Task 1.5 - Prepare Existing Database for Data Guard

Step 1.5.1 - Create Password Files

Oracle Data Guard in database release 10g requires the use of a password file for communications between the primary and standby databases. The method required to implement this feature varies by platform. See Oracle*MetaLink* note 185703.1 for more information, and to get pointers for the commands needed for your platform. An example UNIX invocation:

```
$ cd $ORACLE_HOME/dbs
$ orapwd file=orapw<SID> password=<SYS's password>
```

Repeat this procedure for the current SID (VIS) and the new SIDs (VIS1 and VIS2) so that it will be copied over and used as we clone the Oracle homes.

For the password file to function correctly the following parameter setting must be configured in the database instance.

```
remote_login_passwordfile=EXCLUSIVE
```

This is the default setting in Database 10gR2 and is not overridden by the standard Apps database configuration files, so is not listed in our database configuration parameters elsewhere in this document.

Step 1.5.2 - Configure Database Instance Parameters

We plan to operate the original database as a Data Guard primary before we begin the switchover process. Thusit is necessary to apply the Data Guard parameters for primary operation straight away to keep downtime to a minimum.

Configure the original database instance to perform as a Data Guard primary so it is ready for the switchover to the new RAC instance. We set the parameters both in the <CONTEXT_NAME>_ifile.ora parameter include file and dynamically in the running instance. Here are the parameters we used for our test environment:

Data Guard Parameters for Primary Operation	log_archive_config='dg_config=(VIS, VIS_ha1)' log_archive_dest_2='SERVICE=ha1dbh01:1521/VIS1 valid_for=(online_logfiles,primary_role) db_unique_name=VIS_ha1 LGWR ASYNC=20480 OPTIONAL REOPEN=15 NET_TIMEOUT=30' log_archive_dest_state_2=defer SQL> alter system set log_archive_config='dg_config=(VIS, VIS_ha1)'; SQL> alter system set log_archive_dest_2='SERVICE=ha1dbh01:1521/VIS1 valid_for=(online_logfiles,primary_role) db_unique_name=VIS_ha1 LGWR ASYNC=20480 OPTIONAL REOPEN=15 NET_TIMEOUT=30'; SQL> alter system set log_archive_dest_state_2=defer;

Note we used EZConnect syntax to define the service for redo log transport to the temporary local standby we are creating for this exercise. This made our overall network configuration tasks simpler for this phase.

Step 1.5.3 - Enable Forced Logging

E-Business Suite sometimes uses the NOLOGGING feature, which would cause certain changes to not be populated to the standby database, and this would invalidate the standby database. To avoid this we must turn on "force logging" on the database by issuing the following command from SQL*Plus connected as sysdba:

```
SQL> alter database force logging;
```

See Oracle*MetaLink* note 216211.1 for more information.

Step 1.5.4 - Create Standby Redo Logs

Create standby redo logs on the primary database to support the standby role. The standby redo logs should be the same size or larger than the primary database online redo logs. The recommended number of standby redo logs is one more than the number of online redo logs for each thread. Because this example has two online redo logs for each thread, three standby redo logs are required for each thread. Here is a formula that can be followed:

(maximum # of logfiles +1) * maximum # of threads

To create standby redo log groups, as the ORACLE user on the production database server, connect to SQL*Plus as sysdba and issue a command like this for each standby redo log group you will create:

```
alter database add standby logfile
thread N < group N > (
'<fully qualified logfile member name>',
'<fully qualified logfile member name>')
size NNN;
```

You can see the members created by querying V$LOGFILE view:

```
select * from v$logfile;
select * from v$standby_log;
```

See the "Configure a Standby Redo Log" section in Oracle Data Guard Concepts and Administration Guide for more information.

Step 1.5.5 - Grant Access to Database Nodes

SQL*NET Access Security blocks requests coming to the database at the TNS listener level, from servers that are not specifically authorized. If the SQL*Net Access security is enabled in the existing system (enabled by default from 11i10), it is essential that all the database nodes in the current, RAC and disaster sites are given access for correct database operation. Pay particular attention to nodes that have multiple network interfaces and make sure the appropriate node aliases are included. See Managed SQL*Net Access from Hosts in document 281758.1 on OracleMetalink for instructions on how to achieve this from OAM.

Note: The standard node alias of each application tier node is automatically given access. If different network interfaces are used on the middle tiers then their network aliases must be granted access manually.

Note: It is not necessary to include the network interface used exclusively for cluster interconnect communications in your RAC configuration.

Task 1.6 - Clone Existing Database Software and Prepare New Instance

Step 1.6.1 - Prepare the Database Oracle_Home for Cloning

As the ORACLE user on the production database server, run the following commands:

$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>

```
$ perl adpreclone.pl dbTier
```

Supply the APPS password when requested.

Step 1.6.2 - Backup, Ship and Restore the Database Oracle Home

Using the appropriate operating system utilities, create a backup of the database Oracle home, ship the backups to the new database server, and then restore the Oracle home to the new RAC database node.

Note that this document assumes that the new node is not the same as the current production database node. Using the existing node would result in conflicting database listener configurations during the switchover. See the section entitled

Add the Original Node as a Secondary RAC Instance into the RAC Cluster for details on how to add the original database node to the cluster after the switchover.

You must use a different ORACLE_HOME location for the RAC instances to prevent a conflict when you reintroduce the original node to the configuration.. In our case we used this location:

```
/u01/app/oracle/visdbRAC/10.2.0
```

Step 1.6.3 - Configure the New Database Oracle Home

On the new clustered database server for site 1, execute the following process for the new Oracle home:

```
$ cd <ORACLE_HOME>/appsutil/clone/bin
$ perl adcfgclone.pl dbTechStack
```

Respond to the prompts as follows:

```
Do you want to use a virtual hostname for the target node
   (y/n) [n] ?:n

Target instance is a Real Application Cluster (RAC)
   instance (y/n) [n]:y

Current node is the first node in an N Node RAC Cluster
   (y/n)[n]:y

Number of instances in the RAC Cluster [1]:2

Target System database name [VIS]:VIS  ←same as current production

Do you want to preserve the port values from the source
   system on the target system (y/n) [y] ?:y

Provide information for the Node 1 (current node):

    Host name [ha1dbh01]:ha1dbh01

    Virtual Host name [null]:ha1dbh01-vip

    Instance number [1]:1

Provide information for the Node 2:

    Host name:ha1dbh02

    Virtual Host name:ha1dbh02-vip

    Instance number:2

Target system RDBMS ORACLE_HOME directory
   [/u01/app/oracle/visdb/10.2.0]:
   /u01/app/oracle/visdbRAC/10.2.0

Target system utl_file accessible directories list
   [/usr/tmp]:/usr/tmp

Number of DATA_TOP's on the target system [1]:1

Target system DATA_TOP 1:+DATA/vis_ha1   ←ASM Path

Do you want to preserve the Display set to ha1dbh:0.1 (y/n)
   [y] ?:n

Target system Display [ha1dbh01:0.0]:
```

The script prompts for following information:

Prompt	Response	Comment
Do you want to use a virtual hostname for the target node	n	Used for HW layer failover, not for RAC, Virtual Hostname used for RAC will be prompted later.
Number of instances in the RAC Cluster	<Total Number of RAC Nodes>	Although initially it'll setup as single instance RAC, provide total number of nodes here.
Target System database name	<DB_NAME>	Database name (i.e. PROD), not the RAC instance name (i.e. PROD1)
Target system RDBMS ORACLE_HOME directory	<NEW ORACLE_HOME >	New RAC ORACLE_HOME location
Target system DATA_TOP	<ASM PATH>	ASM Disk path where database files are located

The adcfgclone script creates a new environment file that should be sourced before using this environment:

```
$ cd <ORACLE_HOME>
$ . ./VIS1_ha1dbh01.env
```

Step 1.6.4 - Configure and Restart Standby Listener

The listener service definition generated by adcfgclone, while pointing to the standby database, does not include all the required interface names. To correct this, create a listener include file named "listener_ifile.ora" under the $TNS_ADMIN directory on the standby server, with contents similar to this example:

Listener configuration	

```
VIS1=
   ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP)(Host=
      ha1dbh01.ha.us.oracle.com)(Port=1521))
    (ADDRESS=(PROTOCOL=TCP)(Host=
      ha1dbh01-vip.ha.us.oracle.com)(Port=1521))
   )
```

Stop and restart the listener on the new standby database node:

```
$ lsnrctl stop VIS1
$ lsnrctl start VIS1
```

Step 1.6.5 - Configure Standby Database Instance Parameters

Configure the new database instance to perform as a Data Guard standby so it is ready for the switchover. The instance must also be configured for correct RMAN operation when restoring to the new ASM disk groups. Finally, if you will be

configuring parallel concurrent processing (PCP), you should set two parameters now to avoid a database bounce later.

Here are example parameters:

Data Guard Parameters for Standby Operation	db_unique_name=VIS_ha1 log_archive_config='dg_config=(VIS,VIS_ha1)' db_file_name_convert='/oradb/oradata/visdata', '+DATA/VIS_ha1/datafile', '+DATA/VIS_ha2/datafile', '+DATA/VIS_ha1/datafile' log_file_name_convert='/oradb/oradata/visdata', '+DATA/VIS_ha1/onlinelog', '+DATA/VIS_ha2/onlinelog', '+DATA/VIS_ha1/onlinelog' fal_server=ha1dbh:1521/VIS fal_client=ha1dbh01:1521/VIS1 standby_archive_dest= 'LOCATION=USE_DB_RECOVERY_FILE_DEST' standby_file_management=AUTO parallel_execution_message_size=8192
File Management Parameters (used by RMAN)	db_create_file_dest='+DATA' db_recovery_file_dest='+FLASH' db_recovery_file_dest_size=1228800M control_files= +DATA/VIS_HA1/CONTROLFILE/control01.ctl
PCP Parameters (optional)	_lm_global_posts=TRUE _immediate_commit_propagation=TRUE

Task 1.7 - Establish Standby Database

The steps to establish the standby database are as follows:

Step 1.7.1 - Backup the Production Database using RMAN

RMAN should be used to backup the current production database so that it is ready to be restored to the new standby. You must backup the database and the archived redo logs, and then the control file. Here is an example of how this can be achieved:

```
$ rman target /

RMAN> backup device type disk format
   '/NAS/oracle/rman_backups/%U' database plus archivelog;

RMAN> backup device type disk format
   '/NAS/oracle/rman_backups/%U' current controlfile for
   standby;
```

Step 1.7.2 - Ship Backup to the new Database Server

Using the appropriate system tools, ship the backup to the new database server.

Step 1.7.3 - Start one RAC instance

On the new clustered server for Site 1, start the RAC instance with the NOMOUNT option:

```
SQL> startup nomount
```

Step 1.7.4 - Restore the Database Backup to the Standby Using RMAN

The database must be restored with the "for standby" option, for example:

```
$ rman target sys/manager@ha1dbh:1521/VIS auxiliary /
RMAN> duplicate target database for standby;
```

Step 1.7.5 - Start Managed Recovery

Pay close attention to which instance the following commands are executed.

On the **standby database**, check to make sure the database is in standby mode by executing this query from SQL*Plus connected as sysdba:

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

   DATABASE_ROLE
   ----------------
   PHYSICAL STANDBY
```

On the **primary database** enable the previously deferred remote destination by executing this command from SQL*Plus connected as sysdba:

```
SQL> alter system set log_archive_dest_state_2=enable;
```

Also, update the database configuration in the <context_name>_ifile.ora database parameter file on the **primary database server**:

```
log_archive_dest_state_2=enable
```

Place the **standby database** in managed recovery by executing this command:

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

Step 1.7.6 - Verify Correct Standby Operation

Validate that the standby is correctly applying redo from the primary.

On the **primary database**, archive the current log using the following statement:

```
SQL> alter system archive log current;
```

On the **standby database**, query the gv$archived_log view to verify that the logs are received and applied:

```
SQL> select sequence#, applied,
    to_char(first_time, 'mm/dd/yy hh24:mi:ss') first,
    to_char(next_time, 'mm/dd/yy hh24:mi:ss') next,
    to_char(completion_time, 'mm/dd/yy hh24:mi:ss')
    completion
    from gv$archived_log order by first_time;
```

Task 1.8 - Clone Application Software and Configure for Switchover

We will reuse the existing production application tiers to access the single node RAC, Flashback, and ASM-enabled database just built. To avoid any risk to

production operations and get as much configuration work done as possible ahead of time, we cloned the application tier software to a new location on the application tiers.

Step 1.8.1 - Prepare the Application Tier for Cloning

For each type of application tier install at your site (e.g., Concurrent Processing, Administration, Web, Forms), log on as the APPLMGR user and run the following commands:

```
$ cd <COMMON_TOP>/admin/scripts/<CONTEXT_NAME>
$ perl adpreclone.pl appsTier
```

If you share a single common APPL_TOP or have only full installs, you need only do this on one application tier.

Step 1.8.2 - Copy the Application Tier File System

Using the appropriate tool, make a copy of the application tier files to a new directory location on the existing application tiers. We will use this new copy of the software to run the application after the switchover.

Step 1.8.3 - Configure New Application Tier File System

Run the following commands on each application server node, use the original context file or copy of it as source context file to the adclonectx.pl tool. For example, our new mount point is /u01/appltopRAC and we are pointing to the original context file at /u01/appltop.

```
$ cd /u01/appltopRAC/viscomn/clone/bin
$ perl adclonectx.pl contextfile=
    /u01/appltop/visappl/admin/<CONTEXT_NAME>.xml
```

Provide the values required for creation of the new APPL_TOP Context file:

```
Do you want to use a virtual hostname for the target node
    (y/n) [n] ?: n
Target hostname [ha1mth01]:
Do you want the inputs to be validated (y/n) [n] ?:n
Target system database SID [VIS]:
Username for the applications file system owner [applmgr]:
Group for the applications file system owner [dba]:
Target system database server node [ha1dbh]:  ha1dbh01
Target system database domain name [ha.us.oracle.com]:
Does the target system have more than one application tier
    server node (y/n) [y] ?:
Does the target system application tier utilize multiple
    domain names (y/n) [n] ?:
Target system concurrent processing node [ha1mth01]:
Target system administration node [ha1mth01]:
Target system forms server node [ha1mth01]:
```

```
Target system web server node [ha1mth01]:

Is the target system APPL_TOP divided into multiple mount
   points (y/n) [n] ?:

Target system APPL_TOP mount point [/u01/appltop/visappl]:
   /u01/appltopRAC/visappl

Target system COMMON_TOP directory [/u01/appltop/viscomn]:
   /u01/appltopRAC/viscomn

Target system 8.0.6 ORACLE_HOME directory
   [/u01/appltop/visora/8.0.6]:
   /u01/appltopRAC/visora/8.0.6

Target system iAS ORACLE_HOME directory
   [/u01/appltop/visora/iAS]: /u01/appltopRAC/visora/iAS

Do you want to preserve the Display set to ha1dbh01:0.0
   (y/n) [y] ?:n

Target system Display [ha1mth01:0.0]:

Location of the JDK on the target system [/opt/java1.4]:

Enter the port pool number [0-99]:0

New context path and file name
   [/u01/appltop/visappl/admin/VIS_ha1mth01.xml]:
      /u01/appltopRAC/visappl/admin/VIS_ha1mth01.xml
```

The scripts prompts for following information:

Prompt	Response	Comment
Do you want to use a virtual hostname for the target node	n	Used for HW layer failover, not for RAC setup unless you have such setup answer "n"
Do you want the inputs to be validated	n	The new standby database is not yet available, so the inputs cannot be validated
Target system database server node	<STANDBY DB NODE NAME>	Hostname of the new database server configured for local standby
Target system APPL_TOP mount point	<NEW APPL_TOP>	New APPL_TOP mount point that will be configured for RAC.
New context path and file name	<PATH for conext file name under new APPL_TOP>	Name and the location of the context file under new APPL_TOP that will be configured for RAC

Step 1.8.4 - Complete the New Application Tier File System Configuration

Run the following commands on each application server node, using the new context file generated in previous step. You need to first set your APPL_TOP environment variable to the new value:

```
$ export APPL_TOP=/u01/appltopRAC/visappl
    (Workaround for known issue, Bug 5520384)

$ cd /u01/appltopRAC/viscomn/clone/bin

$ perl adcfgclone.pl appsTier
    /u01/appltopRAC/visappl/admin/VIS_ha1mth01.xml
```

Note: Since the database is not available at this time, you can ignore the error related to the failed AutoConfig run.

Task 1.9 - Switchover and Enable Flashback

It is important that the flash recovery area has sufficient disk spindles and IO bandwidth to support Flashback database or there may be degradation in application performance. Oracle recommends that a representative workload is tested with flashback before rolling into production.

The steps to switch over to the standby database are as follows. The E-Business Suite application will be down during this stage, so it is recommended that the steps be carefully practiced and scripted.

Step 1.9.1 - Prepare the Standby Database

On the **standby database**, if it has been opened read only since the last time it was shut down, close it and bring it up in managed recovery mode.

Make sure managed recovery is underway, and is current.. See the section entitled Verify Correct Standby Operation for details.

Step 1.9.2 - Shutdown E-Business Suite

Ensure that the application is completely shutdown.

Step 1.9.3 - Switchover to Standby Database

On the **primary database** commit to switchover to standby:

```
SQL> alter database commit to switchover to standby;
```

Note: If the status returns SESSIONS_ACTIVE then you should issue the switchover command using the WITH SESSION SHUTDOWN clause.

```
SQL> alter database commit to switchover to standby with
    session shutdown;
```

Shutdown the **primary database** and keep it down for the time being:

```
SQL> shutdown immediate
```

Stop the **primary** database listener:

```
$ lsnrctl stop VIS
```

On the **standby database**, verify that it is ready to be converted to the new primary:

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

    SWITCHOVER_STATUS
```

```
-----------------
TO PRIMARY
```

On the **standby** database, execute the following command to convert it to be the new primary:

```
SQL> alter database commit to switchover to primary;
```

Step 1.9.4 - Enable Flashback

Enable the flashback feature on the new production database:

```
SQL> alter database flashback on;
```

Step 1.9.5 - Open the Database

Beginning with Oracle Database 10g Release 2, you can open the new primary database from the mount state if the standby database has not been opened read-only since the last time the database was started. If the database has been opened read-only, it will need to be restarted.

```
SQL> alter database open;
```

Step 1.9.6 - Remove the Old Application Topology

Connect to the new production database using SQL*Plus as user APPS and execute the following commands:

```
SQL> exec FND_NET_SERVICES.remove_system('VIS');
     commit;
```

Step 1.9.7 - Configure Database Tier

Run the following command on the new database server to complete the configuration of the new Oracle home for use by E-Business Suite:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 1.9.8 - Restart Listeners

Start the correct listener on the new production database node:

```
$ lsnrctl stop VIS1
$ lsnrctl start LISTENER_ha1dbh01
```

Note: AutoConfing utility configure the sql*net listener using the <SID> as the listener name when database is not available, and LISTENER_<hostname> when database is available, that's why we need to switch between listeners as needed.

Step 1.9.9 - Configure Application Tiers

Run AutoConfig using the new APPL_TOP on all the application tier nodes. Note, these can be run in parallel.

On all application tier hosts running as applmgr:

```
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 1.9.10 - Startup E-Business Suite

Startup E-Business Suite on all application tier hosts as applmgr

```
$ cd <APPL_TOP>
$ . ./APPS<CONTEXT_NAME>.env
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adstrtal.sh
```

Online users can gain access at this time.

PHASE 2 - ADD THE ORIGINAL NODE AS A SECONDARY RAC INSTANCE INTO THE RAC CLUSTER

Once the application has stabilized on the new database server it is possible to take the original database server out of service. We then install the cluster related hardware, required patches, HP SGeRAC software, Oracle Clusterware, ASM and RAC on the original server and add it to the cluster creating a two node RAC cluster with ASM. The tasks to achieve this are documented below:

Task 2.1 - Original Database Node Joins the SGeRAC Cluster

Step 2.1.1 - Change hostname to ha1dbh02

Change the hostname HA1DBH to HA1DBH02 to distinguish the before and after states for this node. Edit the /etc/rc.config.d/netconf file and modify the HOSTNAME="<original hostname>" statement to reflect the new hostname.

Step 2.1.2 - Install cluster hardware for the Original Database node

Shutdown original database node and install and configure additional cluster hardware such as NICs and FC HBAs.

Step 2.1.3 - Apply the SGeRAC patch bundle to the Original Database Node

Refer to Serviceguard Version A.11.17 Release Notes for a list of required patches.

Step 2.1.4 - Install SGeRAC on the Original Database Node

Refer to Using Serviceguard Extension for RAC for instructions for installing Serviceguard and SGeRAC A.11.17 software.

Step 2.1.5 - Edit the Cluster Configuration File

Add the second node information to the cluster.conf file located on the first node:/etc/cmcluster/cluster.conf. The following SGeRAC cluster details were added:

Parameter	Value
NODE_NAME	ha1dbh02
NETWORK_INTERFACE	lan1
STATIONARY_IP	X.X.174.71
NETWORK_INTERFACE	lan2
HEARTBEAT_IP	X.X.174.146
NETWORK_INTERFACE	lan3
FIRST_CLUSTER_LOCK_PV	/dev/dsk/c6t0d1

Step 2.1.6 - Add the second node (original DB node) to the SGeRAC Cluster

To check the new Cluster configuration:

```
# cmcheckconf -v -k -C cluster.conf
```

To apply the changes to the Cluster:

```
# cmapplyconf -k -v -C cluster.conf
```

To add the new node to the Cluster:

```
# cmrunnode –v ha1dbh02
```

To verify the Cluster:

```
# cmviewcl –v
```

Step 2.1.7 - Import the maps and activage the Shared Volume Group

For instructions on exporting and importing shared volume groups for SGeRAC please refer to section 4.1.2.2 of Sample Configurations with SGeRAC and Oracle RAC 10gR2

In the following steps <sharable_vg> represents a single shared volume group and will need to be run once for each shared volume group to be activated.

To activate sharable volume groups:

```
# vgchange -a s /dev/<sharable_vg>
```

Task 2.2 - Establish Oracle Clusterware and ASM

Follow the instructions in Oracle® Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide, Section: 10 Adding and Deleting Nodes and Instances on UNIX-Based Systems to:

1. Establish Oracle Clusterware and add this node to the cluster at the Oracle Clusterware layer.
2. Establish an ASM Oracle Home and add a new ASM instance. In this context, you are adding a RAC node and instance to ASM.
3. Make sure Oracle Clusterware and ASM are up and running.

Note, the addNode feature will establish the software and instances on the new node. There is no need to separately install the software.

Parameter	Value
Oracle Home	/u01/app/oracle/product/10.2.0/crs
Cluster Name	ha1crs
Public Node Name	ha1dbh02.ha.us.oracle.com
Private Node Name	ha1dbh02-ci.ha.us.oracle.com
Virtual Node Name	ha1dbh02-vip.ha.us.oracle.com

Parameter	Value
Network Usage	Lan2: Private Lan1: Public Lan3: Do Not Use Lan0: Do Not Use Lan4: Do Not Use
OCR	External Redundancy /dev/vg_crs/rlv_ocr
Voting Disk	External Redundancy /dev/vg_crs/rlv_vote

Task 2.3 - Clone Database Software and Prepare New Instance

Step 2.3.1 - Prepare the RAC Oracle_Home for Cloning

Log on to the RAC database server as the ORACLE user and run the following commands:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ perl adpreclone.pl dbTier
```

Supply the APPS password when requested.

Step 2.3.2 - Backup, Ship and Restore the Database Oracle Home

Using the appropriate operating system utilities, create a backup of the database Oracle home and restore it to the new database server restore the Oracle home to the original database node.

Step 2.3.3 - Configure the Database Oracle Home

On the original database node, execute the following process for the new Oracle home:

```
$ cd <ORACLE_HOME>/appsutil/clone/bin
$ perl adcfgclone.pl dbTechStack
```

Respond to the prompts as follows:

```
Do you want to use a virtual hostname for the target node
   (y/n) [n] ?: n

Target instance is a Real Application Cluster (RAC)
   instance (y/n) [y]: y

Current node is the first node in an N Node RAC Cluster
   (y/n)[n]: n

Please provide the details to connect to one of live RAC
   nodes

Host name of the live RAC node: ha1dbh01

Domain name of the live RAC node: ha.us.oracle.com

Database SID of the live RAC node: VIS1
```

```
Listener port number of the live RAC node: 1521
Current host is not registered, do you want to add this
   node to the RAC cluster(y/n)[y]: y
Number of new instances to be added in the RAC Cluster [1]:
Provide information for the Node 2 (current node):
    Host name:ha1dbh02
    Instance number:2
    Private interconnect name:ha1dbh02-ci
Current Node:
    Host Name       : ha1dbh02
    SID             : VIS2
    Instance Name   : VIS2
    Instance Number : 2
    Instance Thread : 2
    Undo Table Space: APPS_UNDOTS2
    Listener Port   : 1521
Target system utl_file accessible directories list
   [/usr/tmp]:
Number of DATA_TOP's on the target system [2]:1
Target system DATA_TOP 1:+DATA/vis_ha1
Do you want to preserve the Display set to ha1dbh01:0.1
   (y/n) [y] ?:
```

The script prompts for following information:

Prompt	Response	Comment
Do you want to use a virtual hostname for the target node	n	Used for HW layer failover, not for RAC, Virtual Hostname used for RAC will be prompted later.
Current node is the first node in an N Node RAC Cluster	n	Current node will be added as a new node.
Please provide the details to connect to one of live RAC nodes	Information about live RAC node	Connect information for one of the existing RAC node where database is Up and Running.
Provide information for the Node 2	Information about current node	Information about new node
Target system DATA_TOP	<ASM PATH>	ASM Disk path where database files are located

The adcfgclone script creates a new environment file, which should be sourced before using this environment::

```
$ cd <ORACLE_HOME>
$ . ./VIS2_ha1dbh02.env
```

Step 2.3.4 - Configure Database Instance Parameters

On the original database server in the new database Oracle home, add the following parameters to your database configuration via the "include" file found at $ORACLE_HOME/dbs/<CONTEXT_NAME>_ifile.ora. .

Note: Keeping your changes in a separate "include" file gives you the freedom to recreate the base database parameter files at any time using AutoConfig, without erasing your customizations.

Note: We continue to refer to the single instance node with the data guard parameters because these parameters must match across RAC instances.

Data Guard Parameters for Standby Operation	db_unique_name=VIS_ha1 log_archive_config='dg_config=(VIS, VIS_ha1)' db_file_name_convert='/oradb/oradata/visdata', '+DATA/VIS_ha1/datafile', '+DATA/VIS_ha2/datafile', '+DATA/VIS_ha1/datafile' log_file_name_convert='/oradb/oradata/visdata', '+DATA/VIS_ha1/onlinelog', '+DATA/VIS_ha2/onlinelog', '+DATA/VIS_ha1/onlinelog' fal_server=ha1dbh:1521/VIS fal_client=ha1dbh01:1521/VIS1 standby_archive_dest= 'LOCATION=USE_DB_RECOVERY_FILE_DEST' standby_file_management=AUTO parallel_execution_message_size=8192
File Management Parameters (used by RMAN)	db_create_file_dest='+DATA' db_recovery_file_dest='+FLASH' db_recovery_file_dest_size=1228800M control_files= +DATA/VIS_HA1/CONTROLFILE/control01.ctl
PCP Parameters (optional)	_lm_global_posts=TRUE _immediate_commit_propagation=TRUE

Step 2.3.5 - Start the New Instance

Start the new instance on the original database server and make sure it can join the cluster.

```
SQL> startup
```

Step 2.3.6 - Run Autoconfig

The adcfgclone script does not request "virtual hostname." Edit the context variable "s_virtual_hostname" in the context file found at $ORACLE_HOME/appsutil/<CONTEXT_NAME>.xml, giving the value for your virtual host. In our tests, we changed the host name "ha1dbh01" to "ha1dbh01-vip" in this variable.

Then run the following command to configure the new node for use by E-Business Suite:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 2.3.7 - Restart Listener

For the new instance on the original database server, start the correct listener :

```
$ lsnrctl stop VIS2
$ lsnrctl start LISTENER_ha1dbh02
```

Step 2.3.8 - Run AutoConfig on the First Database Node

Run the following command on the first database node to generate a TNS configuration that includes the RAC instance just added. This can be done while the system is up:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Task 2.4 - Configure Application Tiers for RAC

At this point, the original database server is now running the second RAC instance, but the application tiers are still only accessing the first RAC instance on the new database server. To complete the picture, AutoConfig must be run once more on the application tiers, and the Application processes on the application tiers bounced. To reduce the effect of this outage, you might control the bounce of the application tiers by "starving" them one by one via your hardware load balancer and bouncing each one when idle.

Step 2.4.1 - Modify Context Variables for Load Balancing

The adcfgclone script does not set variables to use the load balanced services. To adjust these parameters, run the Context Editor through Oracle Applications Manager to set the value of "Tools OH TWO_TASK" (s_tools_two_task), "iAS OH TWO_TASK" (s_weboh_twotask), and "Apps JDBC Connect Alias".

To load balance the forms based applications database connections, set the value of "Tools OH TWO_TASK" to point to the <database_name>_806_balance alias generated in the tnsnames.ora file.

To load balance the self-service applications database connections, set the value of iAS OH TWO_TASK" to point to the <database_name>_balance alias generated in the tnsnames.ora file.

If you are load balancing for Parallel Concurrent Processing, set the value of "Concurrent Manager TWO_TASK" to point to the <database_name>_806_balance alias generated in the tnsnames.ora file.

Step 2.4.2 - Run AutoConfig

Run AutoConfig on all the application tiers:

```
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 2.4.3 - Restart E-Business Suite

Source the new environment file and restart the applications processes on all the application tier nodes.

```
$ cd <APPL_TOP>
$ . ./APPS<CONTEXT_NAME>.env
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adstpall.sh
$ ./adstrtal.sh
```

Step 2.4.4 - Configure Parallel Concurrent Processing (PCP)

If there is more than one Concurrent Processing Node and you would like to take advantage of PCP, refer to the "Configure Parallel Concurrent Processing" section of Meta*Link* Note 362135.1, to complete PCP configuration.

Task 2.5 - Update Clusterware Configuration

If you want to use srvctl to control the resources you need to add the new resources to CRS.

Step 2.5.1 - Add Database to the Clusterware Configuration

Add the database to CRS using the following command:

```
srvctl add database -d VIS -o <ORACLE_HOME>
```

Step 2.5.2 - Add Instances to the Clusterware Configuration

Add instances to CRS using the following commands:

```
srvctl add instance -d VIS -i VIS1 -n ha1dbh01
srvctl add instance -d VIS -i VIS2 -n ha1dbh02
srvctl setenv instance -d VIS -i VIS1
    -t TNS_ADMIN=<ORACLE_HOME>/network/admin/VIS1_ha1dbh01
srvctl setenv instance -d VIS -i VIS2
    -t TNS_ADMIN=<ORACLE_HOME>/network/admin/VIS2_ha1dbh02
srvctl modify instance -d VIS -i VIS1 -s +ASM1
```

```
srvctl modify instance -d VIS -i VIS2 -s +ASM2
```

Step 2.5.3 - Add Listeners to the Clusterware Configuration

Use NETCA to add listeners to CRS by following the steps listed below:

- Edit $ORACLE_HOME/bin/racgwrap, and add commands similar to this:

  ```
  TNS_ADMIN=<ORACLE_HOME>/network/admin/<CONTEXT_NAME>
  export TNS_ADMIN
  ```

- Make sure the listener is running so the AutoConfig generated listener definition is used all the time.
- Run netca and choose "Cluster Configuration". Choose only local node.
- Add (or reconfigure) the listener named "LISTENER" listening on the database port, ignore the errors "Listener is already running" or "Port already in use".
- Run AutoConfig to overwrite the listener.ora generated by netca.

Repeat these steps on all nodes, since each has different $TNS_ADMIN settings.

Step 2.5.4 - Verify CRS Setup

Verify the resources were added correctly using following command:

```
$ $CRS_HOME/bin/crs_stat
```

Task 2.6 - Create the SGeRAC packages for Oracle Clusterware

The SGeRAC Integration Framework/Toolkit is free and fully supported by HP. The Toolkit provides templates and scripts for setting up Multi-Node Packages (MNP) for SGeRAC. Consult the README file included with the Toolkit for instructions on how to setup and configure SGeRAC MNPs.

Step 2.6.1 - Download the SGeRAC Toolkit from the HP web site and install it on Node1.

Obtain the SGeRAC Toolkit from your HP representative and deposit the software on Node1:/etc/cmcluster/toolkit

Step 2.6.2 - Modify the MNP template and scripts

Edit the fwent.sh script to reflect the correct ORA_CRS_HOME.

Parameter	Value
ORA_CRS_HOME	/u01/app/oracle/product/10.2.0

Details used in creating the MNP Package control script for CRS

Parameter	Value

Parameter	Value
Set Path	export ORACLE_HOME=/u01/app/oracle/product/10.2.0 PATH=$SGSBIN:/urs/bin:/usr/sbin:/etc:/bin:$ORACLE_HOME
VGCHANGE	"vgchange -a s"
VG[0]	vg_crs
VG[1]	vg_data
VG[2]	vg_flash
Service_Name[0]	ORACLE_CRS
Service_CMD[0]	"/etc/cmcluster/crsp/fwent.sh check"
Service_Restart[0]	"-r 2"

Function customer_defined_run_cmd	{
	# Start Oracle Clusterware
	/etc/cmcluster/crsp/fwent.sh start
	Test_return 51
	}

Function customer_defined_halt_cmd	{
	# Stop Oracle Clusterware
	/etc/cmcluster/crsp/fwent.sh stop
	Test_return 52
	}

Details used in creating the MNP Package configuration file for CRS

Parameter	Value	Description
Package_Name	crs_pkg	
Package_Type	Multi_Node	
#Failover_Policy	X	Comment out
#Failback_Policy	X	Comment out
Node_Name	*	All Nodes in the Cluster
Auto_Run	YES	Cluster start = Package start
Local_LAN_ Failover_Allowed	YES	Support for Standby Heartbeat

Parameter	Value	Description
Run_Script	/etc/cmcluster/crsp / crs_pkg.sh	
Run_Script_Timeout	No_Timeout	
Halt_Script	/etc/cmcluster/crsp / crs_pkg.sh	
Halt_Script_Timeout	No_Timeout	
Service_Name	ORACLE_CRS	
Service_Fail_Fast_Enable	NO	
Service_Halt_Timeout	60	# of seconds before SIGKILL is sent

Step 2.6.3 - Modify the auto start feature of CRS on Node1

Disable the automatic start feature for CRS on Node1. This will allow SGeRAC to bring the shared logical volumes online prior to starting CRS and take them offline after shutting down CRS.

Verify that Node2 is up and running

```
# <CRS_HOME>/bin/crs_stat |grep STATE
```

Stopping the CRS will halt the RAC instance on Node1

```
# <CRS_HOME>/bin/crsctl stop crs
# <CRS_HOME>/bin/crsctl disable crs
```

Step 2.6.4 - Deactivate the Shared Volume Groups on Node1

In the following steps <sharable_vg> represents a single shared volume group and will need to be run once for each shared volume group to be deactivated.

To deactivate sharable volume groups:

```
# vgchange -a n /dev/<sharable_vg>
```

Step 2.6.5 - Verify and startup the SGeRAC package on Node1

Starting the CRS MNP will activate the shared logical volumes and startup Oracle's Clusterware service.

Verify the CRS MNP configuration

```
# cmcheckconf -P crs_pkg.conf
```

Apply changes

```
# cmapplyconf -P crs_pkg.conf
```

Startup CRS MNP on Node1

```
# cmrunpkg -n ha1dbh01 crs_pkg
```

Verify that the new SGeRAC package is running

```
# cmveiwcl -v
```

Step 2.6.6 - Copy the Toolkit including the newly modified template and scripts to Node 2

Note any change to the Toolkit's template or scripts will need to be replicated on all nodes participating in the SGeRAC package. When copying the Toolkit make sure that the directory structures are duplicated.

Step 2.6.7 - Stop the Oracle instance on Node 2

Disable the automatic start feature for CRS on Node2. This will allow SGeRAC to bring the shared logical volumes online prior to starting CRS and take them offline after shutting down CRS.

Verify that the RAC instance on Node1 is up and running.

```
# <CRS_HOME>/bin/crs_stat |grep STATE
```

Stopping the CRS will halt the RAC instance on Node2:

```
# <CRS_HOME>/bin/crsctl stop crs
# <CRS_HOME>/bin/crsctl disable crs
```

Step 2.6.8 - Deactivate the Shared Volume Groups on Node2

In the following steps <sharable_vg> represents a single shared volume group and will need to be run once for each shared volume group to be deactivated.

To deactivate sharable volume groups:

```
# vgchange -a n /dev/<sharable_vg>
```

Step 2.6.9 - Startup the SGeRAC package on Node2

To startup CRS MNP:

```
# cmrunpkg -n ha1dbh02 crs_pkg
```

To verify the cluster and packages are running:

```
# cmviewcl -v
```

PHASE 3 - ESTABLISH DISASTER RECOVERY SITE UTILIZING ORACLE DATA GUARD

To complete the MAA it is necessary to establish a disaster recovery site in case the primary site is lost. To do this we establish a two node 10gR2 RAC database cluster on database servers at the DR site utilizing Oracle Clusterware and ASM. We then backup the production database and establish a Data Guard physical standby on the new server. We also clone the E-Business Suite application tier software from the primary site to the disaster site. The standby system will constantly apply redo from the primary site so that it stays up-to-date and will be ready to take over in the event of an emergency. We also include the steps for switchover and failover should they prove necessary. The details tasks required to establish the disaster site are as follows:

Task 3.1 - Establish HP Serviceguard and Shared Logical Volumes

If not completed earlier, please refer to the earlier section entitled Establish HP Serviceguard and Shared Logical Volumes for details on how to install and configure HP Serviceguard and Shared Logical Volumes. The following SGeRAC cluster details were used for the cluster for Disaster Site:

Parameter	Value
CLUSTER_NAME	cluster2
FIRST_CLUSTER_LOCK_VG	/dev/vg_crs
NODE_NAME	ha2dbh01
NETWORK_INTERFACE	lan1
STATIONARY_IP	X.X.175.70
NETWORK_INTERFACE	lan2
HEARTBEAT_IP	X.X.175.145
NETWORK_INTERFACE	lan3
FIRST_CLUSTER_LOCK_PV	/dev/dsk/c6t0d1
NODE_NAME	ha2dbh02
NETWORK_INTERFACE	lan1
STATIONARY_IP	X.X.175.71
NETWORK_INTERFACE	lan2
HEARTBEAT_IP	X.X.175.146
NETWORK_INTERFACE	lan3
FIRST_CLUSTER_LOCK_PV	/dev/dsk/c6t0d1
OPS_VOLUME_GROUP	/dev/vg_crs
OPS_VOLUME_GROUP	/dev/vg_data
OPS_VOLUME_GROUP	/dev/vg_flash

Task 3.2 - Establish Oracle Clusterware

Once HP Serviceguard is in place we can install Oracle Clusterware on the DR database nodes and start Oracle Clusterware.

Step 3.2.1 - Install Oracle Clusterware 10gR2

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform (see http://www.oracle.com/pls/db102/homepage) to install Oracle Clusterware on both/all database servers at the disaster site. This should be installed in an Oracle home that is separate from the Oracle Application Database and ASM homes. For example, we supplied the following parameters during our tests:

Parameter	Value
Oracle Home	/u01/app/oracle/product/10.2.0/crs
Cluster Name	ha2crs
Public Node Name	ha2dbh01.ha.us.oracle.com
Private Node Name	ha2dbh02-ci.ha.us.oracle.com
Virtual Node Name	ha2dbh01-vip.ha.us.oracle.com
Public Node Name	ha2dbh02.ha.us.oracle.com
Private Node Name	ha2dbh02-ci.ha.us.oracle.com
Virtual Node Name	ha2dbh02-vip.ha.us.oracle.com
Network Usage	Lan2: Private Lan1: Public Lan3: Do Not Use Lan0: Do Not Use Lan4: Do Not Use
OCR	External Redundancy /dev/vg_crs/rlv_ocr
Voting Disk	External Redundancy /dev/vg_crs/rlv_vote

Step 3.2.2 - Apply Patchset 10.2.0.2 or Later

The latest database patch set for your platform can be located on Oracle*MetaLink* and should be applied.

Step 3.2.3 - Create SGeRAC package for Oracle Clusterware

See the previous section entitled Create the SGeRAC packages for Oracle Clusterware for details on how to create the SGeRAC packages for Oracle Clusterware.

Task 3.3 - Establish ASM

In this task we establish an ASM installation and create the disk groups. In our tests we created one disk group for data and one disk group for flash recovery.

Step 3.3.1 - Install 10gR2 Database for ASM

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform (see http://www.oracle.com/pls/db102/homepage) to create a clustered ASM installation on the database servers at the disaster site. This should be installed in an Oracle home that is separate from the Oracle Application Database and Oracle Clusterware homes. Choose the "software only" installation because we want to apply the patch before creating the instance.

Step 3.3.2 - Apply Patchset 10.2.0.2 or Later

The latest database patch set for your platform can be located on Oracle*MetaLink* and should be applied.

Step 3.3.3 - Create ASM Instance and Disk Groups

Use the DBCA utility to create an ASM instance and configure the disk groups. You will need data and flash recovery disk groups.

Disk Group	Logical Volume
DATA	/dev/vg_data/rlv_data
FLASH	/dev/vg_flash/rlv_flash

Step 3.3.4 - Remove the Listeners Created by DBCA.

While configuring ASM, DBCA creates listeners that conflict with those that will be created by the E-Business Suite, and registers them to CRS. From either DR site node, deregister the ASM listeners from CRS with netca.

Task 3.4 - Prepare Existing Database for Data Guard

Step 3.4.1 - Create Password Files

Make sure a password file is available for each of the DR database instances. See the earlier section Create Password Files for details.

Step 3.4.2 - Configure SQL*Net for Communication between Sites

You need to define SQL*net services that specify load balance no, failover yes, to be used for communication between the production and disaster recovery sites. To avoid overwriting them with AutoConfig, create them in the include file "<CONTEXT_NAME>_ifile.ora" under the $TNS_ADMIN directory on each database server node. Here is an example of the configuration that was used in our test environment:

SQL*net Configuration	

```
VIS_HA1=
   (DESCRIPTION=
     (LOAD_BALANCE=NO)
     (FAILOVER=YES)
     (ADDRESS_LIST=
       (ADDRESS=
         (PROTOCOL=tcp)
         (HOST=ha1dbh01.ha.us.oracle.com)
         (PORT=1521))
       (ADDRESS=
         (PROTOCOL=tcp)
         (HOST=ha1dbh02.ha.us.oracle.com)
         (PORT=1521))
      )
      (CONNECT_DATA=(SERVICE_NAME=VIS))
   )
VIS_HA2=
   (DESCRIPTION=
     (LOAD_BALANCE=NO)
     (FAILOVER=YES)
     (ADDRESS_LIST=
       (ADDRESS=
         (PROTOCOL=tcp)
         (HOST=ha2dbh01.ha.us.oracle.com)
         (PORT=1521))
       (ADDRESS=
         (PROTOCOL=tcp)
         (HOST=ha2dbh02.ha.us.oracle.com)
         (PORT=1521))
      )
      (CONNECT_DATA=(SERVICE_NAME=VIS))
   )
```

We configured the remote archiving to take the advantage of sql*net layer failover in RAC environment, for some reason is first node is not available archiving will be done to second node and as long as archive destination is in shared disks (+FLASH) recovery will find the log files regardless of which node performs the log application,

Step 3.4.3 - Configure Database Instance Parameters

It is necessary to apply the Data Guard parameters for primary operation immediately so that standby operation can be started without a database restart.

Configure the existing database instances to perform as a Data Guard primary and standby, for RMAN, and, if desired, for Parallel Concurrent Processing (PCP). These parameters should be placed in the database parameter include file, generally named $ORACLE_HOME/dbs/<CONTEXT_NAME>_ifile.ora (the file is called via the generated database parameter file <CONTEXT_NAME>_APPS_BASE.ora).

As this will be a permanent Data Guard configuration, we have used service names for database connections, not EZConnect. We have, however, retained the file name conversion entries for the original single-instance database here, as this parameter must match across RAC instances and cannot be changed dynamically. Changing it across the production cluster to remove the original reference would require a complete database shutdown, thus could be scheduled for a maintenance window at the primary site.

Data Guard Parameters used when this database is Primary	SQL> alter system set log_archive_config= 'dg_config=(VIS_ha1,VIS_ha2)'; SQL> alter system set log_archive_dest_2='SERVICE=VIS_ha2 valid_for=(online_logfiles,primary_role) db_unique_name=VIS_ha2 LGWR ASYNC=20480 OPTIONAL REOPEN=15 NET_TIMEOUT=30'; SQL> alter system set log_archive_dest_state_2=defer; db_unique_name=VIS_ha1 log_archive_config='dg_config=(VIS_ha1,VIS_ha2)' log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST MANDATORY' log_archive_dest_2='SERVICE=VIS_ha2 valid_for=(online_logfiles,primary_role) db_unique_name=VIS_ha2 LGWR ASYNC=20480 OPTIONAL REOPEN=15 NET_TIMEOUT=30' log_archive_dest_state_2 = defer
Data Guard Parameters used when this database is Standby	db_file_name_convert='/oradb/oradata/visdata', '+DATA/VIS_ha1/datafile', '+DATA/VIS_ha2/datafile', '+DATA/VIS_ha1/datafile' log_file_name_convert='/oradb/oradata/visdata', '+DATA/VIS_ha1/onlinelog', '+DATA/VIS_ha2/onlinelog', '+DATA/VIS_ha1/onlinelog' fal_server='VIS_ha2' fal_client='VIS_ha1' standby_archive_dest= 'LOCATION=USE_DB_RECOVERY_FILE_DEST' standby_file_management=AUTO parallel_execution_message_size=8192
File Management Parameters (used by RMAN)	db_create_file_dest='+DATA' db_recovery_file_dest='+FLASH' db_recovery_file_dest_size=1228800M control_files = +DATA/VIS_HA1/CONTROLFILE/control01.ctl
PCP Parameters (Optional)	_lm_global_posts=TRUE _immediate_commit_propagation=TRUE

Step 3.4.4 - Grant Access to Database Nodes

If SQL*Net Access security is enabled in the existing system (enabled by default from 11i10), you need to give the database nodes in the production and disaster sites access for standby communications. See Managed SQL*Net Access from Hosts in document 281758.1 on OracleMetalink for instructions on how to achieve this from OAM.

Step 3.4.5 - Add Standby Redo Logs

If you haven't already created standby redo logs, see section Create Standby Redo Logs earlier in this document for more information.

Task 3.5 - Clone Database Software and Prepare New Instances

Step 3.5.1 - Prepare the Database Oracle Home for Cloning

As the ORACLE user on one of the production database servers, run the following commands:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ perl adpreclone.pl dbTier
```

Supply the APPS password when requested.

Step 3.5.2 - Backup, Ship and Restore the Database Oracle Home

Using the appropriate operating system utilities, create a backup of the database Oracle home, ship the backups to the new database server, then restore the Oracle home to the first RAC database node. It is best to use the same directory structures on the DR site.

Step 3.5.3 - Configure the New Database Oracle Homes

Execute the following process for each new Oracle home on the DR site:

```
$ cd <ORACLE_HOME>/appsutil/clone/bin
$ perl adcfgclone.pl dbTechStack
```

Respond to the prompts as follows:

```
Do you want to use a virtual hostname for the target node
   (y/n) [n] ?:n
Target instance is a Real Application Cluster (RAC)
   instance (y/n) [n]:y
Current node is the first node in an N Node RAC Cluster
   (y/n)[n]:y .
Number of instances in the RAC Cluster [1]:2
Target System database name [VIS]:VIS
Do you want to preserve the port values from the source
   system on the target system (y/n) [y] ?:y
Provide information for the Node 1 (current node):
    Host name [ha1dbh01]:ha2dbh01
    Virtual Host name [null]:ha2dbh01-vip
    Instance number [1]:1
Private interconnect name [ha1dbh01-ci]: ha2dbh01-ci
Provide information for the Node 2:
    Host name [ha1dbh01]:ha2dbh02
    Virtual Host name [null]:ha2dbh02-vip
    Instance number [1]:2   ← Instance number for the other node
```

```
    Private interconnect name [ha1dbh02-ci]: ha2dbh02-ci
Target system RDBMS ORACLE_HOME directory
   [/u01/app/oracle/visdbRAC/10.2.0]:
Target system utl_file accessible directories list
   [/usr/tmp]:/usr/tmp
Number of DATA_TOP's on the target system [2]:1
Target system DATA_TOP 1:+DATA/vis_ha2   ← The same on all nodes
Do you want to preserve the Display set to ha1dbh02:0.1
   (y/n)  [y]  ?:y
```

The scripts prompts for following information:

Prompt	Response	Comment
Do you want to use a virtual hostname for the target node	n	Used for HW layer failover, not for RAC, Virtual Hostname used for RAC will be prompted later.
Current node is the first node in an N Node RAC Cluster	y	Always answer "y" on the second (and others if more) node since there is no live RAC node to retrieve topology.
Target System database name	<DB_NAME>	Database name (i.e. PROD), not the RAC instance name (i.e. PROD1)
Provide information for the Node 1 (current node)	Information about current node,	Provide the information for the node that script executed
Provide information for the Node 2	Information about other node	Information about other node
Target system DATA_TOP	<ASM PATH>	ASM Disk path where database files are located

The adcfgclone script creates a new context file and a new environment file. The environment file should be sourced before executing commands for this instance:

```
$ cd <ORACLE_HOME>
$ . ./<CONTEXT_NAME>.env
```

Step 3.5.4 - Additional Configuration for the Second Node

The second node cannot retrieve topology information from the database since it is not running yet, so the following manual changes are needed:

Edit the context file (found at $ORACLE_HOME/appsutil/<CONTEXT_NAME>.xml) and modify following variables:

```
instance_number, specifying the instance number
instance_thread, specifying the redo log thread
undotablespace, specifying the undo tablespace name
```

Remove the database initialization files so that AutoConfig can recreate them:

```
$ cd $ORACLE_HOME/dbs
$ rm init<SID>.ora <SID>_APPS_BASE.ora
```

Run AutoConfig:

```
$ cd $ORACLE_HOME/appsutil/bin
$ ./adconfig.sh  contextfile =
    <ORACLE_HOME>/appsutil/<CONTEXT_NAME> run=INSTE8
```

Step 3.5.5 - Configure SQL*Net for Communication between Sites

When adjusting parameters on the production site for DR setup, you created an include file holding TNS service definitions with failover, not load balancing, across RAC instances for each node. Copy that file to the $TNS_ADMIN directory for each instance at your DR site and name it <CONTEXT_NAME>_ifile.ora.

Step 3.5.6 - Configure and Restart Listeners

The listener service definition generated by AutoConfig, while pointing to the standby database, does not include all the required interface names. To correct this, create a listener include file named "listener_ifile.ora" under the $TNS_ADMIN directory on each standby database server, with contents similar to this example:

Listener configuration for the first node	VIS1= ADDRESS_LIST= (ADDRESS=(PROTOCOL=TCP)(Host= ha2dbh01.ha.us.oracle.com)(Port=1521)) (ADDRESS=(PROTOCOL=TCP)(Host= ha2dbh01-vip.ha.us.oracle.com)(Port=1521)))
Listener configuration for the second node	VIS2= ADDRESS_LIST= (ADDRESS=(PROTOCOL=TCP)(Host= ha2dbh02.ha.us.oracle.com)(Port=1521)) (ADDRESS=(PROTOCOL=TCP)(Host= ha2dbh02-vip.ha.us.oracle.com)(Port=1521)))

Stop/Start Listener on each node

```
$ lsnrctl stop <SID>
$ lsnrctl start <SID>
```

Step 3.5.7 - Configure Database Instance Parameters

Configure the new database instance to perform as a Data Guard standby and primary so it is ready for the switchover and switchback respectively. The instance must also be configured for correct RMAN operation when restoring to the new ASM disk groups. Finally, if you will be implementing Parallel Concurrent Processing, you may wish to add the database parameters for using queues for Transaction Manager communications at this time.

This table holds sample parameters:

Data Guard Parameters for Primary Operation	db_unique_name=VIS_ha2 log_archive_config='dg_config=(VIS_ha1,VIS_ha2)' log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST MANDATORY' log_archive_dest_2='SERVICE=VIS_ha1 valid_for=(online_logfiles,primary_role) db_unique_name=VIS_ha1 LGWR ASYNC=20480 OPTIONAL REOPEN=15 NET_TIMEOUT=30' log_archive_dest_state_2 = defer
Data Guard Parameters for Standby Operation	db_file_name_convert='+DATA/VIS_ha1/datafile', '+DATA/VIS_ha2/datafile' log_file_name_convert='+DATA/VIS_ha1/onlinelog', '+DATA/VIS_ha2/onlinelog' fal_server='VIS_ha1' fal_client='VIS_ha2' standby_archive_dest= 'LOCATION=USE_DB_RECOVERY_FILE_DEST' standby_file_management = AUTO parallel_execution_message_size=8192
File Management Parameters (used by RMAN)	db_create_file_dest='+DATA' db_recovery_file_dest='+FLASH' db_recovery_file_dest_size=1228800M control_files = +DATA/VIS_HA2/CONTROLFILE/control01.ctl
PCP Parameters	_lm_global_posts=TRUE _immediate_commit_propagation=TRUE

Task 3.6 - Establish Standby Database

The steps to establish the standby database are as follows:

Step 3.6.1 - Backup the Database using RMAN

RMAN should be used to backup the current production database so that it is ready to be restored to the new standby. You must backup the database and the archived redo logs, and then the control file. Here is an example of how this can be achieved, executed from one of the production database nodes:

```
$ rman target /

RMAN> backup device type disk format
   '/NAS/oracle/rman_backups/%U' database plus archivelog;

RMAN> backup device type disk format
   '/oradb/rman_backups/%U' current controlfile for
   standby;
```

Step 3.6.2 - Ship Backup to the new Database Server

Using the appropriate system tools, ship the backup to the new database server. The backup must be restored to the same folder name on the destination as where it was created on the source.

Step 3.6.3 - Start one RAC instance

Choose one of the RAC instances at the DR site and start it with the NOMOUNT option:

```
SQL> startup nomount
```

Step 3.6.4 - Restore the Database Backup to the Standby Using RMAN

Restore the database using RMAN on the same server that has the instance started NOMOUNT. The database must be restored with the "for standby" option, for example:

```
$ rman target sys/manager@ha1dbh01:1521/VIS auxiliary /
RMAN> duplicate target database for standby;
```

Step 3.6.5 - Start Managed Recovery

Pay close attention to which instance the following commands are executed.

On the **standby database**, check to make sure the database is in standby mode by executing this query from SQL*Plus connected as sysdba:

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

   DATABASE_ROLE----------------PHYSICAL STANDBY
```

On the **primary database** enable the previously deferred remote destination by executing this command from SQL*Plus connected as sysdba:

```
SQL> alter system set log_archive_dest_state_2=enable
   SID='*';
```

Also, update the database configuration "include" files on all nodes of the **primary database**:

```
log_archive_dest_state_2=enable
```

Place the **standby database** in managed recovery by executing this command from SQL*Plus connected as sysdba:

```
SQL> recover managed standby database using current logfile
   disconnect;
```

Step 3.6.6 - Verify Correct Standby Operation

Validate that the standby is correctly applying redo from the primary.

On the **primary database**, archive the current log using the following SQL statement:

```
SQL> alter system archive log current;
```

On the **standby database**, query the gv$archived_log view to verify that the logs are received and applied:

```
SQL> select thread#, sequence#, applied,
   to_char(first_time, 'mm/dd/yy hh24:mi:ss') first,
   to_char(next_time, 'mm/dd/yy hh24:mi:ss') next,
   to_char(completion_time, 'mm/dd/yy hh24:mi:ss')
   completion
   from gv$archived_log order by first_time;
```

Task 3.7 - Clone Application Software and Configure on the DR Site

Step 3.7.1 - Prepare the Application Tier for Cloning

Log on to the production system as the APPLMGR user and run the following commands:

```
$ cd <COMMON_TOP>/admin/scripts/<CONTEXT_NAME>
$ perl adpreclone.pl appsTier
```

Step 3.7.2 - Copy the Application Tier File System

Using appropriate tool make a copy of the Applications software and tech stack to the DR application servers. We will use this clone of the software to run the application after the switchover.

Step 3.7.3 - Configure Application Tier File System

Run the following commands on each application server., Specify the original context file or copy of it as the source context file to the adclonectx.pl tool.

```
$ cd /u01/appltopRAC/viscomn/clone/bin
$ perl adclonectx.pl
   contextfile=/u01/appltopRAC/visappl/admin/VIS_ha1mth01.x
   ml
```

Provide the values required for creation of the new Application Context file:

```
Do you want to use a virtual hostname for the target node
   (y/n) [n] ?: n
Target hostname [ha1mth01]: ha2mth01
Do you want the inputs to be validated (y/n) [n] ?:n
Target system database SID [VIS]:
Username for the applications file system owner [applmgr]:
Group for the applications file system owner [dba]:
Target system database server node [ha1dbh01]:  ha2dbh01
Target system database domain name [ha.us.oracle.com]:
Does the target system have more than one application tier
   server node (y/n) [y] ?: n
Is the target system APPL_TOP divided into multiple mount
   points (y/n) [n] ?:
Target system APPL_TOP mount point
   [/u01/appltopRAC/visappl]:
Target system COMMON_TOP directory
   [/u01/appltopRAC/viscomn]:
Target system 8.0.6 ORACLE_HOME directory
   [/u01/appltopRAC/visora/8.0.6]:
Target system iAS ORACLE_HOME directory
   [/u01/appltopRAC/visora/iAS]:
Do you want to preserve the Display set to ha1dbh01:0.0
   (y/n) [y] ?:n
Target system Display [ha2mth01:0.0]:
Location of the JDK on the target system [/opt/java1.4]:
```

```
Select the Perl to be used on the target system:
      1. /u01/appltopRAC/visora/iAS/Apache/perl/bin/perl
      2. /usr/bin/perl
Enter your choice [1]:
Do you want to preserve the port values from the source
   system on the target system (y/n) [y] ?:
New context path and file name
   [/u01/appltopRAC/visappl/admin/VIS_ha2mth01.xml]:
```

Step 3.7.4 - Finalize Configuration of the Application Tier File System

Run the following commands on each application server node, using the new context file generated in the previous step:

```
$ export APPL_TOP=/u01/appltopRAC/visappl
   (Workaround for known issue, Bug 5520384)
$ cd /u01/appltopRAC/viscomn/clone/bin
$ perl adcfgclone.pl appsTier
   /u01/appltopRAC/visappl/admin/VIS_ha1mth01.xml
```

Note: Since the database is not available at this time, you can ignore the error related to the failed AutoConfig run.

Step 3.7.5 - Adjust SQL*Net Settings

Since the database instance could not be accessed, the generated SQL*Net configuration is not correct and needs to be adjusted to successfully connect when you run AutoConfig on switchover. To do this:

Set the environment for the new software location:

```
$ cd <APPL_TOP>
$ . ./APPS<CONTEXT_NAME>.env
```

Edit $TNS_ADMIN/tnsnames.ora and modify the entry for your database to point to a particular instance. Here, we modified our definition for VIS to point to VIS1):

```
VIS=(DESCRIPTION=
       (ADDRESS=(PROTOCOL=tcp)(HOST=ha1dbh01)(PORT=1521))
       (CONNECT_DATA=(SID=VIS1))
    )
```

Note: We are editing the tnsnames.ora file directly instead of the tns include file, as this is a temporary configuration and we do not expect to run AutoConfig again until we switch to the new environment. Were we to run AutoConfig again, this change would need to be repeated.

Repeat this on all the application tiers and verify that it is working using tnsping

```
$ tnsping VIS
```

Step 3.7.6 - Configure Application Tiers for RAC

The adcfgclone script does not set variables to use the load balanced services. To adjust these parameters, edit the context file found at

$ORACLE_HOME/appsutil/<CONTEXT_NAME>.xml to set the value of "Tools OH TWO_TASK" (s_tools_two_task) "iAS OH TWO_TASK" (s_weboh_twotask), and Apps JDBC Connect Alias (s_apps_jdbc_connect_alias). Repeat the steps for all the application tiers.

To load balance the forms based applications database connections, set the value of "Tools OH TWO_TASK" to point to the <database_name>_806_balance alias generated in the tnsnames.ora file.

To load balance the self-service applications database connections, set the value of iAS OH TWO_TASK" and "Apps JDBC Connect Alias" to point to the <database_name>_balance alias generated in the tnsnames.ora file

If you are load balancing for Concurrent Processing, set the value of "Concurrent Manager TWO_TASK" (s_cp_twotask) to point to the <database_name>_806_balance alias generated in the tnsnames.ora file.

If there is more than one Concurrent Processing Node and you would like to take advantage of PCP refer to "Configure Parallel Concurrent Processing" section of the Note 362135.1.

Task 3.8 - Update Clusterware Configuration

If you want to use srvctl to control the resources you need to add the new resources to CRS. See the earlier section Update Clusterware Configuration for details of this task.

PHASE 4 - ONGOING SWITCHOVER AND FAILOVER TESTING

The DR site can be used via switchover to provide application services while the production platform or site is undergoing planned maintenance or via failover if the production site is rendered unusable by a severe unplanned outage. We recommend that switchover and failover procedures are tested regularly to validate your MAA configuration.

Task 4.1 - Switchover Procedure

The steps to switch over to the standby database are as follows. The E-Business Suite application will be down during this stage, so it is recommended that the steps be carefully practiced and scripted.

Step 4.1.1 - Prepare the Standby Database

If the database has been opened read-only since it was last shut down, bring it down and back up.

Make sure that managed recovery is running, and is up to date.

Shut down all RAC instances but the one performing recovery on the standby site.

Check to see if flashback has been enabled with the following query:

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

   FLASHBACK_ON
   ------------
   NO
```

Step 4.1.2 - Shutdown E-Business Suite and Extra Database Instances

Shut down the E-Business Suite completely. Ensure the application is completely shutdown.

Stop all RAC instances but one, on the production site.

Step 4.1.3 - Switchover to Standby Database

On the **primary database** commit to switchover to standby:

```
SQL> alter database commit to switchover to standby;
```

Note: If the status returns SESSIONS_ACTIVE then you should issue the switchover command using the WITH SESSION SHUTDOWN clause.

```
SQL> alter database commit to switchover to standby with
   session shutdown;
```

Shutdown the **primary database** and keep it down for now:

```
SQL> shutdown immediate
```

Stop the **primary** database listener: on each node:

```
$ lsnrctl stop LISTENER_<hostname>
```

On the **standby database**, verify that it is ready to be converted to the new primary:

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

   SWITCHOVER_STATUS
   -----------------
   TO PRIMARY
```

On the **standby** database, execute the following command to convert it to be the new primary:

```
$ alter database commit to switchover to primary;
```

NOTE: The following steps call this site "New Primary."

Step 4.1.4 - Enable Flashback on the New Primary (Optional)

If flashback has not yet been enabled on this database, enable it now with this command:

```
SQL> alter database flashback on;
```

Step 4.1.5 - Open the New Primary Database

```
SQL> alter database open;
```

Step 4.1.6 - Start Other Instances on the New Primary

Other instances of the new primary database can be started as normal:

```
SQL> startup
```

Step 4.1.7 - Remove the Old Application Topology

Connect to the new primary database using SQL*Plus as user APPS and execute the following commands:

```
SQL> exec FND_NET_SERVICES.remove_system('VIS');
SQL> commit;
```

Step 4.1.8 - Configure New Primary Database Tier

Run the following command on each new primary database node to configure the Oracle home for use by the E-Business Suite:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Note: This command needs to be run once on each database node to register the node, then after all nodes are registered it must be run once again on each node to generate the correct SQL*Net configuration files.

Step 4.1.9 - Restart Listeners

Start the correct listener on each new primary database node:

```
lsnrctl stop <SID><INSTANCE_NUMBER>
lsnrctl start LISTENER_<HOSTNAME>
```

Step 4.1.10 - Configure Application Tiers

Run AutoConfig on all the new primary application tier nodes. This step can be run in parallel.

```
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 4.1.11 - Start E-Business Suite

Start the E-Business Suite applications on the new primary site. On each application tier node:

```
$ cd <APPL_TOP>
$ . ./APPS<CONTEXT_NAME>.env
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adstrtal.sh
```

Online users can gain access at this time.

Step 4.1.12 - Start Original Primary as Standby

Start the database listeners on at least one database node at the original primary site:

```
$ lsnrctl start LISTENER_<HOSTNAME>
```

On the same node, start the database in mount mode, then start managed recovery:

```
SQL> startup mount;
SQL> recover managed standby database using current logfile
   disconnect;
```

On the **new primary database** enable remote archiving by executing this command from SQL*Plus connected as sysdba:

```
SQL> alter system set log_archive_dest_state_2=enable
   SID='*';
```

Step 4.1.13 - Perform the cloning finishing tasks

Perform the "Finishing Tasks" outlined in the Oracle*MetaLink* note 230672.1 – Cloning Oracle Applications Release 11*i* with Rapid Clone:

Step 4.1.14 - Direct users to the new system.

The ex-standby system should be available to your users as your new production system. Direct your users to the new URL.

Task 4.2 - Switch Back Procedure

It may be necessary to switch back to the primary after the switchover. The process to switch back to the original primary is the mirror image of the switchover. Please see the previous task for details.

Task 4.3 - Failover Procedure

In this section we assume that the primary site has been lost and a disaster site is available. We document the steps to switch the production system to the disaster site and then enable the original primary as a Data Guard standby using the Flashback feature.

The E-Business Suite application will be down during this stage, so it is recommended that the steps are carefully practiced and scripted so that they can be executed quickly in an emergency.

Step 4.3.1 - Failover to Standby Database

On the **standby** database, execute the following command to convert it to be the new primary:

```
SQL> recover managed standby database cancel;
SQL> recover managed standby database finish force;
SQL> alter database commit to switchover to primary;
```

Note: The following steps refer to this environment as the "new primary".

Step 4.3.2 - Enable Flashback

If flashback has never been enabled for the new primary database, you should enable it now. If you enable it and it has already been enabled, the command will simply give you a warning.

```
SQL> alter database flashback on;
```

To check flashback status, run the following query:

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

   FLASHBACK_ON
   ------------
   YES
```

Step 4.3.3 - Open the New Primary Database

```
SQL> alter database open;
```

Note: If the database has been opened read-only since it was last shut down, it will need to be shut down and restarted.

Step 4.3.4 - Start Other Rac Instances on the New Primary site

The other RAC instances can be started normally:

```
SQL> startup
```

You may continue with the next step while the instances are starting.

Step 4.3.5 - Remove the Old Application Topology

Connect to the new primary database using SQL*Plus as user apps and execute the following commands:

```
SQL> exec FND_NET_SERVICES.remove_system('VIS');
```

```
SQL> commit;
```

Step 4.3.6 - Configure Database Tier

Run the following command once on each new primary database node to add that node to the application topology. Then once all nodes are in the topology, it must be run once again on each node to generate the correct network configuration files:

```
$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 4.3.7 - Restart Listeners

Start the correct listener on each new primary database node:

```
$ lsnrctl stop <SID><INSTANCE_NUMBER>
$ lsnrctl start LISTENER_<HOSTNAME>
```

Step 4.3.8 - Configure Application Tier

Run AutoConfig on all new primary application tier nodes. This can be run in parallel.

```
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adautocfg.sh
```

Step 4.3.9 - Startup E-Business Suite

Startup the E-Business Suite applications on each new primary application tier node.

```
$ cd <APPL_TOP>
$ . ./APPS<CONTEXT_NAME>.env
$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
$ ./adstrtal.sh
```

Online users can gain access at this time.

Step 4.3.10 - Establish Original Primary as Standby Using Flashback (Optional)

Once access to the failed production site is restored, if you had flashback database enabled, you may be able to reinstate the original primary database as a physical standby of the new primary database.

Make sure Oracle Clusterware and ASM are restarted on the **original** primary system.

On the **new primary site**, get the SCN when the database became primary:

```
SQL> select to_char(standby_became_primary_scn) from
   v$database;
```

On the **original primary site**, flashback and start managed recovery:

```
SQL> shutdown immediate;
SQL> startup mount;
```

```
SQL> flashback database to scn
   <standby_became_primary_scn>;

SQL> alter database convert to physical standby;

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database recover managed standby database using
   current logfile disconnect;
```

Step 4.3.11 - Perform the cloning finishing tasks

Perform the "Finishing Tasks" outlined in the Oracle*MetaLink* note 230672.1 – Cloning Oracle Applications Release 11*i* with Rapid Clone:

Step 4.3.12 - Direct users to the new system.

The ex-standby system should be available to your users as your new production system. Direct your users to the new URL.

Task 4.4 - DR Testing Procedure using Flashback Database

In this section we document how the DR configuration can be tested while the primary site is in live operation, and how Flashback Database can be used to quickly restore the DR site to standby operation afterwards.

As a starting point we assume that the primary site is in live operation and the DR site is in standby mode and applying redo.

Step 4.4.1 - Activate and Open the DR Standby Database

On the **DR site**, cancel managed recovery:

```
ha2dbh01:SQL> recover managed standby database cancel;
```

On the **DR site**, enable flashback if it has not been enable before:

```
ha2dbh01:SQL> alter database flashback on;
```

On the **DR site**, create a guaranteed restore point (named "testing _starts" in this example):

```
ha2dbh01:SQL> create restore point testing_starts guarantee
   flashback database;
```

On **primary site**, switch the current log and then defer the archive destination:

```
ha1dbh01:SQL> alter system archive log current;

ha1dbh01:SQL> alter system set
   log_archive_dest_state_2=defer SID='*';
```

On **DR site**, activate and open the database:

```
ha2dbh01:SQL> alter database activate standby database;

ha2dbh01:SQL> alter database set standby database to
   maximize performance;

ha2dbh01:SQL> alter database open;
```

Step 4.4.2 - Perform Testing

The database is now open and can be used for testing. Any changes that are made to the database will be rolled back afterwards using flashback database. You can start additional databases instances and test the application as you wish.

Note that the DR site will be getting behind on redo application during the testing period and so make sure that you do not get too far behind.

Step 4.4.3 - Flashback the Database and Resume Standby Operation

On the **DR site**, shutdown all but one database instance. In this example, only the instance on ha2dbh01 remains.

On the **DR site**, flashback and start managed recovery:

```
ha2dbh01:SQL> startup mount force;
ha2dbh01:SQL> flashback database to restore point
   testing_starts;
ha2dbh01:SQL> drop restore point testing_starts;
ha2dbh01:SQL> alter database convert to physical standby;
ha2dbh01:SQL> startup mount force;
ha2dbh01:SQL> alter database recover managed standby
   database using current logfile disconnect;
```

On the **primary site**, enable the archive destination:

```
ha1dbh01:SQL> alter system set
   log_archive_dest_state_2=enable SID='*';
```

Task 4.5 - Automating Switchover and Failover Procedures

In this section we provide an example of how to achieve complete automation of switchover and failover procedures by utilizing Oracle Data Guard Broker.

The Oracle Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations. The broker automates and simplifies the following operations:

- Management of an entire Data Guard configuration, including all databases, redo transport services, and log apply services, through a client connection to any database in the configuration.
- Management of the protection mode for the broker configuration.
- Invoking switchover or failover with a single command to initiate and control complex role changes across all databases in the configuration.
- Configuring failover to occur automatically upon loss of the primary database, increasing availability without manual intervention. This is known as Fast-Start Failover (FSFO).
- Monitoring the status of the entire configuration, capturing diagnostic information, reporting statistics such as the log apply rate and the redo

generation rate, and detecting problems quickly with centralized monitoring, testing, and performance tools.

All management operations can be performed locally or remotely through the broker's easy-to-use interfaces: the Data Guard management pages in Oracle Enterprise Manager, which is the broker's graphical user interface (GUI), and the Data Guard command-line interface called DGMGRL.

In our example we have automated the entire switchover/failover process including automatic triggering of failover. It is possible to omit certain pieces, such as automatic triggering, if they do not make sense for your implementation.

The process works like this:

1. Fast-Start Failover (FSFO) determines that a failover is necessary and initiates a failover to the standby database automatically, or the administrator instructs the broker to switchover or failover.
2. When the database switchover/failover has completed the DB_ROLE_CHANGE database event is fired.
3. The event causes a trigger to be fired which calls a script that configures and starts the E-Business Suite application..

As a starting point for our setup procedures we assume that the primary site is in live operation and the DR site is in standby mode and applying redo.

Step 4.5.1 - Develop E-Business Suite Configuration and Startup Script

Develop a script that will automate the E-Business Suite configuration and startup process. This equates to the following steps in the switchover procedure:

Step 4.1.7 - Remove the Old Application Topology

Step 4.1.8 - Configure New Primary Database Tier

Step 4.1.10 - Configure Application Tiers

Step 4.1.11 - Start E-Business Suite

An example script is provided for your reference in Appendix 2. Modify the script to suit your environment and requirements.

Make sure ssh (or equivalent) is configured so that remote shell scripts can be executed without password prompts.

Step 4.5.2 - Automate Script Execution by Trigger

Create a database event "DB_ROLE_CHANGE" trigger, which fires after database role changes from standby to primary. For example:

```
CREATE OR REPLACE TRIGGER postover
AFTER DB_ROLE_CHANGE ON DATABASE
DECLARE
```

```
  v_db_unique_name varchar2(30);
BEGIN
  select upper(VALUE) into v_db_unique_name
  from v$parameter where NAME='db_unique_name';

  dbms_scheduler.create_job(
    job_name=>'postover',
    job_type=>'executable',
    job_action=>
      '/NAS/oracle/FSFO/' || v_db_unique_name||'.fsfo.sh',
    enabled=>TRUE
  );
END;
```

The trigger calls a wrapper script named <DB_UNIQUE_NAME>.fsfo.sh which in turn calls the fsfo.sh script. This is done because it is not possible to directly pass arguments to a script from dbms_scheduler. A wrapper script must be created for the primary and DR databases:

In our case, we create a script named VIS_HA1.fsfo.sh as follows:

```
#!/bin/sh
/NAS/oracle/FSFO/fsfo.sh VIS_HA1
```

And, we create a script named VIS_HA2.fsfo.sh as follows:

```
#!/bin/sh
/NAS/oracle/FSFO/fsfo.sh VIS_HA2
```

Step 4.5.3 - Configure Fast-Start Failover

Follow the steps in Oracle Database 10g Release 2 Best Practices: Data Guard Fast-Start Failover to configure Fast-Start Failover.

As always, use include files (ifiles) for additional SQL*NET configuration so that your changes are preserved across AutoConfig executions.

APPENDIX 1 - HP SYSTEM SPECIFICATIONS

Application Servers

- HP 9000 rp3440 servers
- Mirrored 146GB disk drives internal
  - HP-UX operating system software
- NFS mounted shared APPL_TOP
- Networks:
  - MP (Maintenance Port) 100TX
  - Public 1GB
  - Private 1GB (Dedicated Application Tier to DB-Tier network)
  - NAS 1GB (Dedicated NAS network)

Database Servers

- HP Integrity rx4640 servers
- Mirrored 72GB disk drives internal
  - HP-UX operating system software
  - Local $ORACLE_HOME
- HP StorageWorks SAN solution (Shared Storage)
  - HP StorageWorks Enterprise Virtual Array 4000 (EVA4000) 2TB
  - HP StorageWorks dual port 2GB Fiber Channel HBA
  - HP StorageWorks SAN Switch 2GB ports
- Networks:
  - MP (Maintenance Port) 100TX
  - Public 1GB
  - Private 1GB (Dedicated Application to DB-Tier network)
  - Cluster Interconnect Primary 1GB
  - Cluster Interconnect Secondary 1GB
  - WAN Link 1GB (Dedicated network to connect site1 and site2 - Dataguard)

APPENDIX 2 - E-BUSINESS SUITE SWITCHOVER/FAILOVER SCRIPT

Below is an example script that will configure and startup the E-Business Suite application after a switchover or failover, including running AutoConfig on the database tier. It is designed to be run from any database node and it uses "ssh" to run remote commands on the other nodes.

```
#!/bin/sh

# Enable/Disable the script,
# set value to 1 to perform the steps in the script
############################################################
ENABLED=0

# Arg1 DB_UNIQUE_NAME determines the site
# that needs to be activated.
############################################################
DB_UNIQUE_NAME=$1

# Constants, modify according to your environments
############################################################
DB_NAME=VIS

SITE1=VIS_HA1
SITE2=VIS_HA2

DB_NODES_SITE1="ha1dbh01 ha1dbh02"
DB_NODES_SITE2="ha2dbh01 ha2dbh02"

APPS_NODES_SITE1="ha1mth01 ha1mth02 ha1mth03"
APPS_NODES_SITE2="ha2mth01"

OH=/u01/app/oracle/visdbRAC/10.2.0

AT=/u01/appltopRAC/visappl
CT=/u01/appltopRAC/viscomn

DBOSUSER=oracle
APPSOSUSER=applmgr

APPSPWD=APPS
```

```
# Logfile
#############################################################
LOGF=/NAS/oracle/FSFO/fsfo.log
exec >>$LOGF 2>&1

# Start executing
#############################################################

echo ""
echo "-----------------------------------------------------"
echo "script started at `date`"
echo "-----------------------------------------------------"
echo ""

# Initialize the variables for the correct Site
#############################################################
if [ ${DB_UNIQUE_NAME}x = ${SITE1}x ]; then
  DB_NODES=$DB_NODES_SITE1
  APPS_NODES=$APPS_NODES_SITE1
elif [ ${DB_UNIQUE_NAME}x = ${SITE2}x ]; then
  DB_NODES=$DB_NODES_SITE2
  APPS_NODES=$APPS_NODES_SITE2
else
  echo "`date` -- Error !"
  echo "(Err) Missing/Invalid argument DB_UNIQUE_NAME:
   \"$DB_UNIQUE_NAME\""
  exit 1
fi

echo "`date` -- Start Configuring"
echo "Site: $DB_UNIQUE_NAME on `hostname` as `id`"
echo "-----------------------------------------------------"

# Start Configuring the Site if script enabled
#############################################################
if [ $ENABLED -ne 1 ]; then
  echo "(Wrn) Script is not enabled: No Config Changes."
  exit 0
fi

# env needs to set to be able to use sqlplus or srvctl
echo "`date` -- Setting Environment"
```

```
echo "------------------------------------------------"
. ${OH}/${DB_NAME}?_`hostname`.env

sqlplus -S APPS/$APPSPWD << EOF
rem set feedback off
BEGIN
  FND_NET_SERVICES.remove_system('${DB_NAME}');
END;
/
commit;
EOF

echo "`date` -- Run Autoconfig on DB Nodes first Round"
echo "------------------------------------------------"

for node in $DB_NODES; do
  ssh ${DBOSUSER}@$node
   "${OH}/appsutil/scripts/${DB_NAME}?_${node}/adautocfg.sh
   -appspass=$APPSPWD" &
done
wait

echo "`date` -- Run Autoconfig on DB Nodes second Round"
echo "------------------------------------------------"

for node in $DB_NODES; do
  ssh ${DBOSUSER}@$node
   "${OH}/appsutil/scripts/${DB_NAME}?_${node}/adautocfg.sh
   -appspass=$APPSPWD" &
done
wait

echo "`date` -- Run Autoconfig on APPS Nodes"
echo "------------------------------------------------"

for node in $APPS_NODES; do
  ssh ${APPSOSUSER}@$node
   "${CT}/admin/scripts/${DB_NAME}_${node}/adautocfg.sh -
   appspass=$APPSPWD" &
done
wait

echo "`date` -- Start Apps on All Nodes"
echo "------------------------------------------------"
```

```
for node in $APPS_NODES; do
  ssh ${APPSOSUSER}@$node
   "${CT}/admin/scripts/${DB_NAME}_${node}/adstrtal.sh
   APPS/$APPSPWD" &
done
wait


echo ""
echo "------------------------------------------------------"
echo "script completed at `date`"
echo "------------------------------------------------------"
echo ""
```

REFERENCES

Oracle Maximum Availability Architecture (MAA) on OTN:
http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

Oracle Data Guard Overview
http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardOverview.html

Oracle® Database Reference 10*g* Release 2:
http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14237/toc.htm

Interoperability Notes - Oracle Applications 11i with Oracle Database 10g Release 2:
http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=362203.1

Oracle® Database Administrator's Guide 10*g* Release 2:
http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14231/toc.htm

Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide for HP Itanium
http://www.oracle.com/pls/db102/homepage

Oracle*MetaLink* Note 185703.1 - How to Avoid Common Flaws and Errors Using Passwordfile
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=185703.1

Oracle*MetaLink* Note 216211.1 - Nologging In The E-Business Suite
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=216211.1

Oracle® Data Guard Concepts and Administration 10g Release 2 (10.2)
http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14239/toc.htm

Oracle*MetaLink* Note 281758.1 - Additional Features in Oracle Applications Manager in Oracle Applications Release 11.5.10
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=281758.1

Oracle® Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide 10g Release 2
http://download-west.oracle.com/docs/cd/B19306_01/rac.102/b14197/toc.htm

Oracle*MetaLink* Note 362135.1- Configuring Oracle Applications Release 11i with 10g Release2 Real Application Clusters and Automatic Storage Management
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=362135.1

HP Integrity rx4640 server overview
http://www.hp.com/products1/servers/integrity/entry_level/rx4640/index.html

HP 9000 rp3440 server overview
http://www.hp.com/products1/servers/rackoptimized/rp3400_series/index.html

HP StorageWorks information site
http://h18006.www1.hp.com/storage/index.html

HP-UX 11i v2 for HP 9000 and Integrity Servers
http://www.hp.com/products1/unix/operating/choice/hpux_itanium.html

HP-UX 11i Systems Administration Handbook, Second Edition, Prentice Hall PTR; 2nd edition (February 10, 2003)

Oracle Applications Release 11*i* (11.5.10) Concepts
http://oraclesvca2.oracle.com/docs/cd/B16981_01/current/acrobat/11iconcepts.pdf

Oracle 9*i* Release Notes for HP-UX 11i for the Itanium Processor Family
http://download-west.oracle.com/docs/html/B10566_05/toc.htm

Oracle*MetaLink* Note 304489.1- Using Oracle Applications with a Split Configuration Database Tier on Oracle 9i Release 2
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=304489.1

Oracle*MetaLink* Note 216550.1- Interoperability notes, Oracle Applications Release 11*i* with Oracle 9*i* Release 2 (9.2.0)
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=216550.1

Oracle*MetaLink* Note 216205.1- Database Initialization Parameters for Oracle Applications11i
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=216205.1

The CVU download and FAQ
http://www.oracle.com/technology/products/database/clustering/cvu/cvu_download_homepage.html

Transitioning E-Business Suite to the Maximum Availability Architecture on HP Systems
October 2006
Authors: Richard Exley (Oracle), Lyn Pratt (Oracle), Michael Nowak (Oracle), Metin Yilmaz (Oracle), Andrew Babb (Oracle), Anthony Greathouse (HP)

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