Transitioning Oracle E-Business Suite to the Maximum Availability Architecture on the Solaris Operating System

An Oracle White Paper
September 2007

Executive Summary

Designed to remove the complexity of creating an optimal high availability architecture, the Oracle Maximum Availability Architecture (MAA) is a best practice blueprint based on proven Oracle high-availability technologies and recommendations. Published as part of a series of MAA white papers, this document explains how to transition an Oracle E-Business Suite application to an MAA configuration yet keep application downtime to a minimum. The most current version of this document is located at Oracle Maximum Availability Architecture (MAA).

The procedures outlined in this document describe how to transition from Oracle E-Business Suite 11.5.10.2 running a single instance on Oracle Database 10*g* Release 2 to a clustered system. Indeed, the ultimate configuration consists of Oracle E-Business Suite 11.5.10.2 running on two nodes with Oracle Real Application Clusters (Oracle RAC), Oracle Flashback, Oracle Automatic Storage Management (ASM), Oracle Clusterware, Solaris Cluster software, and a disaster recovery site leveraging Oracle Data Guard. All steps detailed in this document were tested, with Oracle E-Business Suite application downtime limited to five minutes for the transition to MAA. In addition, an approach was formulated for the switchover and failover to the disaster site that limit Oracle E-Business Suite application downtime to five minutes.

For additional information please see the Oracle Applications Maximum Availability Architecture information located at http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm#Applications.

HIGHLIGHTS

- Presents Oracle's Maximum Availability Architecture
- Describes the technologies used to create effective deployments
- Defines a four-phased approach for success
- Provides step-by-step guidelines for development and deployment
- Highlights the benefits of the Maximum Availability Architecture approach

Introduction

Several technologies are used in this study to create an MAA environment for the Oracle E-Business Suite Application. This configuration consists of Oracle RAC and Oracle Clusterware (which is used for internal communication) along with the Solaris Cluster software, Oracle Data Guard, Oracle Flashback, and Oracle Automatic Storage Management. Storage technologies include Sun StorageTek Network Attached Storage (NAS), Sun StorageTek Fibre Channel RAID systems, and Sun StorageTek tape and tape automation systems. Figure 1 provides a high-level overview of the Oracle Maximum Availability Architecture.

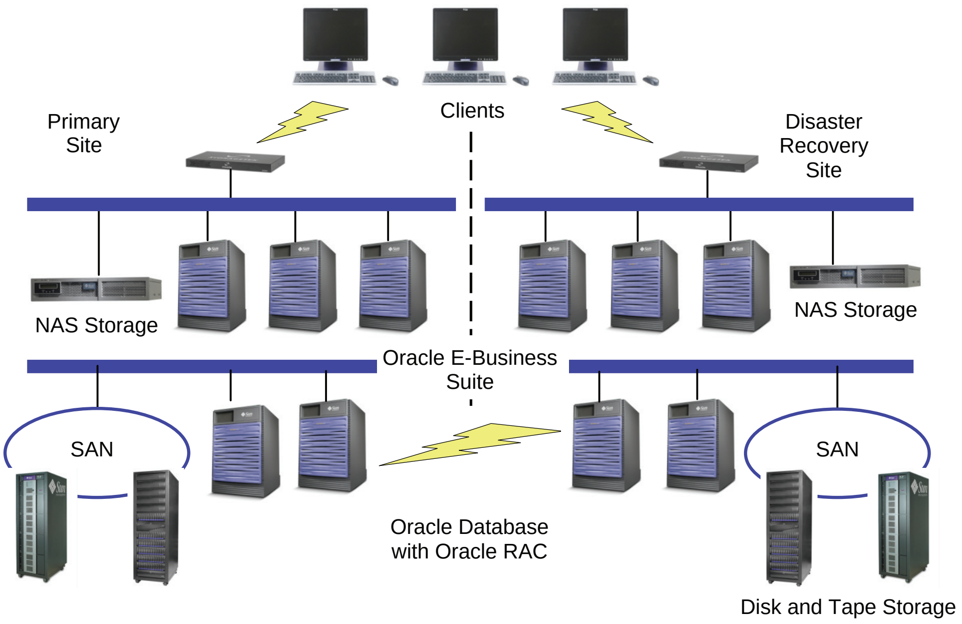


Figure 1. Oracle Maximum Availability Architecture

Solaris Cluster Software

The Solaris Cluster software is a premier high availability platform for improving the predictability and resilience of business-critical applications. Designed and optimized for the Solaris 10 Operating System (OS), the Solaris Cluster software ensures the reliable and fast failover of mission-critical applications.

Customers can optionally use Solaris cluster with Oracle Clusterware for Oracle RAC deployments. Solaris Cluster is a mature product with an extensive, robust, and proven feature set that offers protection, resiliency, and fast failure detection and recovery for Oracle RAC deployments. Overall, deploying the Solaris Cluster software in combination with Oracle Clusterware can help organizations guard against outages caused by data corruption, system hangs, or misconfiguration. The Solaris Cluster 3.2 software includes several features specifically designed to work well with the manageability of Oracle RAC deployments. These new resource types ensure close coordination of the Oracle startup and shutdown processes in a clustered environment and minimize the likelihood of failures.

Oracle Real Application Clusters

Oracle Real Application Clusters (Oracle RAC) enables Oracle Database to run commercially available and custom applications unchanged across a set of clustered nodes in order to provide high availability and flexible scalability. If a clustered node fails, Oracle Database continues to run on the surviving nodes. When more processing power is needed, another node can be added to the cluster 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. Oracle Data Guard maintains the standby databases as transactionally consistent copies of the production database. If the production database becomes unavailable due to a planned or an unplanned outage, Oracle Data Guard can switch any standby database to the production role, significantly reducing application downtime. Oracle 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 Database

Oracle Flashback Database quickly rewinds Oracle Database to a previous time to correct problems caused by logical data corruptions or user errors. It provides database point in time recovery without requiring a backup of the database to be restored, speeding recovery efforts. This document explains how to use Oracle Flashback Database to return the production database to standby mode after failover without a lengthy database restore, thereby accelerating the return to normal production operation after a disaster.

Oracle Automatic Storage Management

Oracle Automatic Storage Management (ASM) provides a vertically integrated file system and volume manager directly in the Oracle kernel. In order to provide optimal performance and protect against data loss, Oracle ASM spreads files across all available storage and extends the concept of the stripe and mirror everything (SAME) approach. Indeed, Oracle ASM is the implementation and automation of SAME. For greater flexibility, Oracle ASM enables database files to be mirrored, rather than the traditional approach of mirroring entire disks.

Key benefits of Oracle ASM include:

- Minimizes the work required to provision database storage
- Increases availability levels
- Eliminates the expense, installation, and maintenance of specialized storage products
- Offers unique capabilities to database applications

Oracle Clusterware

Oracle Clusterware is a portable solution that is integrated and designed specifically for Oracle Database[1]. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle resources, such as instances and listeners. In the event of a failure, Oracle Clusterware automatically restarts the failed resource or relocates the processing performed by the failed component to a backup resource. For example, if a node in the cluster fails, Oracle Clusterware moves services used by the application and notifies client processes to reconnect to a surviving node.

High availability configurations incorporate redundant hardware and software components that maintain operation by avoiding single points of failure. As a result, Oracle Clusterware is installed as part of the Oracle RAC installation process.

[1] While optimized for Oracle Database, Oracle Clusterware can support third-party applications as well

Sun StorageTek Fibre Channel Hardware RAID

Hardware RAID systems with Fibre Channel attachments provide low latency, high bandwidth solutions that are well-suited for Oracle database files. The Sun StorageTek 6540 array is an ideal building block for supporting Oracle databases in the MAA environment. Using Fibre Channel technology and hardware RAID techniques, the Sun StorageTek 6540 array provides significant benefits to Oracle Databases, including:

- Isolates the database from hardware faults in the storage system
- Reduces storage service and maintenance costs
- Fosters predictable data access times for specific data access rates that aids information lifecycle management (ILM)

Sun StorageTek NAS Appliance

Network Attached Storage (NAS) eases storage deployment and management costs and supports the application tier needs of Oracle systems. The Sun StorageTek 5320 NAS Appliance uses Sun StorageTek Fibre Channel systems for complete hardware protection, along with a NAS gateway that helps organizations quickly deploy, protect, and maintain data in an Oracle database deployment for key applications that require shared access to data for a variety of purposes. Figure 2 depicts a data mining instance with Oracle software deployed on a Sun enterprise class server utilizing grid computing to accomplish key elements of a data mining application.

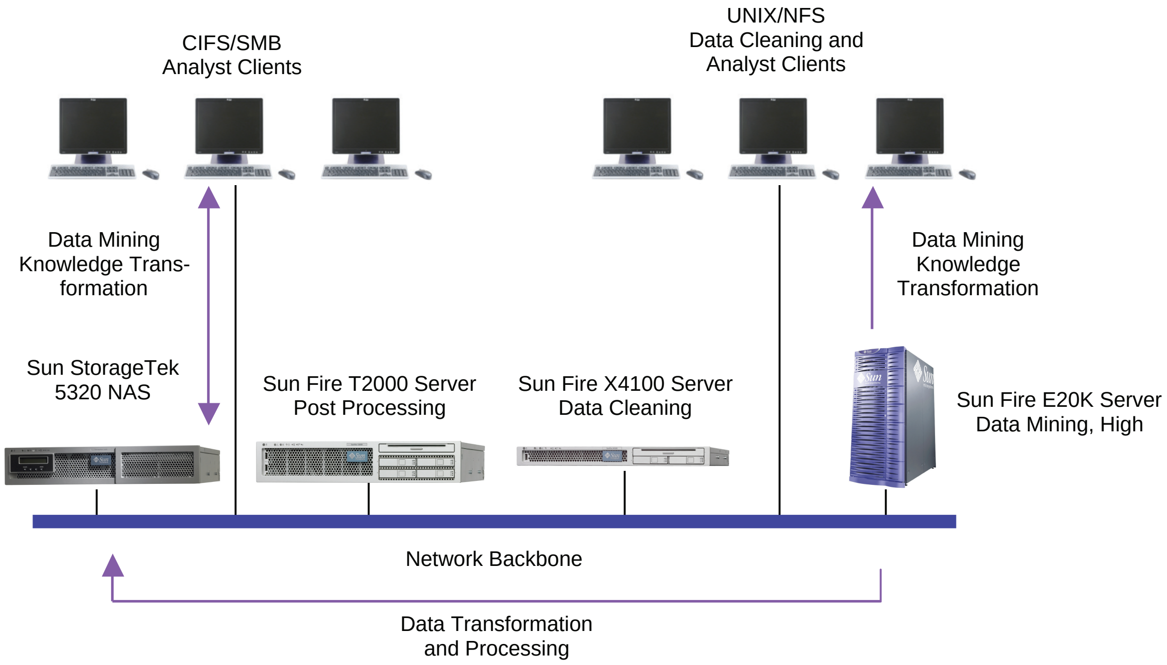


Figure 2. Sun's network attached storage systems enable data to be shared.

Sun StorageTek Tape and Tape Automation Systems

Low cost, robust, removable storage is vital for complete data protection. The combination of Sun StorageTek Tape and Tape Automation systems and media management solutions such as Oracle Secure Backup, provides a cost-effective and scalable data protection architecture.

Approach

Transitioning to MAA with minimal application downtime requires a four-phased approach (Figure 3).

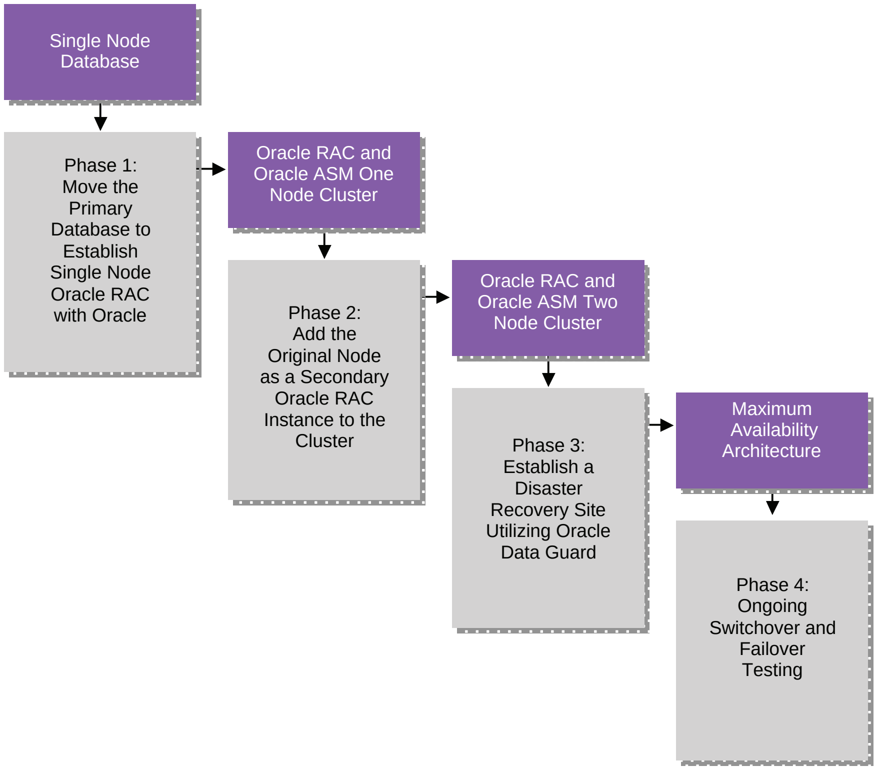

Single Node Database
Phase 1: Move the Primary Database to Establish Single Node Oracle RAC with Oracle
Oracle RAC and Oracle ASM One Node Cluster
Phase 2: Add the Original Node as a Secondary Oracle RAC Instance to the Cluster
Oracle RAC and Oracle ASM Two Node Cluster
Phase 3: Establish a Disaster Recovery Site Utilizing Oracle Data Guard
Maximum Availability Architecture
Phase 4: Ongoing Switchover and Failover Testing

Figure 3. A four-phased approach eases the transition process

Phase 1 — Move the Primary Database to an Established Single Node Oracle RAC with Oracle ASM

The first step in the transition process involves creating a single node Oracle Database 10*g* Release 2 with Real Application Clusters instance on a new database server utilizing the Solaris Cluster software, Oracle Clusterware, and Oracle ASM. A backup of the production database is used to create a local Oracle Data Guard physical standby instance for the production database on the new server. Finally, the production environment is created by switching to the new database server and enabling Oracle Flashback Database.

Phase 2 — Add the Original Node as a Secondary Oracle RAC Instance to the Oracle RAC Cluster

Once the application is stabilized on the new database server, the original database server can be taken out of service. Next, cluster related hardware is installed, and the Solaris Cluster software, Oracle Clusterware, Oracle ASM, and Oracle RAC are established on the original production server. The production server is then added to the cluster to create a two node Oracle RAC cluster.

Phase 3 — Establish Disaster Recovery Site with Oracle Data Guard

Before the transition to MAA can be complete, a disaster recovery site must be established. Toward this end, a two node Oracle Database 10*g* Release 2 with Real Application Clusters cluster is created on servers at the disaster recovery site using the Solaris Cluster software, Oracle Clusterware, and Oracle ASM. The Oracle Data Guard physical standby is instantiated by using backups taken from the production database. The Oracle E-Business Suite application tier software is cloned from the primary site to the disaster site. The standby system applies changes received from the primary site to ensure that it stays up to date and is ready to take over in the event of an emergency.

Phase 4 — Ongoing Switchover and Failover Testing

The disaster recovery site can be made active and provide application services via a switchover when the production platform or site is performing planned maintenance. In the event the production site is unavailable due to a severe unplanned outage, the disaster site can be made active via a failover of services. Phase 4 tests the switchover and failover procedures. It is important to test these procedures on a regular basis to validate the MAA configuration.

Application Downtime

Minimizing application downtime during the transition process is critical to establishing the MAA in an organization. Testing efforts revealed the ability to limit Oracle E-Business Suite application downtime to five minutes for the switchover to Oracle RAC. The table below details the time taken for each step of the process.

Step	Downtime Steps to Switch to Oracle RAC	Time (minutes:seconds)
1.10.3	Switchover to the local standby database	0:43
1.10.4	Enable Oracle Flashback	0:01
1.10.5	Open the database	0:05
1.10.6	Remove the old application topology	0:02
1.10.7	Configure the database tier	1:34
1.10.8	Restart the listeners	0:02
1.10.9	Configure the application tiers	2:50
	Total	**5:17**

During the disaster simulation test, Oracle E-Business Suite application downtime was limited to five minutes for the failover to a disaster recovery site. The table below lists the time taken for each step in the failover process. It is important to note that the time needed to start and stop services is not included in the downtime estimates, as the time needed can vary in each environment. Application shutdown and startup took approximately three minutes during testing efforts.

Step	Downtime Steps to Failover to Disaster Recovery Site	Time (minutes:seconds)
4.3.1	Failover to the disaster recovery database	0:09
4.3.2	Enable Oracle Flashback	0:01
4.3.3	Open the disaster recovery database	0:12
4.3.5	Remove the old application topology	0:02
4.3.6	Configure the disaster recovery database tier	1:34
4.3.7	Restart the listeners	0:02
4.3.8	Configure the application tiers	2:50
	Total	**4:50**

Example Environment

The table below outlines the testing environment, including node names and system configurations used for each phase of the transition. Based on a UNIX system, the environment begins with a database named VIS located on node ha1db. The database transitions to a two node Oracle RAC cluster running on nodes ha1db01 and ha1db02, and a disaster recovery site is established on nodes ha2db01 and ha2db02. Note that ha1db is renamed to ha1db02 during the transition process.

	Primary Site			Disaster Site
	Single Node Database	Single Node Oracle RAC and Oracle ASM	Two Node Oracle RAC and Oracle ASM	
Network Domain	ha.us.oracle.com	ha.us.oracle.com	ha.us.oracle.com	ha.us.oracle.com
Application Tier Nodes	ha1mt01 ha1mt02	ha1mt01 ha1mt02	ha1mt01 ha1mt02	ha2mt01 ha2mt02
APPL_TOP	/u01/appltop	/u01/appltopRAC	/u01/appltopRAC	/u01/appltopRAC
Database Tier Nodes	ha1db	ha1db01	ha1db01 ha1db02 (renamed from ha1db)	ha2db01 ha2db02
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
DB File Location	/oradb/oradata/visdata	+DATA/VIS_ha1/datafile	+DATA/VIS_ha1/datafile	+DATA/VIS_ha2/datafile
Log File Location	/oradb/oradata/visdata	+DATA/VIS_ha1/onlinelog	+DATA/VIS_ha1/onlinelog	+DATA/VIS_ha2/onlinelog

Database Instance Configuration

When configuring a database in an Oracle E-Business Suite environment, it is essential to make changes to the parameter include file (ifile) located at $ORACLE_HOME/dbs/*CONTEXT_NAME*_ifile.ora to ensure the AutoConfig utility does not overwrite parameters. Information on all database parameters can be found in the Oracle Database Reference.

During the transition to Oracle RAC, the original instance and the Oracle RAC instances must be configured. In addition, the primary and standby instances must be configured when establishing the disaster recovery site. The discussions below detail the configurations used in the test environment. Organizations should prepare similar configuration files and scripts and have them available to use at the appropriate steps in the process. Descriptions of some of the pertinent parameters follow.

parallel_execution_message_size

The parallel_execution_message size parameter should be set to a value of 8192 to ensure optimal redo processing on the standby database.

```
parallel_execution_message_size=8192
```

db_recovery_file_dest

The db_recovery_file_dest parameter represents the location of the Oracle Flash Recovery Area, which contains Oracle recovery related files and automates the management of those files based on user specified settings. Automation capabilities simplify the administration and configuration of recovery related files, such as backup sets, image copies, archive log files, flashback log files and more. The Flash Recovery Area can be located on a file system or in an Oracle ASM disk group.

db_recovery_file_dest_size

The db_recovery_file_dest_size parameter indicates the amount of space the Flash Recovery Area is permitted to use. The size needed for the Flash Recovery Area depends on the application. For the Oracle recommended backup strategy, the Flash Recovery Area is generally two and a half times larger than the database.

db_file_name_convert and log_file_name_convert

The db_file_name_convert and log_file_name_convert parameters ensure the database and log file names are matched properly when identical names are not used in a primary control file and a standby control file[2]. These parameter names cannot be set dynamically. As a result, settings must be considered carefully to avoid unnecessary downtime.

fal_server and fal_client

The fal_server and fal_client parameters automate the detection and fetching of log sequence gaps between the primary and the standby instances. For example, gaps can occur if managed recovery is turned off on the standby system while many logs are archived on the primary system.

Testing efforts used the EZConnect syntax for the fal_server and fal_client parameter settings on the temporary local standby system configuration used to convert the production system to Oracle RAC and Oracle ASM. Doing so simplified the TNS settings for the interim configuration. Services for these parameters were defined and referenced for permanent communications between the production and disaster recovery site.

db_unique_name

The db_unique_name parameter specifies a name that differentiates databases running in an Oracle Data Guard standby configuration.

[2] This is generally the case when using Oracle ASM, which uses the db_unqiue_name parameter value as part of the datafile path.

Conventions

The following table lists the conventions used during the testing effort and throughout the steps detailed in this document.

Convention	Meaning
Production or primary system	Initial applications system
applmgr	The user that owns the application file system (APPL_TOP and application tier technology stack)
oracle	The 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 these commands exactly as shown, except as noted.
< >	Text enclosed in angle brackets represents a variable. Substitute a value for the variable text. Do not type the angle brackets.
Application Tier	A tier in the architecture comprised of all nodes in a single site running the Admin, Concurrent Processing, Forms and Web servers.
APPL_TOP	The folder containing applications software.
Shared APPL_TOP	APPL_TOP located on a shared file system. Multiple nodes can access APPL_TOP.
Database Tier	A tier in the architecture consisting of all nodes in a single site running the RDBMS database.
Oracle RAC Database	A database running with Oracle RAC.
Oracle RAC Instance	One instance of an Oracle RAC Database.
ORACLE_HOME	The database and technical stack software location.
ORACLE_SID	The Database Service Identifier.
<node>:$	An operating system level command executed as the oracle or applmgr user on node <node>.
<node>:SQL>	A command executed using SQL*Plus on node <node>.
<node>:RMAN>	A command executed using the RMAN command tool on node <node>.

Prerequisites

While this white paper provides a detailed look at transitioning to MAA, it assumes a detailed understanding of the software components involved. As a result, it is important to gain a thorough understanding of Oracle E-Business Suite deployment, as well as Oracle RAC, Oracle ASM, Oracle Data Guard, and other technologies. In order to implement and manage Oracle RAC, Oracle ASM and a physical standby database, reading and understanding the appropriate documentation is key. Please see the References section for suggested reading materials.

Note — This document assumes the following prerequisites are in place prior to the start of the transformation process.

Operating System

See the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for the Solaris platform located at http://www.oracle.com/pls/db102/homepage for the appropriate operating system versions, kernel settings, and patch levels. For general Solaris Operating System requirements and patches, see the *Solaris Release Notes* located at http://docs.sun.com.

Solaris Cluster Software

Prerequisites for the Solaris Cluster software can be found in the Sun Cluster Software and Hardware Collection for Solaris OS. Specific details on planning, installing, and configuring the Solaris Cluster software are addressed in the *Sun Cluster Software Installation Guide* at http://docs.sun.com.

Shared Storage

This document assumes the appropriate shared storage needed to support Oracle RAC and Oracle ASM configurations is available. In addition, shared storage that meets the physical data requirements of the database must also be available. Oracle RAC and Oracle ASM interact with the storage through the host operating system. As a result, any storage device certified to run on the operating system, operating system driver stack, and Sun Cluster software can be used with Oracle RAC and Oracle ASM. Complete connectivity details are available from Sun.

All storage devices pertaining to the Sun Cluster software, Oracle ASM, Oracle Cluster Ready Services (CRS), and Oracle Database must be accessible by all nodes in the cluster. In order to maintain a highly available architecture, each cluster node must be able to access both controllers of the Sun StorageTek 6540 storage array, which can support clusters with up to four nodes in a direct attach configuration. Clusters with more than four nodes require a storage area network (SAN) to meet storage connectivity requirements. In practice, redundant SANs provide the most fault tolerant architecture for clusters of any size.

Cabling

Sun StorageTek 6540 array cabling on both the drive and host side is very important to the performance of any configuration. The Sun StorageTek 6540 array provides four 2/4 Gigabit host and drive side connections per controller, for a total of eight drive side and eight host side connections. Each drive side Fibre Channel port shares a loop switch with the other disk array controller.

Drive Side Cabling

Fibre channel drive trays include two loops. Drive trays should be configured in multiples of four behind the Sun StorageTek 6540 array. When configuring volumes on the Sun StorageTek 6540 array, distribute the drives equally between the drive trays.

Each Sun StorageTek 6540 array controller uses four Fibre Channel loops. Based on this cabling scheme, the volume groups in drive stacks one and three are dedicated to controller A, and the volume groups in drive stacks two and four are dedicated to controller B. Such a cabling scheme provides both optimal performance and high availability. Should a connection be lost due to failed hardware, an alternate path from the controller to drive tray is available.

Figure 4 shows 16 trays connected to the Sun StorageTek 6540 array. Other enterprises may have smaller storage needs. It is recommended with the XBB that a minimum of four drive trays be connected. These drives trays do not have to be fully populated with drives. However, be sure to split the drives among the four trays to achieve optimal performance.

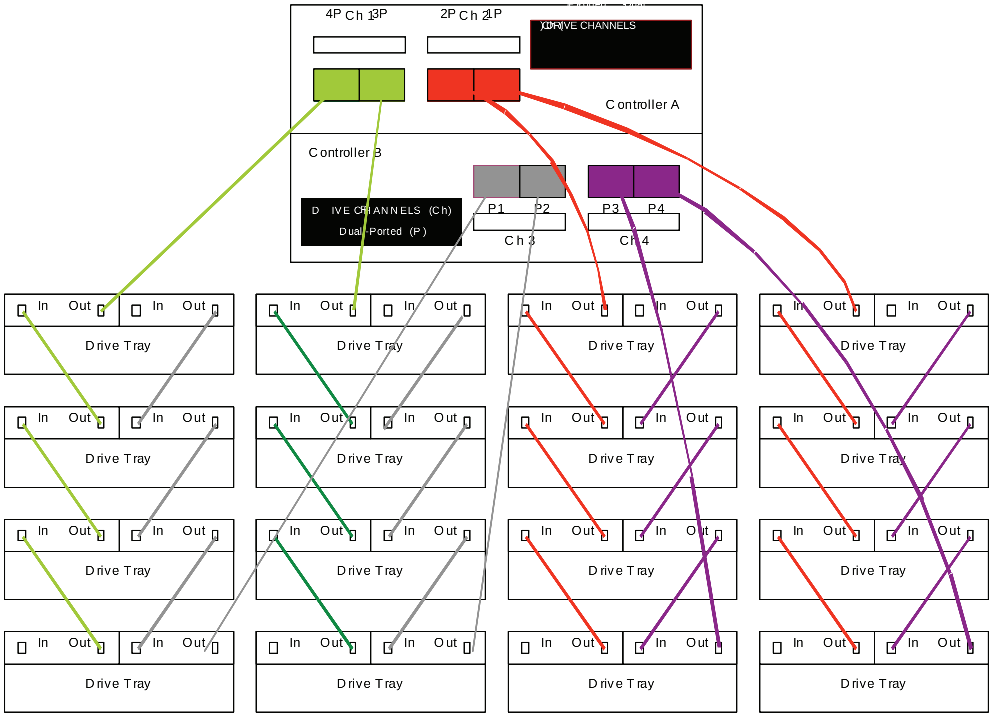


Figure 4. An example drive side cabling scheme

Host to Array Connection

Cabling from the host to the Sun StorageTek 6540 array can be done in one of two ways. The first option is to have the host(s) connected directly to the array. In this configuration, the host should have two host bus adapter ports to connect to each controller in a single Sun StorageTek 6540 array controller module to enable dual paths from the host to the controller. The second option is to connect the hosts and controllers through a Fibre Channel switch. This option enables a single host to connect to more arrays without adding more host adapter ports, and enables multiple hosts to connect to a single array.

Host Cabling for Redundancy

To ensure the storage array remains accessible to the host in the event of a host channel failure, establish two physical paths from each host to the controllers and install alternate path software on the host. When used with alternate path software, such a cabling strategy ensures a redundant path from the host to the controllers.

Host Cabling for Remote Mirroring

If Remote Mirroring is to be used, one of the host ports on each controller must be dedicated for the communication that occurs between the two storage arrays. If the Remote Mirroring feature is not being used, these hosts ports are available for ordinary host connections.

Cabling for Performance

In general, performance is enhanced by maximizing bandwidth — processing more I/O across more channels. A configuration that maximizes the number of host channels and the number of drive channels available to process I/O can maximize performance.

Perform Cluster Verification

Oracle Cluster Verification Utility (CVU) is a single tool that can verify all prerequisites when building a cluster, including operating system patches, shared storage accessibility, user equivalence, and more. Oracle CVU utility should be run at each stage while building a cluster to ensure prerequisites are met. The Oracle 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 the purposes of the testing effort, Oracle E-Business Suite 11.5.10.2 was installed with the following patches and patch sets. The same release and patch level should be installed before beginning the transition to MAA. This document assumes the application is configured with multiple application tier nodes fronted by a hardware load balancer.

- 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)
- Performance improvements for the adautocfg automatic configuration script (4637088)
- CloneContext, which skips DB port check if VALIDATE=NO on application tier (5456078)
- Enable EditContext on the database tier (2873456)
- For a list of other required database patches, refer to Interoperability Notes - Oracle Applications 11i with Oracle Database 10*g* Release 2

Oracle Database

For the testing effort, the initial production database was a single instance Oracle Database 10.2.0.2. No additional patches are required. This document assumes the database is in archive log mode. If this is not the case, schedule time to place the database in archive log mode before the transition is started. Note that this requires some database downtime. In addition, the database should be configured with mirrored online redo logs and control files. Finally, the steps outlined in this paper assume the availability of sufficient network bandwidth between the primary and recovery site to support the Oracle Data Guard software.

Phase 1 — Move the Primary Database to a Single Node Oracle RAC with Oracle ASM

Phase 1 establishes a single node Oracle Database 10*g* Release 2 with Oracle RAC instance on a new database server utilizing the Solaris Cluster software, Oracle Clusterware, and Oracle ASM. Next, the production database is backed up and a Data Guard physical standby and established on the new server. Finally, a switchover causes the new database server to become the production environment. The table below outlines the tasks executed.

Task	Description
1.1	Implement cluster prerequisites
1.2	Establish the Solaris Cluster software on the new database node
1.3	Establish Oracle Clusterware on the new database node
1.4	Establish Oracle ASM on the new database node
1.5	Prepare the existing database for Oracle RAC
1.6	Prepare the existing database for Oracle Data Guard
1.7	Clone the existing database software and prepare the new instance
1.8	Establish the local standby database
1.9	Clone the application software and configure for the switchover
1.10	Switchover and enable Oracle Flashback

Task 1.1 — Implement Cluster Prerequisites

Use the appropriate platform specific commands to implement cluster prerequisites, such as establishing the shared storage on the new database node and applying operating system patches on the new node. Sufficient shared storage is needed for the Oracle ASM data and flash recovery areas for the data volume. In addition, small storage locations are needed for the Oracle Clusterware registry, Oracle Clusterware vote disk, and the Oracle ASM SPFILE. The Solaris Cluster software requires a local file system approximately 512 MB in size on each node for use by the global device subsystem. Furthermore, quorum devices should be configured for the Solaris Cluster software, and can be in the form of very small shared devices or LUNs. Finally, use the appropriate operating system or storage system utilities to replicate the database Oracle home to the new database node and update the Oracle home on the new Oracle RAC database node.

The following table summarizes the sizes used during testing efforts. It is essential that redundancy be configured for all storage locations.

Purpose	Size
+DATA ASM Diskgroup	500 GB
+FLASH ASM Diskgroup	1250 GB
Oracle Clusterware Registry	1 GB
Oracle Clusterware Vote Disk	1 GB
ASM SPFILE	1 GB
Solaris Cluster quorum device (shared storage)	200 MB
Solaris Cluster global device file system (local disk)	512 MB

Oracle Clusterware release for Oracle Database 10*g* Release 2 provides the capability for Oracle Cluster Registry (OCR) and Voting Disk redundancy, if external redundancy is not used. The installer offers a default mirroring configuration of two OCR devices and three Voting Disk devices.

Oracle ASM can provide disk group redundancy, if external redundancy is not used. Both NORMAL (two-way) and HIGH (three-way) redundancy options are available. If a high-end storage subsystem is used that offers redundancy, then redundancy should be deferred to the subsystem and Oracle ASM redundancy should not be used. Users can configure disk groups with the EXTERNAL redundancy attribute.

Use of External Redundancy through RAID

External based redundancy can be used through RAID, if available, instead of ASM redundancy. This can be done by issuing the EXTERNAL REDUNDANCY clause when creating an ASM disk group. The storage array is then responsible for protecting the underlying database through RAID, and ASM is responsible for the management and redistribution of data files under disk failures and for I/O performance.

Choice of RAID Level

The optimal RAID level chosen depends primarily on the application mix or workload placed on it. Once a workload type is determined an appropriate RAID level can be selected. Understanding the application I/O characteristics is critical as the following factors determine the RAID level and the number of disks in each volume group:

- Is the I/O primarily sequential or random?
- What size is a typical I/O request: large (>256KB), small (<64KB), or in-between?
- What is the I/O mix (number of reads versus writes)?
- What type of I/O does the application use: buffered or unbuffered?
- Is concurrent I/O or multiple I/O threads used? In general, creating more sustained I/O produces the best overall results up to the point of controller saturation. Write-intensive workloads are an exception.

In general, RAID 5 works best for sequential large I/O (>256KB), while RAID 5 or RAID 1 works best for small I/Os (<32KB). For I/O sizes in between, the RAID level is dictated by other application characteristics. RAID 5 and RAID1 have similar characteristics for read environments and for sequential writes. RAID 5 is challenged most by random writes.

For high bandwidth applications, a RAID 5 8+1 volume group is recommended. For high I/O rate applications, use RAID 5 (4+1 or 8+1) or RAID 1 volume groups. These are guidelines — cost and capacity considerations may justify variations, and the Dynamic Capacity Expansion (DCE) feature makes adding a disk to a volume group an easy process

Task 1.2 — Establish Solaris Cluster Software on the New Database Node

Deploying Oracle Clusterware with native cluster solutions, such as the Solaris Cluster software is an optional and fully supported configuration. Through core features, including low-level heartbeat implementations, robust voting algorithms and SCSI reservation ioctls, the kernel-based Solaris Cluster software is extremely robust and rigorous in determining cluster membership. In this environment, Oracle clusterware leverages Solaris Cluster software to provide cluster membership management. In order to implement an Oracle RAC architecture with both Oracle and Solaris Cluster software, the Solaris Cluster software would be installed, configured, and operational prior to the deployment of Oracle Clusterware.

Step 1.2.1 — Install and Configure the Solaris Cluster Software

This step describes the installation and configuration of the Solaris Cluster software on one node that is to be configured as the first node in a two-node cluster. For a quick reference, details on the procedures to accomplish this step are documented in Appendix A. The Sun Cluster Software Installation Guide for Solaris OS should be used as a supplemental guide and full reference.

- Install the Solaris Cluster binary from CDROM or downloaded image by running the `installer`. (See Appendix A.1.1). Select the following components: Sun Cluster 3.X, Sun Cluster Agents — HA for Oracle and Sun Cluster Support for Oracle RAC, and All Shared Components.

- Execute the `/usr/cluster/bin/clsetup` command to configure the Solaris Cluster software with one initial node. See Appendix A.1.2 for details.

- Use the `clsetup` command to configure resource group and data service support for Oracle RAC. See Appendix A.1.3 for details.

Step 1.2.2 — Apply Solaris Core Patches

If applicable, download and apply the latest Solaris Cluster Core patch from http://sunsolve.sun.com. Follow the instructions provided for applying the patch included in the patch shipment.

Task 1.3 — Establish Oracle Clusterware on the New Database Node

Oracle Clusterware provides critical cluster management features, including node membership and resource management. The following steps explain how to install Oracle Clusterware.

Step 1.3.1 — Install Oracle Clusterware for Oracle Database 10g Release 2

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for the Solaris platform to install Oracle Clusterware on the new database server. The guide can be found at http://www.oracle.com/pls/db102/homepage.

Oracle Clusterware should be installed in an Oracle home that is separate from Oracle Database and Oracle ASM homes. When installing Oracle Clusterware in a Solaris Cluster enabled environment, the only private interconnect to be specified is `clprivnet0`, which is the logical interface created by the Solaris Cluster software. Using this interface ensures the redundancy and failover of physical interconnects, as well as the striping of interconnect traffic. Physical interconnect interfaces, such as `ibd3` and `ibd5`, should be indicated as `Do Not Use` when installing and configuring CRS.

Step 1.3.2 — Apply Patchset 10.2.0.2 or Later

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

Task 1.4 — Establish Oracle ASM on the New Database Node

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 when it is not implemented in the backend storage subsystem. Oracle ASM also includes the ability to add and remove storage with no downtime, as data is automatically transferred from the disks being removed to remaining disks in the system.

The following steps explain how to install and configure Oracle ASM. The best practice is to create two disk groups, one for database files and one for flash recovery area files. Mirrored copies of the redo logs and control files should be stored in both the flash recovery and the database file disk groups. Detailed Oracle ASM administration concepts and commands can be found in the Oracle Database 10g Release 2 Administrator's Guide.

Step 1.4.1 — Install Oracle Database 10g Release 2 for Oracle ASM

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for the platform to create a clustered Oracle ASM installation on the new database node. An Oracle home that is separate from Oracle Database and Oracle Clusterware homes should be used for the installation.

Step 1.4.2 — Apply Patchset 10.2.0.2 or Later

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

Step 1.4.3 — Create the Oracle ASM Instance and Disk Groups

Use the DBCA utility to create an Oracle ASM instance and configure the disk groups. Be sure to create data and flash recovery disk groups, as documented in the guide.

Step 1.4.4 — Remove the Listener Created by the DBCA Utility

The DBCA utility creates a listener and registers it to CRS during the Oracle ASM configuration process. If the listener uses the same port or name as the Oracle E-Business Suite, it must be removed with the `netca` command.

Task 1.5 — Prepare the Existing Database for Oracle RAC

The following steps explain how to prepare the existing database for Oracle RAC.

Step 1.5.1 — Create Additional Redo Logs

Each Oracle RAC instance requires its own redo thread. A single instance database has only one redo thread by default. Additional threads must be added and enabled. The following example shows how to create addition redo threads.

```
ha1db: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;
alter database enable public thread 2;
```

Step 1.5.2 — Create Additional Undo Tablespaces

Each Oracle RAC instance requires its own undo tablespace. A single instance database has only one undo tablespace by default. Additional tablespaces must be added. The follow example shows how to create additional undo tablespaces.

```
ha1db: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 1.5.3 — Execute CATCLUST.SQL

The `catclust.sql` script provided by Oracle must be executed to prepare a database for Oracle RAC operation as follows.

```
ha1db:SQL> @$ORACLE_HOME/rdbms/admin/catclust.sql
```

Task 1.6 — Prepare the Existing Database for Oracle Data Guard

The following steps explain how to prepare the existing database for Oracle Data Guard.

TIP

If planning to operate the original database as an Oracle Data Guard primary before beginning the switchover process, apply Oracle Data Guard parameters for primary operation immediately to keep downtime to a minimum.

Step 1.6.1 — Create Password Files

In Oracle Database 10*g*, Oracle Data Guard requires the use of a password file for communication 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 find pointers to the commands needed for a given platform. The following example applies to the UNIX platform. Repeat this procedure for the current SID (VIS) and the new SIDs (VIS1 and VIS2) to be used for the Oracle RAC environment. Doing so ensures the information is copied over and used as the Oracle homes are cloned.

```
ha1db:$ cd $ORACLE_HOME/dbs
ha1db:$ orapwd file=orapw<SID> password=<SYS’s password>
```

For the password file to function correctly, the following parameter setting must be configured in the database instance. This is the default setting in Oracle Database 10*g* Release 2. Because this value is not overridden by the standard Applications database configuration files, it is not listed in the database configuration parameters elsewhere in this document.

```
Remote_login_passwordfile=EXCLUSIVE
```

Step 1.6.2 — Configure Database Instance Parameters

Configure the original database instance to perform as an Oracle Data Guard primary so that it is ready for the switchover to the new Oracle RAC instance. Set the parameters in the *CONTEXT_NAME*_ifile.ora parameter include file, as well as the running instance. The following parameters were used in the test environment. Note that EZConnect syntax is used to define the service for redo log transport to the temporary local standby being created. Doing so simplifies the overall network configuration tasks for this phase.

```
log_archive_config='dg_config=(VIS, VIS_ha1)'
log_archive_dest_2='SERVICE=ha1db01:1521/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=ha1db01:1521/VIS1
    valid_for=(online_logfiles,primary_role) db_unique_name=VIS_ha1 LGWR ASYNC=20480 OP-
    TIONAL REOPEN=15 NET_TIMEOUT=30';
SQL> alter system set log_archive_dest_state_2=defer;
```

Step 1.6.3 — Enable Forced Logging

Oracle E-Business Suite sometimes uses the `NOLOGGING` feature, which can cause certain changes not to be populated to the standby database and invalidate the standby database. As a result, `force logging` must be turned on in the database by issuing the following command from SQL*Plus connected as `sysdba`. See Oracle*MetaLink* note 216211.1 for more information.

```
ha1db:SQL> alter database force logging;
```

Step 1.6.4 — 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 the test environment uses two online redo logs for each thread, three standby redo logs are required for each thread. Use the formula below to calculate the number of standby redo logs.

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

As the `ORACLE` user on the production database server, create standby redo log groups by connecting to SQL*Plus as the `sysdba` user and issue the following command for each standby redo log group to create. See the “Configure a Standby Redo Log” section in the Oracle Data Guard Concepts and Administration Guide for more information.

```
ha1db:SQL> alter database add standby logfile
thread N < group N > (
‘<fully qualified logfile member name>’,
‘<fully qualified logfile member name>’)
size NNN;
```

The following command can be used to query the V$LOGFILE view and see the members created.

```
ha1db:select * from v$logfile;
ha1db:select * from v$standby_log;
```

Step 1.6.5 — 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 SQL*Net Access security is enabled in the existing system (by default from Oracle 11i10), all the database nodes in the current Oracle RAC and disaster recovery sites must be given access for correct database operation. Pay particular attention to nodes with multiple network interfaces and make sure the appropriate node aliases are included. See the “Managed SQL*Net Access from Hosts” section in document 281758.1 Oracle*MetaLink* 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 application tiers, their network aliases must be granted access manually. It is not necessary to include the network interface used exclusively for cluster interconnect communications in an Oracle RAC configuration.

Task 1.7 — Clone Existing Database Software and Prepare the New Instance

The following steps explain how to clone the existing database software and prepare the new instance.

Step 1.7.1 — Prepare the Database Oracle Home for Cloning

Run the following commands on the production database server as the ORACLE user. Supply the `APPS` password when requested.

```
ha1db:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
ha1db:$ perl adpreclone.pl dbTier
```

Step 1.7.2 — Backup, Ship, and Restore the Database Oracle Home

Using the appropriate operating system utilities, create a backup of the Oracle home, ship the backups to the new database server, and restore the Oracle home to the new Oracle RAC database node.
Note that this document assumes the new node is not the same as the current production database node. Using the existing node can result in conflicting database listener configurations during the switchover. See the “Add the Original Node as a Secondary Oracle RAC Instance in the Oracle RAC Cluster” section for details on how to add the original database node to the cluster after the switchover.

Note — A different `ORACLE_HOME` location must be used for Oracle RAC instances to prevent a conflict when the original node is reintroduced to the configuration.

Step 1.7.3 — Configure the New Database Oracle Home

Execute the following process for the new Oracle home on the new clustered database server for the first site.

```
ha1db01:$ cd $ORACLE_HOME/appsutil/clone/bin
ha1db01:$ perl adcfgclone.pl dbTechStack
```

The following table details the proper responses to prompts.

Prompt	Response
Do you want to use a virtual hostname for the target node (y/n) ← *This is not the same as a RAC "virtual host"*	n
Target instance is a Real Application Cluster (Oracle RAC) instance (y/n)	y
Current node is the first node in an N Node Oracle RAC Cluster (y/n)	y
Number of instances in the Oracle RAC Cluster← *Total Number, although not all of them available for time being*	2
Target System database name ← *Not the global_db_name*	VIS
Do you want to preserve the port values from the source system on the target system	y
Provide information for node 1 (current node):	
Host name	ha1db01
Virtual host name	ha1db01-vip
Instance number ← *Instance number for this node*	1
Provide information for node 2 (registered now but established later)	
Host name	ha1db02
Virtual host name	ha1db02-vip
Instance number	2
Target system RDBMS ORACLE_HOME directory	/u01/app/oracle/ visdbRAC/10.2.0
Target system utl_file accessible directories list	/usr/tmp
Number of DATA_TOP's on the target system	1
Target system DATA_TOP 1 ← *In ASM now*	+DATA/vis_ha1
Display	ha1db01:0.1

The `adcfgclone` script creates a new environment file that should be sourced before using the environment. For example:

```
ha1db01:$ cd $ORACLE_HOME

ha1db01:$ . ./VIS1_ha1db01.env
```

Step 1.7.4 — Configure and Restart the Standby Listener

The listener service definition generated by the `adcfgclone` script points to the standby database but does not include all required interface names. This situation can be corrected by creating a listener include file named `listener_ifile.ora` in the `$TNS_ADMIN` directory on the standby server. The file contents should be similar to the following:

```
VIS1=
ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP)(Host=
   ha1db01.ha.us.oracle.com)(Port=1521))
 (ADDRESS=(PROTOCOL=TCP)(Host=
   ha1db01-vip.ha.us.oracle.com)(Port=1521))
)
```

Stop and restart the listener on the new standby database node using the following commands.

```
ha1db01:$ lsnrctl stop VIS1

ha1db01:$ lsnrctl start VIS1
```

Step 1.7.5 — Configure New Database Instance Parameters

Configure the new database instance to perform as an Oracle Data Guard standby so that the instance is ready for the switchover. The instance must also be configured for correct RMAN operation when restoring to the new Oracle ASM disk groups. If the system is to be configured for parallel concurrent processing (PCP), set two parameters now to avoid a database bounce later.

Example parameters for Oracle Data Guard for standby operation include:

```
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=ha1db:1521/VIS

fal_client=ha1db01:1521/VIS1

standby_archive_dest=
     'LOCATION=USE_DB_RECOVERY_FILE_DEST’

standby_file_management=AUTO
parallel_execution_message_size=8192
```

Example file management parameters used by RMAN include:

```
db_create_file_dest='+DATA'

db_recovery_file_dest='+FLASH'

db_recovery_file_dest_size=1228800M
control_files=
+DATA/VIS_HA1/CONTROLFILE/control01.ctl
```

Example PCP parameters include:

```
lm_global_posts=TRUE
_immediate_commit_propagation=TRUE
```

Task 1.8 — Establish the Local Standby Database

The following steps can be used to establish the local standby database.

Step 1.8.1 — Backup the Production Database Using RMAN (Recovery Manager)

RMAN should be used to backup the current production database so that it is ready to be restored to the new standby. Be sure to backup the database, archived redo logs, and the control file. The backup should be written to media, disk or tape to facilitate replicating the backup to the standby location. The following example shows how to backup the production database using a disk backup.

```
ha1db:$ rman target /

ha1db:RMAN> backup device type disk format '/NAS/oracle/rman_backups/%U' database plus
archivelog;

ha1db:RMAN> backup device type disk format '/NAS/oracle/rman_backups/%U' current con-
trolfile for standby;
```

Step 1.8.2 — Ship the Backup to the New Database Server

Use the appropriate system tools to ship the backup to the new database server. Best practices include storage system remote replication of the disk backup (available on the Sun StorageTek 5320 NAS Appliance and Sun StorageTek 6540 array) when network bandwidth is available, or physically transporting tape cartridges or relocating the disk storage device containing the backup when network bandwidth is insufficient.

Step 1.8.3 — Start One Oracle RAC Instance

Start the Oracle RAC instance with the `NOMOUNT` option on the new clustered server for the first site.

```
ha1db01:SQL> startup nomount
```

Step 1.8.4 — Restore the Database Backup to the Standby Using RMAN

The database must be restored with the `for standby` option. For example:

```
ha1db01:$ rman target sys/manager@ha1db:1521/VIS auxiliary /

ha1db01:RMAN> duplicate target database for standby;
```

Step 1.8.5 — Start Managed Recovery

On the standby database, check to make sure the database is in standby mode by executing the following query from SQL*Plus connected as the `sysdba` user.

```
ha1db01:SQL> select DATABASE_ROLE from v$database;

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

On the primary database, enable the previously deferred remote destination by executing the following command from SQL*Plus connected as the `sysdba` user.

```
ha1db:SQL> alter system set log_archive_dest_state_2=enable;
```

On the primary database server, update the database configuration in the *`context_name`*`_ifile.ora` database parameter file.

```
log_archive_dest_state_2=enable
```

Place the standby database in managed recovery by executing the following command.

```
ha1db01:SQL> recover managed standby database disconnect;
```

Step 1.8.6 — Verify Correct Standby Operation

Validate that the standby database is correctly applying redo operations from the primary database. On the primary database, archive the current log using the following statement.

```
ha1db:SQL> alter system archive log current;
```

On the standby database, query the `gv$archived_log` view to verify the logs are received and applied.

```
ha1db01: 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 1.9 — Clone the Application Software and Configure for Switchover

The existing production application tiers are reused to access the new, single node Oracle RAC, Oracle Flashback, and Oracle ASM-enabled database. To avoid risk to production operations, and to get as much configuration work done ahead of time as possible, the application tier software is cloned to a new location on the application tiers.

Step 1.9.1 — Prepare the Application Tier for Cloning

For each type of application tier installed at the site, such Concurrent Processing, Administration, Web, or Forms, log on as the APPLMGR user and prepare the application tier for cloning. If a single common APPL_TOP is shared, or only full installs are available, the following need only be done on one application tier. For example:

```
ha1mt01:$ cd <COMMON_TOP>/admin/scripts/<CONTEXT_NAME>
ha1mt01:$ perl adpreclone.pl appsTier
```

Step 1.9.2 — Copy the Application Tier File System

Use the appropriate tool to make a copy of the application tier files in a new directory location on the existing application tiers. This copy of the software is used to run the application after the switchover.

Step 1.9.3 — Configure the New Application Tier File System

Configure the new application tier file system with the adclonectx.pl script using the original context file, or a copy of it, as the source context file. The script must be run on each application tier node. In the example for node ha1mt01 below, the new mount point is /u01/appltopRAC and points to the original context file located at 7.

```
ha1mt01:$ cd /u01/appltopRAC/viscomn/clone/bin
ha1mt01:$ perl adclonectx.pl contextfile= /u01/appltop/visappl/admin/<CONTEXT_NAME>.xml
```

Provide the values required for the creation of the new APPL_TOP Context file, as follows. Repeat the procedure for all application tier nodes.

Prompt	Response
Do you want to use a virtual hostname for the target node (y/n)	n
Target hostname	ha1mt01
Do you want the inputs to be validated (y/n) ← *The new standby database is not yet available, so the inputs cannot be validated*	n
Target system database SID	VIS
Username for the application's file system owner	applmgr
Group for the application's file system owner	dba
Target system database server node	ha1db01
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

Prompt	Response
Instance number	2
Target system concurrent processing node	ha1mt01
Target system administration node	ha1mt01
Target system forms server node	ha1mt01
Target system web server node	ha1mt01
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 ha1db01:0.0 (y/n)	n
Target system Display	ha1mt01:0.0
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/appltopRAC/visappl/ admin/VIS_ha1mt01.xml

Step 1.9.4 — Complete the New Application Tier File System Configuration

Complete the application tier file system configuration with the `adcfgclone.pl` script using the context file generated in the previous step. The script must be run on each application tier node. Following is an example for node `ha1mt01`. Since the database is not available at this time, ignore the error related to the failed `AutoConfig` run.

```
ha1mt01:$ export APPL_TOP=<APPL_TOP>/visappl
    (Workaround for known issue, Bug 5520384)

ha1mt01:$ cd /u01/appltopRAC/viscomn/clone/bin

ha1mt01:$ perl adcfgclone.pl appsTier <APPL_TOP>/visappl/admin/VIS_ha1mt01.xml
```

Task 1.10 — Switchover and Enable Oracle Flashback

The following steps can be used to switch over to the local standby database. The Oracle E-Business Suite application does not run during this stage. To keep downtime to a minimum, it is important to ensure steps are carefully practiced and scripted.

Step 1.10.1 — Prepare the Standby Database

If the standby database 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 1.10.2 — Shutdown Oracle E-Business Suite

Ensure the Oracle E-Business Suite application is shutdown completely.

Step 1.10.3 — Switchover to the Local Standby Database

Commit the primary database to switchover to standby, shutdown the primary database, and stop the primary database listener.

```
ha1db:SQL> alter database commit to switchover to standby with session shutdown;

ha1db:SQL> shutdown immediate

ha1db:$ lsnrctl stop VIS
```

Verify that the standby database is ready to be converted to be the new primary database.

```
ha1db01:SQL> select switchover_status from v$database;

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

Execute the following command on the standby database to convert it to be the new primary database.

```
ha1db01:SQL> alter database commit to switchover to primary;
```

Step 1.10.4 — Enable Oracle Flashback

Enable the flashback features on the new production server and run a query to check flashback status.

```
ha1db01:SQL> alter database flashback on;

ha1db01:SQL> select flashback_on from v$database;

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

Step 1.10.5 — Open the Database

Use the following command to open the database.

```
Ha1db01:SQL> alter database open;
```

Step 1.10.6 — Remove the Old Application Topology

Connect to the new production database using SQL*Plus as the `APPS` and execute the following commands, substituting the appropriate database name.

```
ha1db01:SQL> exec FND_NET_SERVICES.remove_system('VIS');
ha1db01:SQL> commit;
```

Step 1.10.7 — Configure the Database Tier

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

```
ha1db01:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
ha1db01:$ ./adautocfg.sh
```

Step 1.10.8 — Restart the Listeners

Next, `AutoConfig` starts the listener using the non-RAC naming convention (`SID`) as the listener name when the database is not available. As a result, it is important to start the correct listener on the new production database node using the Oracle RAC naming convention (`LISTENER_`*hostname*).

```
ha1db01:$ lsnrctl stop VIS1
ha1db01:$ lsnrctl start LISTENER_ha1db01
```

Step 1.10.9 — Configure the Application Tiers

Run `AutoConfig` using the new `APPL_TOP` on all application tier nodes. Note that these steps can be run in parallel. Following is an example for `ha1mt01`.

```
ha1mt01:$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
ha1mt01:$ ./adautocfg.sh
```

Step 1.10.10 — Start Up Oracle E-Business Suite

Startup the Oracle E-Business Suite software on all application tier hosts. Online users can gain access at this time.

TIP

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

Phase 2 — Add the Original Node as a Secondary Oracle RAC Instance to the Oracle RAC Cluster

Once the application is stabilized on the new database server, the original application can be taken out of service. Next, the cluster related hardware and software is installed on the original server, including Oracle Clusterware, Oracle ASM, and Oracle RAC. The server is then added to the cluster to create a two node Oracle RAC cluster with Oracle ASM. The following tasks can be used to create the cluster:

- Implement cluster prerequisites
- Establish the Solaris Cluster
- Establish Oracle Clusterware and Oracle ASM
- Clone the database software and prepare the new instance
- Configure the application tier for Oracle RAC
- Update the Oracle Clusterware configuration

In the example that follows, `ha1db` is renamed to `ha1db02` in order to create a more logical node name.

Task 2.1 — Implement Cluster Prerequisites

Using the appropriate platform-specific commands, implement cluster prerequisites such as establishing the shared storage on the new database node, applying operating system patches on the new node, and more.

Task 2.2 — Establish the Solaris Cluster Software

Install and configure the Solaris Cluster software on the second (original) node prior to establishing Oracle Clusterware and Oracle RAC. Previously, the Solaris Cluster software was configured and enabled on the first node of an intended two-node cluster. Up to this point, an operating cluster exists that contains one node. The next task is to add the second node and make it an active member of the cluster.

Step 2.2.1 — Install and Configure the Solaris Cluster Software

This step installs and configures the Solaris Cluster on the second node, and adds the node to the active cluster through the following:

- Install the Solaris Cluster binary from CDROM or downloaded image.
- Select the appropriate components (Appendix A.2.1)
- Configure the Solaris Cluster software
- Add the node to the existing cluster (Appendix A.2.2)
- Configure resource group and data service support for Oracle RAC (Appendix A.2.3)
- Identify and create quorum devices (Appendix A.2.4)

More information on this step can be found Appendix A.2. The *Sun Cluster Software Installation Guide for Solaris OS* can be used as a supplemental guide and full reference.

Step 2.2.2 — Apply Solaris Cluster Core Patches

If applicable, download and apply the latest Solaris Cluster Core patch from http://sunsolve.sun.com. Follow the instructions for applying patches included in the patch shipment.

Task 2.3 — Establish Oracle Clusterware and Oracle 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:

- Establish Oracle Clusterware
- Add the node to the cluster at the Oracle Clusterware layer
- Establish Oracle ASM
- Add a new Oracle ASM instance
- Ensure Oracle Clusterware and Oracle ASM are up and running

Task 2.4 — Clone the Database Software and Prepare the New Instance

The following steps can be used to clone the database software and prepare the new instance of the database.

Step 2.4.1 — Prepare the Oracle RAC Oracle Home for Cloning

Log on to the Oracle RAC database server as the ORACLE user and run the following commands. Supply the APPS password when requested.

```
ha1db02:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
ha1db02:$ perl adpreclone.pl dbTier
```

Step 2.4.2 — 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 original database node.

Step 2.4.3 — Configure the Database Oracle Home

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

```
ha1db02:$ cd $ORACLE_HOME/appsutil/clone/bin
ha1db02:$ perl adcfgclone.pl dbTechStack
```

Respond the prompts as indicated in the following table.

Prompt	Response
Do you want to use a virtual hostname for the target node (y/n)	n
Target instance is a Real Application Cluster (RAC) instance	y
Current node is the first node in an N Node Oracle RAC Cluster	N
Please provide the details to connect to one of the live Oracle RAC nodes	
Host name of the live Oracle RAC node	ha1db01
Domain name of the live Oracle RAC node	ha.us.oracle.com
Database SID of the live Oracle RAC node	VIS1

Prompt	Response
Listener port number of the live Oracle RAC node	1521
Current host is not registered, do you want to add this node to the Oracle RAC cluster (y/n)	y
Number of new instances to be added in the Oracle RAC Cluster	1
Provide information for the Node 2 (current node)	
Host name	ha1db02
Instance number	2
Private interconnect name	ha1db02-ci
Current Node	
Host name	ha1db02
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_TOPs on the target system	1
Target system DATA_TOP 1	+DATA/vis_ha1
Do you want to preserve the Display set to ha1db01:0.1	y

The `adcfgclone` script creates a new environment file, which should be sourced before using the environment. Source the file using the following commands.

```
ha1db02:$ cd $ORACLE_HOME
ha1db02:$ . ./VIS2_ha1db02.env
```

Step 2.4.4 — Configure Database Instance Parameters

On the original database server in the new database Oracle home, add the following parameters to the database configuration via the include file found at `$ORACLE_HOME/dbs/`*CONTEXT_NAME*`_ifile.ora`. Keeping changes in a separate include file gives administrators the freedom to recreate the base database parameter files at any time using `AutoConfig`, without erasing customizations. Note that the following steps continue to refer to the single instance node with the data guard parameters because these parameters must be identical across Oracle RAC instances.

The Oracle Data Guard parameters for standby operation are listed below.

```
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=ha1db:1521/VIS

fal_client=ha1db01:1521/VIS1

standby_archive_dest=
    'LOCATION=USE_DB_RECOVERY_FILE_DEST’

standby_file_management=AUTO
parallel_execution_message_size=8192
```

The file management parameters used by RMAN are listed below.

```
db_create_file_dest='+DATA'

db_recovery_file_dest='+FLASH'

db_recovery_file_dest_size=1228800M
control_files= +DATA/VIS_HA1/CONTROLFILE/control01.ctl
```

Optional PCP parameters are listed below.

```
_lm_global_posts=TRUE
_immediate_commit_propagation=TRUE
```

Step 2.4.5 — Start the New Instance

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

```
ha1db02:SQL> startup
```

Step 2.4.6 — Run AutoConfig on the New Node

The `adcfgclone` script does not request the virtual hostname. This value must be added manually by editing the `s_virtual_hostname context` variable in the `$ORACLE_HOME/appsutil/`*CONTEXT_NAME*`.xml` context file to include the value for the virtual host. During testing efforts, the host name setting was changed from `ha1db02` to `ha1db02-vip`. Next, run the following command to configure the new node for use by Oracle E-Business Suite.

```
ha1db02:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>

ha1db02:$ ./adautocfg.sh
```

Step 2.4.7 — Restart the Listener

Use the following commands to start the correct listener for the new instance on the original database server.

```
ha1db02:$ lsnrctl stop VIS2

ha1db02:$ lsnrctl start LISTENER_ha1db02
```

Step 2.4.8 — Run AutoConfig on the Live Database Nodes

Run the following commands on the first database node to generate a TNS configuration that includes the recently added Oracle RAC instance. This step can be performed while the system is up.

```
ha1db01:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>

ha1db01:$ ./adautocfg.sh
```

Task 2.5 — Configure the Application Tiers for Oracle RAC

At this point, the original database server is now running the second Oracle RAC instance. However, the application tiers are still accessing the first Oracle RAC instance on the new database server. To complete the configuration, AutoConfig must be run on the application tiers, and the application processes must be bounced. To reduce the effect of this outage, consider controlling the application tier bouncing by starving them in turn via the hardware load balancer, and bouncing each one when idle.

Step 2.5.1 — Modify Context Variables for Load Balancing

Several context variables may need to be set for load balancing functionality to be employed.

- The `adcfgclone` script does not set variables to use the load balanced services. To adjust these parameters manually, run the `Context Editor` through the 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` (`s_apps_jdbc_connect_alias`). Repeat these 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 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 multiple Concurrent Processing Nodes are in use, and the use of PCP is desired, refer to the *Configure Parallel Concurrent Processing* section of MetaLink Note 362135.1.

Step 2.5.2 — Run AutoConfig

Run `AutoConfig` on all application tier nodes. The following is an example for how to run `AutoConfig` on node `ha1mt01`.

```
ha1mt01:$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>

ha1mt01:$ ./adautocfg.sh
```

Step 2.5.3 — Restart Oracle E-Business Suite

Source the new environment file and restart the Oracle E-Business Suite processes on all application tier nodes.

Step 2.5.4 — Configure Parallel Concurrent Processing

If multiple Concurrent Processing Nodes are in use, and the use of PCP is desired, refer to the *Configure Parallel Concurrent Processing* section of MetaLink Note 362135.1 to complete PCP configuration.

Task 2.6 — Update the Oracle Clusterware Configuration

In order for `srvctl` to be used to control resources, the new resources must be added to CRS.

Step 2.6.1 — Add the Database to the Oracle Clusterware Configuration

Add the database to CRS using the following command.

```
ha1db01:srvctl add database -d VIS -o $ORACLE_HOME
```

Step 2.6.2 — Add Instances to the Oracle Clusterware Configuration

Add the instances to CRS using the following sequence of commands.

```
ha1db01:srvctl add instance -d VIS -i VIS1 -n ha1db01

ha1db01:srvctl add instance -d VIS -i VIS2 -n ha1db02

ha1db01:srvctl setenv instance -d VIS -i VIS1
    -t TNS_ADMIN=$ORACLE_HOME/network/admin/VIS1_ha1db01

ha1db01:srvctl setenv instance -d VIS -i VIS2
    -t TNS_ADMIN=$ORACLE_HOME/network/admin/VIS2_ha1db02

ha1db01:srvctl modify instance -d VIS -i VIS1 -s +ASM1

ha1db01:srvctl modify instance -d VIS -i VIS2 -s +ASM2
```

Step 2.6.3 — Add Listeners to the Oracle Clusterware Configuration

Run `NETCA` to add the listeners to CRS as outlined in the steps below.

- Edit `$ORACLE_HOME/bin/racgwrap` and add the following commands:

```
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 at all times.
- Run `netca` and choose `Cluster Configuration` and only local node.

- Add or reconfigure the `LISTENER` listener that is listening on the database port. Ignore the errors `Listener is already running` and `Port already in use`, if displayed.
- Run `AutoConfig` to overwrite the `listener.ora` file generated by `netca`.
- Repeat these steps on all nodes, since each node uses different `$TNS_ADMIN` settings.

Step 2.6.4 — Verify the CRS Setup

Verify the resources were added correctly using following command.

```
ha1db01:$ $CRS_HOME/bin/crs_stat
```

Phase 3 — Establish the Disaster Recovery Site with Oracle Data Guard

To complete the MAA, the disaster recovery site must be established in case the primary site is lost. Toward this end, a two-node Oracle Database 10*g* Release 2 with Oracle RAC database cluster is created on database servers at the disaster recovery site utilizing Oracle Clusterware and Oracle ASM. In addition, the production database is backed up, an Oracle Data Guard physical standby is established on the new server, and the Oracle E-Business Suite application tier software is cloned from the primary site to the disaster site. The standby system constantly apply redos from the primary site to ensure it stays up to date and is ready to take over in the event of an emergency. Steps explaining switchover and failover are made available, in the event they prove necessary.

The tasks required to establish the disaster site include:

- Implement cluster prerequisites
- Establish the Solaris Cluster software
- Establish Oracle Clusterware
- Establish Oracle ASM
- Prepare the existing database for Oracle Data Guard
- Clone the database software and prepare the new instances
- Establish the standby database
- Clone the application software and configure it on the disaster recovery site
- Update the Oracle Clusterware configuration

Task 3.1 — Implement Cluster Prerequisites

Using the appropriate platform-specific commands, implement cluster prerequisites such as establishing the shared storage on the new database node, applying operating system patches on the new node, and more. Note that shared storage is needed for the Oracle ASM data and flash recovery areas that is sufficient for the data volume. Small storage locations are needed for the Oracle Clusterware registry, Oracle Clusterware vote disk, and the Oracle ASM SPFILE. In addition, the Solaris Cluster software requires a local file system (approximately 512 MB) to be created on each node for use by the global device subsystem. Quorum devices should be configured for the Solaris Cluster software, and can take the form of very small shared devices or LUNs. The following table summarizes the sizes used during testing efforts.

Purpose	Size
+DATA ASM Diskgroup	500 GB
+FLASH ASM Diskgroup	1250 GB
Oracle Clusterware Registry	1 GB
Oracle Clusterware Vote Disk	1 GB
Oracle ASM SPFILE	1 GB
Solaris Cluster Quorum Device (Shared Storage)	200 MB
Solaris Cluster Global Device File System (Local Disk)	512 MB

Task 3.2 — Establish the Solaris Cluster Software

The Solaris Cluster software must be installed and configured on the two nodes of the disaster recovery site prior to establishing Oracle Clusterware and Oracle RAC. The following steps outline the procedure for installing and configuring the Solaris Cluster software.

Step 3.2.1 — Install and Configure the Solaris Cluster Software

Installing and configuring the Solaris Cluster software on the disaster recovery site are the same as those performed at the primary site. In particular:

- Install the Solaris Cluster binary from CDROM or downloaded image.
- Select the appropriate components (Appendix A.3.1)
- Configure the Solaris Cluster software to form a two-node cluster (Appendix A.3.2)
- Configure resource group and data service support for Oracle RAC (Appendix A.3.3)
- Identify and create quorum devices (Appendix A.3.4)

More information on this step can be found Appendix A.3. The *Sun Cluster Software Installation Guide for Solaris OS* can be used as a supplemental guide and full reference.

Step 3.2.2 — Apply Solaris Cluster Core Patches

If applicable, download and apply the latest Solaris Cluster Core patch from http://sunsolve.sun.com. Follow the instructions provided for applying the patches included in the patch shipment

Task 3.3 — Establish Oracle Clusterware

The following steps explain how to install Oracle Clusterware.

Step 3.3.1 — Install Oracle Clusterware for Oracle Database 10g Release 2

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for the platform to install Oracle Clusterware on all database servers at the disaster site. The guide can be found at http://www.oracle.com/pls/db102/homepage. The software should be installed in an Oracle home that is separate from Oracle Database and Oracle ASM homes.

Remember to specify `clprivnet0` as the only private interconnect for Oracle Clusterware. Set the physical interconnect interfaces `ibd3` and `ibd5` to `Do Not Use`.

Step 3.3.2 — Apply Patchset 10.2.0.2 or Later

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

Task 3.4 — Establish Oracle ASM

This task establishes an Oracle ASM installation and creates the disk groups. During the testing effort, one disk group named `DATA` was created for data, and one named `FLASH` was created for flash recovery.

Step 3.4.1 — Install Oracle Database 10g Release 2 for Oracle ASM

Follow the instructions in the *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for the platform to create a clustered Oracle ASM installation on the database servers at the disaster site. The guide can be found at http://www.oracle.com/pls/db102/homepage. The software should be installed in an Oracle home that is separate from the Oracle Database and Oracle Clusterware homes. Choose the `software only` installation so that patches can be applied before creating the instance.

Step 3.4.2 — Apply Patchset 10.2.0.2 or Later

The latest database patch set for a platform can be found on Oracle MetaLink and should be applied.

Step 3.4.3 — Create the Oracle ASM Instance and Disk Groups

Use the DBCA utility to create an Oracle ASM instance and configure the disk groups. Be sure to create data and flash recovery disk groups.

Step 3.4.4 — Remove the Listener Created by DBCA

While configuring Oracle ASM, `DBCA` creates listeners that conflict with those that will be created by the Oracle E-Business Suite, and registers them to CRS. Use `netca` on either disaster recovery node to deregister the Oracle ASM listeners from CRS.

Task 3.5 — Prepare the existing database for Oracle Data Guard

The following steps explain how to prepare the existing database for Oracle Data Guard.

Step 3.5.1 — Create Password Files

Refer to the earlier Create Password Files section and ensure a password file is available for each of the disaster recovery database instances.

Step 3.5.2 — Configure SQL*Net for Communication Between Sites

Defining SQL*Net services, combined with keeping the standby redo logs and archive destination on shared storage in Oracle ASM, enables recovery operations to continue in the event a node fails at the standby site. In particular, set SQL*Net services that specify load balance to `no`, and those that specify failover to `yes` so that redos can be shipped between the production and disaster recovery sites. Create the configuration in the *CONTEXT_NAME*`_ifile.ora` include file located in the `$TNS_ADMIN` directory on each database server node in order to keep `AutoConfig` from overwriting the configuration.

The following SQL*Net configuration was used during testing efforts.

```
VIS_HA1=
    (DESCRIPTION=
      (LOAD_BALANCE=NO)
      (FAILOVER=YES)
      (ADDRESS_LIST=
        (ADDRESS=
          (PROTOCOL=tcp)
          (HOST=ha1db01.ha.us.oracle.com)
          (PORT=1521))
        (ADDRESS=
          (PROTOCOL=tcp)
          (HOST=ha1db02.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=ha2db01.ha.us.oracle.com)
      (PORT=1521))
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=ha2db02.ha.us.oracle.com)
      (PORT=1521))
   )
   (CONNECT_DATA=(SERVICE_NAME=VIS))
)
```

Step 3.5.3 — Configure Database Instance Parameters

Configure the existing database instances to perform as an Oracle Data Guard primary and standby for RMAN, and for Parallel Concurrent Processing, if desired. The 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`.

TIP

Oracle Data Guard parameters for primary operation must be applied immediately to enable standby operation to be initiated without a database restart.

As this is intended to be a permanent Oracle Data Guard configuration, service names are used for database connections, rather than EZConnect. However, the file name conversion entries are retained for the original single-instance database here, as this parameter must be identical across Oracle RAC instances and cannot be changed dynamically. Changing this parameter across the production cluster to remove the original reference would require a complete database shutdown, and could be scheduled for a future maintenance window at the primary site.

The following Oracle Data Guard parameters were used during testing efforts when the database was the 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 OP-
    TIONAL 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
```

The following Oracle Data Guard parameters were used during testing efforts when the database was the 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
```

The following file management parameters, used by RMAN, were used during testing efforts.

```
db_create_file_dest='+DATA'
db_recovery_file_dest='+FLASH'
db_recovery_file_dest_size=1228800M
control_files = +DATA/VIS_HA1/CONTROLFILE/control01.ctl
```

Optional PCP parameters used during testing included the following.

```
_lm_global_posts=TRUE
_immediate_commit_propagation=TRUE
```

Step 3.5.4 — Grant Access to Database Nodes

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

Step 3.5.5 — Add the Standby Redo Logs

If the standby redo logs are not already created, see the *Create Standby Redo Logs* section earlier in this document for more information.

Task 3.6 — Clone the Database Software and Prepare the New Instances

The following steps outline how to clone the database software and prepare the new instances.

Step 3.6.1 — Prepare the Database Oracle Home for Cloning

Run the `adpreclone.pl` script as the `ORACLE` user on one of the production database servers. Supply the `APPS` password when requested.

```
ha1db01:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
ha1db01:$ perl adpreclone.pl dbTier
```

Step 3.6.2 — 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 restore the Oracle home to the first Oracle RAC database node. It is best to use the same directory structures on the disaster recovery site.

Step 3.6.3 — Configure the New Database Oracle Homes

Execute the following `adcfgclone.pl` script for each new database Oracle home on the disaster recovery site. The following example uses the host `ha2db01`.

```
ha2db01:$ cd $ORACLE_HOME/appsutil/clone/bin
ha2db01:$ perl adcfgclone.pl dbTechStack
```

Respond to the prompts presented as follows. Repeat this step for all disaster recovery database nodes.

Prompt	Response
Do you want to use a virtual hostname for the target node (y/n)	N
Target instance is a Real Application Cluster (Oracle RAC) instance (y/n)	Y
Current node is the first node in an N Node Oracle RAC Cluster (y/n) *Answer "y" for the 2nd node also since DB is not up.*	y
Number of instances in the Oracle RAC Cluster	2
Target System database name *The db_name, not the global_db_name*	VIS
Do you want to preserve the port values from the source system on the target system (y/n)	y
Provide information for the Node 1 (current node). *Information for this node.*	
Host name	ha2db01
Virtual host name	ha2db01-vip
Instance number	1
Private interconnect name	ha2db01-ci
Provide information for the Node 2: *Information for the other node.*	
Host name	ha2db02
Virtual host name	ha2db02-vip
Instance number [1]:2	2
Private interconnect name	ha2db02-ci
Target system RDBMS ORACLE_HOME directory	/u01/app/oracle/visdbRAC/10.2.0
Target system utl_file accessible directories list	/usr/tmp
Number of DATA_TOP's on the target system	1
Target system DATA_TOP 1	+DATA/vis_ha2
Do you want to preserve the Display set to ha1db02:0.1 (y/n)	y

The `adcfgclone` script creates new context and environment files. The environment file should be sourced before executing commands for each instance. The following example used host `ha2db01`.

```
ha2db01:$ cd $ORACLE_HOME

ha2db01:$ . ./<CONTEXT_NAME>.env
```

Step 3.6.4 — Additional Configuration for the Second Node

Since the second node is not running at this point in the process, it cannot retrieve topology information from the database. The following manual changes are needed on the second node. Note that node `ha2db02` is used in the example below.

- Edit the $`ORACLE_HOME/appsutil/`CONTEXT_NAME`.xml` context and modify the following variables.

Variable	Description
Instance_number	Specifies the instance number
Instance_thread	Specifies the redo log thread
Undotablespace	Specifies the undotablespace name

- Remove the database initialization files to enable `AutoConfig` to recreate them.

```
ha2db02:$ cd $ORACLE_HOME/dbs

ha2db02:$ rm init<SID>.ora <SID>_APPS_BASE.ora
```

- Run `AutoConfig`.

```
ha2db02:$ cd $ORACLE_HOME/appsutil/bin

ha2db02:$ ./adconfig.sh  contextfile =
$ORACLE_HOME/appsutil/<CONTEXT_NAME> run=INSTE8
```

Step 3.6.5 — Configure SQL*Net for Communication Between Sites

When adjusting parameters on the production site for disaster recovery setup, an include file was created that holds TNS service definitions for failover, rather than load balancing, across Oracle RAC instances for each node. Copy the file to the `$TNS_ADMIN` directory for each instance at the disaster recovery site and name it *CONTEXT_NAME*`_ifile.ora`.

Step 3.6.6 — Configure and Restart the Listeners

While the listener service definition generated by `AutoConfig` points to the standby database, it does not include all required interface names. Create a listener include file named `listener_ifile.ora` in the `$TNS_ADMIN` directory on each standby database server. Example contents from the testing effort are shown below.

Example listener configuration for the first node.

```
VIS1=
ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP)(Host=
   ha2db01.ha.us.oracle.com)(Port=1521))
 (ADDRESS=(PROTOCOL=TCP)(Host=
   ha2db01-vip.ha.us.oracle.com)(Port=1521))
)
```

Example listener configuration for the second node.

```
VIS2=
ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP)(Host=
   ha2db02.ha.us.oracle.com)(Port=1521))
 (ADDRESS=(PROTOCOL=TCP)(Host=
   ha2db02-vip.ha.us.oracle.com)(Port=1521))
)
```

Start and stop the listeners. For example:

```
ha2db01:$ lsnrctl stop VIS1

ha2db01:$ lsnrctl start VIS1

ha2db02:$ lsnrctl stop VIS2

ha2db02:$ lsnrctl start VIS2
```

Step 3.6.7 — Configure Database Instance Parameters

Configure the new database instance to perform as an Oracle Data Guard standby and primary so that is ready for the switchover and switchback, respectively. The instance must also be configured for correct RMAN operation when restoring to the new Oracle ASM disk groups. If implementing Parallel Concurrent Processing is desired, consider adding the database parameters for using queues for Transaction Manager communication at this time.

Example Oracle 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
```

Example Oracle 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
```

Example 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
```

Example PCP parameters:

```
_lm_global_posts=TRUE
_immediate_commit_propagation=TRUE
```

Task 3.7 — Establish the Standby Database

The follows steps outline the process for establishing the standby database.

Step 3.7.1 — Backup the Database Using RMAN

Use the appropriate system tools to backup the new database server. Best practices include storage system remote replication of the disk backup (available on the Sun StorageTek 5320 NAS Appliance and Sun StorageTek 6540 array) when network bandwidth is available, or physically transporting tape cartridges or relocating the disk storage device containing the backup when network bandwidth is insufficient. The following example was executed on one of the production database nodes during testing efforts.

```
ha1db01:$ rman target /

ha1db01:RMAN> backup device type disk format '/NAS/oracle/rman_backups/%U' database plus
     archivelog;

ha1db01:RMAN> backup device type disk format '/NAS/oracle/rman_backups/%U' current
controlfile for standby;
```

Step 3.7.2 — Ship the Backup to the New Database Server

Use the appropriate system tools to ship the backup to the new database server. Best practices include storage system remote replication of the disk backup (available on the Sun StorageTek 5320 NAS Appliance and Sun StorageTek 6540 array) when network bandwidth is available, or physically transporting tape cartridges or relocating the disk storage device containing the backup when network bandwidth is insufficient.

Step 3.7.3 — Start One Oracle RAC Instance

Start one of the Oracle RAC instances at the disaster recovery site with the NOMOUNT option.

```
ha2db01:SQL> startup nomount
```

Step 3.7.4 — Restore the Database Backup to the Standby Using RMAN

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

```
ha2db01:$ rman target sys/manager@ha1db01:1521/VIS auxiliary /

ha2db01:RMAN> duplicate target database for standby;
```

Step 3.7.5 — Start Managed Recovery

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

```
ha2db01:SQL> select DATABASE_ROLE from v$database;

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

On the primary database, enable the previously deferred remote destination by executing the following command from SQL*Plus connected as the sysdba user.

```
ha1db:SQL> alter system set log_archive_dest_state_2=enable SID=’*’;
```

On all nodes of the primary database, update the database configuration include file with following parameter.

```
log_archive_dest_state_2=enable
```

Place the standby database in managed recovery by executing the following command from SQL*Plus connected as the sysdba user.

```
ha2db01:SQL> recover managed standby database using current logfile disconnect;
```

Step 3.7.6 — Verify Correct Standby Operation

Validate that the standby database is correctly applying redo operations from the primary database. On the primary database, archive the current log using the following statement.

```
ha1db:SQL> alter system archive log current;
```

On the standby database, query the `gv$archived_log` view to verify the logs are received and applied.

```
ha2db01: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 3.8 — Clone the Application Software and Configure on the Disaster Recovery Site

The following steps explain how to clone the application software and configure it on the disaster recovery site.

Step 3.8.1 — Prepare the Application Tier for Cloning

For each type of application tier installed, such as Concurrent Processing, Administration, Web, or Forms, log on as the `APPLMGR` user and prepare the application tier for cloning. If a single common APPL_TOP is shared, or if full installs are used, this step need only be done on one application tier.

```
ha1mt01:$ cd <COMMON_TOP>/admin/scripts/<CONTEXT_NAME>

ha1mt01:$ perl adpreclone.pl appsTier
```

Step 3.8.2 — Copy the Application Tier File System

Using appropriate tool, make a copy of the application software and tech stack to the disaster recovery application servers. The software clone will be to run the application after the switchover.

Step 3.8.3 — Configure New Application Tier File Systems

Configure the new application tier file systems with the `adclonectx.pl` `APPL_TOP` script using the original context file, or a copy of it, as the source context file. The script must be run on each application tier node. The example below uses node `ha2mt01`.

```
ha2mt01:$ cd /u01/appltopRAC/viscomn/clone/bin

ha2mt01:$ perl adclonectx.pl contextfile=/u01/appltopRAC/visappl/admin/VIS_ha1mt01.xml
```

Provide the values required for the creation of the new APPL_TOP Context file, as follows. Repeat the procedure for all application tier nodes.

Prompt	Response
Do you want to use a virtual hostname for the target node (y/n)	n
Target hostname	ha2mt01
Do you want the inputs to be validated (y/n)	n
Target system database SID	VIS
Username for the applications file system owner	applmgr
Group for the applications file system owner	applmgr

Prompt	Response
Target system database server node	ha2db01
Target system database domain name	ha.us.oracle.com
Does the target system have more than one application tier server node (y/n)	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
Display	ha2mt01:0.0
Location of the JDK on the target system	/opt/java1.4
Perl to be used on the target system:	/usr/bin/perl
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_ha2mt01.xml
Do you want to use a virtual hostname for the target node (y/n)	n
Target hostname	ha2mt01
Do you want the inputs to be validated (y/n)	n
Target system database SID	VIS
Username for the applications file system owner	applmgr

Step 3.8.4 — Complete the New Application Tier File System Configuration

Use the `adcfgclone.pl` script to complete the application tier file system configuration with the context file generated in the previous step. The script must be run on each application tier node. The following example assumes node `ha2mt01`. Since the database is not available at this time, it is safe to ignore any errors related to a failed `AutoConfig` run. Repeat this step for all application tier nodes.

```
ha2mt01:$ export APPL_TOP=<APPL_TOP>/visappl
(Workaround for known issue, Bug 5520384)

ha2mt01:$ cd /u01/appltopRAC/viscomn/clone/bin

ha2mt01:$ perl adcfgclone.pl appsTier <APPL_TOP>/visappl/admin/VIS_ha1mt01.xml
```

Step 3.8.5 — Configure the Application Tiers for Oracle RAC

Several context variables may need to be set for load balancing services to be employed.

- The `adcfgclone` script does not set variables to use the load balanced services. To adjust these parameters, edit the `$ORACLE_HOME/appsutil/`*`CONTEXT_NAME`*`.xml` context file 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 these 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 multiple Concurrent Processing Nodes are in use, and the use of PCP is desired, refer to the *Configure Parallel Concurrent Processing* section of MetaLink Note 362135.1.

Task 3.9 — Update the Oracle Clusterware Configuration

If srvctl is to be used to control resources, add the new resources to CRS. See the *Update the Clusterware Configuration* earlier in this document for details.

Phase 4 — Ongoing Switchover and Failover Testing

The disaster recovery site can be made active and provide application services via a switchover when the production platform or site is performing planned maintenance. In the event the production site is unavailable due to a severe unplanned outage, the disaster site can be made active via a failover of services. The switchover and failover procedures should be tested regularly to validate the MAA configuration. This section explains the switchover, switch back, and failover procedures.

Task 4.1 — The Switchover Procedure

The steps to switch over to the remote standby database are detailed below. Because the Oracle E-Business Suite application is down during this stage, the following steps should be carefully practiced and scripted to minimize downtime.

Step 4.1.1 — Prepare the Remote Standby Database

If the database has been opened read only since the last time it was shut down, restart the database and ensure managed recovery is running and up to date. When the standby database is shut down, the Snapshot Copy and Local Mirroring facility of the Sun StorageTek 6540 array may be used to duplicate the standby database to facilitate rapid test cycles and increased data protection. When combined with Oracle Flashback Database, Sun StorageTek Snapshot Copy and Local Mirroring provide a flexible and comprehensive data management solution. Verify that flashback is enabled with the following query.

```
ha2db01:SQL> select flashback_on from v$database;

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

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

Shut down the Oracle E-Business Suite software and ensure it is completely shutdown. Stop all Oracle RAC instances, with the exception of the production site.

Step 4.1.3 — Switchover to the Remote Standby Database

Switching over to the remote standby involves several steps. On the primary database, commit to switchover to the standby.

```
ha1db01:SQL> alter database commit to switchover to standby with session shutdown;
```

Shutdown the primary database.

```
ha1db01:SQL> shutdown immediate
```

Stop the primary database listener on each node.

```
ha1db01:$ lsnrctl stop LISTENER_<hostname>
```

Verify the standby database is ready to be converted to be the new primary database.

```
ha2db01:SQL> select switchover_status from v$database;

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

Execute the following command on the standby database to make it the new primary database.

```
ha2db01:$ alter database commit to switchover to primary;
```

Step 4.1.4 — Enable Oracle Flashback on the New Primary (Optional)

Enable flashback on the database if it is not enabled.

```
ha2db01:SQL> alter database flashback on;
```

Step 4.1.5 — Open the Disaster Recovery Database

Use the following command to open the disaster recovery database.

```
ha2db01:SQL> alter database open;
```

Step 4.1.6 — Start Other Disaster Recovery Database Instances

Start other instances of the new primary database using normal procedures.

```
ha2db02:SQL> startup;
```

Step 4.1.7 — Remove the Old Application Topology

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

```
ha2db01:SQL> exec FND_NET_SERVICES.remove_system('VIS');
ha2db01:SQL> commit;
```

Step 4.1.8 — Configure the Disaster Recovery Database Tier

Run `AutoConfig` twice on each disaster recovery database node to configure the Oracle home for use by the Oracle E-Business Suite. The command needs to be run once on each database node to register the node. Once all nodes are registered, the command must be run again on each node to generate the correct SQL*Net configuration files. The following example assumes node `ha2db01`.

```
ha2db01:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>
ha2db01:$ ./adautocfg.sh
```

Step 4.1.9 — Restart the Listeners

Start the correct listener on each new primary database node. The following example uses node `ha2db01`.

```
ha2db01:$ lsnrctl stop VIS1
ha2db01:$ lsnrctl start LISTENER_ha2db01
```

Step 4.1.10 — Configure the Application Tiers

Run `AutoConfig` on all new primary application tier nodes. This step can be run in parallel on all nodes. The following example assumes node `ha2mt01`.

```
ha2mt01:$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
ha2mt01:$ ./adautocfg.sh
```

Step 4.1.11 — Start Oracle E-Business Suite

Start the Oracle E-Business Suite applications on the disaster recovery site. Online users can gain access at this time.

Step 4.1.12 — Start the Original Primary as the Standby

Start the database listeners on at least one database node at the original primary site. Next, start the database in mount mode and begin managed recovery.

```
ha1db01:$ lsnrctl start LISTENER_ha1db01
ha1db01:SQL> startup mount;
ha1db01:SQL> recover managed standby database using current logfile disconnect;
```

On the disaster recovery database, enable remote archiving by executing the following command from SQL*Plus connected as the `sysdba` user.

```
ha2db01: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 11i with Rapid Clone*.

Step 4.1.14 — Direct Users to the New System

The old standby system is now available to users as the production system. Direct users to the new URL.

Task 4.2 — The Switch Back Procedure

It may be necessary to switch back to the primary site after the switchover. Switching back to the original primary is the same as the switchover process. See the previous task for details.

Task 4.3 — The Failover Procedure

This section assumes the primary site is down and a disaster site is available. The following steps describe how to switch the production system to the remote disaster site and enable the original primary as an Oracle Data Guard standby using the Oracle Flashback feature. The Oracle E-Business Suite application is unavailable during this stage. As a result, the steps should be carefully practiced and scripted to ensure prompt execution in an emergency.

Step 4.3.1 — Failover to the Disaster Recovery Database

Execute the following command on the standby database to change it to be the new primary database.

```
ha2db01:SQL> recover managed standby database cancel;
ha2db01:SQL> recover managed standby database finish force;
ha2db01:SQL> alter database commit to switchover to primary;
```

Step 4.3.2 — Enable Oracle Flashback

If flashback was never been enabled for the disaster recovery database, enable it now. If flashback is already enabled the command provides a warning message.

```
ha2db01:SQL> alter database flashback on;
```

Use the following command to check flashback status.

```
ha2db01:SQL> select flashback_on from v$database;
```

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

Step 4.3.3 — Open the Disaster Recovery Database

Open the disaster recovery database. If the database was opened read-only since the last shutdown, it must be shutdown and restarted.

```
ha2db01:SQL> alter database open;
```

Step 4.3.4 — Start Other Oracle RAC Instances on the Disaster Recovery Site

The other Oracle RAC instances can be started normally at this time. Continue with the next step while the instances are starting up.

```
ha2db02:SQL> startup
```

Step 4.3.5 — Remove the Old Application Topology

Connect to the disaster recovery database using SQL*Plus as the `apps` user and execute the following commands to remove the old application topology.

```
ha2db01:SQL> exec FND_NET_SERVICES.remove_system('VIS');

ha2db01:SQL> commit;
```

Step 4.3.6 — Configure the Disaster Recovery Database Tier

Run AutoConfig twice on each disaster recovery database node to configure the Oracle home for use by the Oracle E-Business Suite. The command needs to be run on each database node to register the node. After all nodes are registered, the command must be run again on each node to generate the correct SQL*Net configuration files. The following example assumes node `ha2db01`.

```
ha2db01:$ cd $ORACLE_HOME/appsutil/scripts/<CONTEXT_NAME>

ha2db01:$ ./adautocfg.sh
```

Step 4.3.7 — Restart the Listeners

Start the correct listener on each new primary database node. The following example uses node `ha2db01`.

```
ha2db01:$ lsnrctl stop VIS1

ha2db01:$ lsnrctl start LISTENER_ha2db01
```

Step 4.3.8 — Configure the Application Tiers

Run AutoConfig on all new primary application tier nodes. This step can be run in parallel on all nodes. The following example uses node `ha2mt01`.

```
ha2mt01:$ cd $COMMON_TOP/admin/scripts/<CONTEXT_NAME>
ha2mt01:$ ./adautocfg.sh
```

Step 4.3.9 — Start Oracle E-Business Suite

Start the Oracle E-Business Suite applications on the disaster recovery site. Online users can access the applications at this time.

Step 4.3.10 — Establish the Original Primary as the Standby Using Oracle Flashback (Optional)

Once access to the failed production site is restored, it may be possible to reinstate the original primary database as a physical standby of the new primary database if flashback database is enabled. Make sure Oracle Clusterware and Oracle ASM are restarted on the original primary system. On the disaster recovery site, get the SCN when the database becomes the primary database.

```
ha2db01:SQL> select to_char(standby_became_primary_scn) from v$database;
```

On the original primary site, flashback and start managed recovery.

```
ha1db01:SQL> shutdown immediate;
ha1db01:SQL> startup mount;
ha1db01:SQL> flashback database to scn <standby_became_primary_scn>;
ha1db01:SQL> alter database convert to physical standby;
ha1db01:SQL> shutdown immediate;
ha1db01:SQL> startup mount;
ha1db01: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 11i with Rapid Clone.*

Step 4.3.12 — Direct Users to the New System

The old standby system should be available to users as the new production system. Direct users to the new URL.

Task 4.4 — Disaster Recovery Testing Procedure Using Flashback Database

This section describes how the disaster recovery configuration can be tested while the primary site is in live operation. It also explains how Flashback Database can be used to quickly restore the disaster recovery site to standby operation once testing is complete.

Step 4.4.1 — Activate and Open the Disaster Recovery Standby Database

Activating and opening the disaster recovery standby database requires several step, as outlined below.

- Cancel managed recovery on the disaster recovery site.

```
ha2db01:SQL> recover managed standby database cancel;
```

- Enable flashback on the disaster recovery site, if it is not enabled.

```
ha2db01:SQL> alter database flashback on;
```

- Create a guaranteed restore point on the disaster recovery site. The example below uses the name `testing_starts`.

```
ha2db01:SQL> create restore point testing_starts guarantee flashback database;
```

- Switch the current log on the primary site and defer the archive destination.

```
ha1db01:SQL> alter system archive log current;
ha1db01:SQL> alter system set log_archive_dest_state_2=defer SID=’*’;
```

- Activate and open the database on the disaster recovery site.

```
ha2db01:SQL> alter database activate standby database;
ha2db01:SQL> alter database set standby database to maximize performance;
ha2db01:SQL> alter database open;
```

Step 4.4.2 — Perform Testing

The database is now open and can be used for testing. Any changes made to the database will be rolled back later using the flashback database. Additional database instances can be started, and applications tested. Note that the disaster recovery site will likely lag behind on redo application during the testing period. Be sure to not get too far behind.

Step 4.4.3 — Flashback the Database and Resume Standby Operation

Shutdown all but one database instance on the disaster recovery site. In the following example, only the database instance on node `ha2db01` remains.

- Flashback and started managed recovery on the disaster recovery site.

```
ha2db01:SQL> startup mount force;
ha2db01:SQL> flashback database to restore point testing_starts;
ha2db01:SQL> drop restore point testing_starts;
ha2db01:SQL> alter database convert to physical standby;
ha2db01:SQL> startup mount force;
ha2db01:SQL> recover managed standby database using current logfile disconnect;
```

- Enable the archive destination on the primary site.

```
ha1db01:SQL> alter system set log_archive_dest_state_2=enable SID=’*’;
```

Appendix A — Solaris Cluster Software Configuration

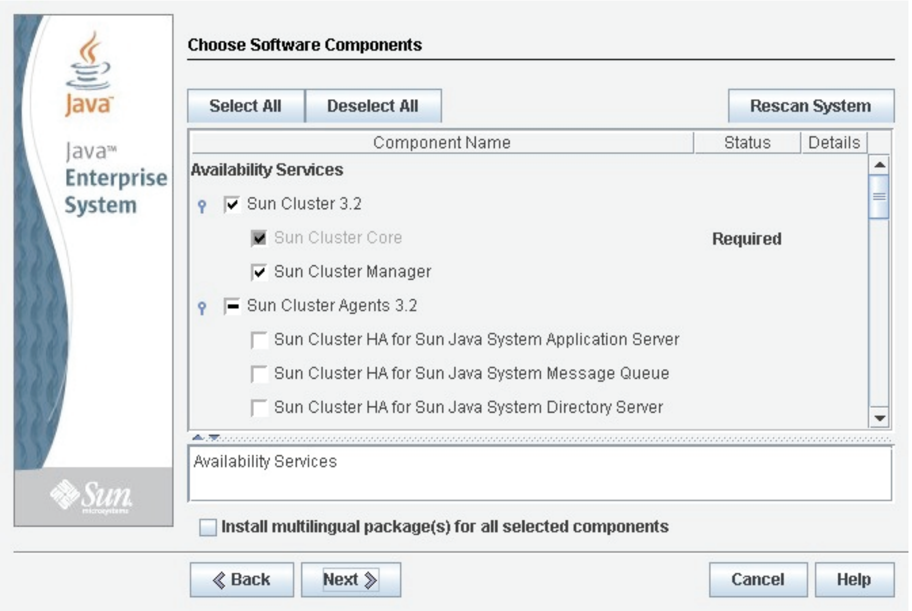
The following steps explain how to configure the Solaris Cluster software for MAA.

Task A.1 — Creating a Cluster with the First Node Added to the Two-Node Cluster

The following steps show how to create a cluster with a single node.

Step A.1.1 — Install the Solaris Cluster Software

Run the installer and select the following component for installation: Sun Cluster 3.X, Sun Cluster Agents: Sun Cluster HA for Oracle and Sun Cluster Support for Oracle RAC, and all shared components. Proceed through the installation screens and select the `Configure Later` option. Review the selected options and click `Install`.



Choose Software Components

Component Name	Status	Details
Availability Services		
☑ Sun Cluster 3.2		
☑ Sun Cluster Core	Required	
☑ Sun Cluster Manager		
▬ Sun Cluster Agents 3.2		
□ Sun Cluster HA for Sun Java System Application Server		
□ Sun Cluster HA for Sun Java System Message Queue		
□ Sun Cluster HA for Sun Java System Directory Server		

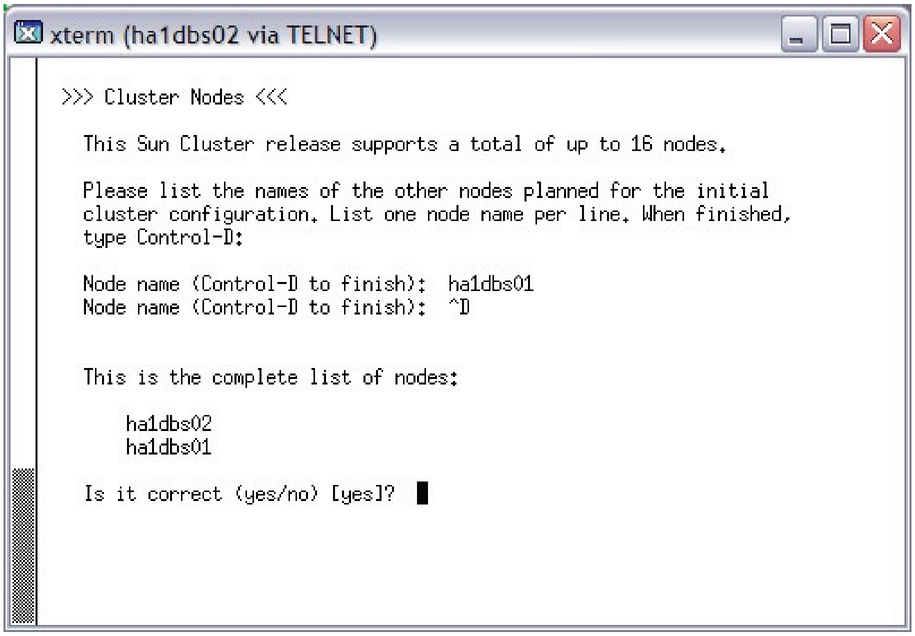
Availability Services

□ Install multilingual package(s) for all selected components

Step A.1.2 — Create and Configure the Cluster with the Initial Node

The following steps explain how to create and configure the cluster with an initial node.

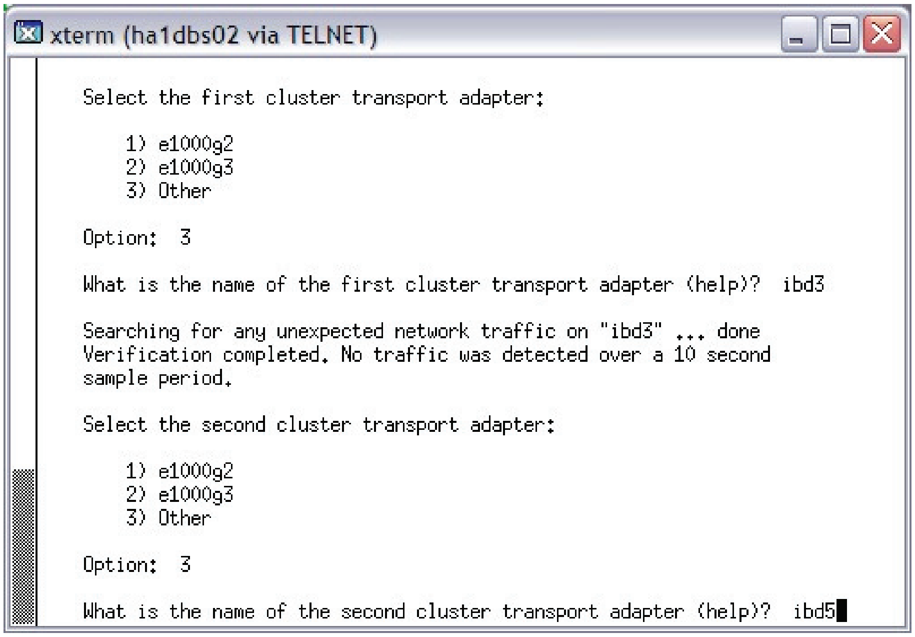
- Move to the `/usr/cluster/bin` directory and execute the `scinstall` script.
- Section Option 1 to create a new cluster.
- Section Option 2 to create just the first node of a new cluster on this machine.
- Choose `Typical` mode and enter the name for the cluster, such as `MAA-X64`.
- Answer `Yes` to `sccheck` and proceed.
- The next screen prompts for a list of the names of other nodes that will be members of the cluster. Since a two-node cluster is being built, enter the name of the other node, such as `ha1dbs01`, and enter `Ctrl-D` to end the list. Note that the name of the node being configured does not need to be entered.



xterm (ha1dbs02 via TELNET)

```
>>> Cluster Nodes <<<

  This Sun Cluster release supports a total of up to 16 nodes.

  Please list the names of the other nodes planned for the initial
  cluster configuration. List one node name per line. When finished,
  type Control-D:

  Node name (Control-D to finish):  ha1dbs01
  Node name (Control-D to finish):  ^D


  This is the complete list of nodes:

      ha1dbs02
      ha1dbs01

  Is it correct (yes/no) [yes]?
```

- Identify the cluster transport adapters, or cluster interconnects, by selecting `Other` and entering the name of the first cluster transport, `ibd3`.
- Select and input the second transport adapter, `ibd5`.



xterm (ha1dbs02 via TELNET)

```
Select the first cluster transport adapter:

    1) e1000g2
    2) e1000g3
    3) Other

Option:  3

What is the name of the first cluster transport adapter (help)?  ibd3

Searching for any unexpected network traffic on "ibd3" ... done
Verification completed. No traffic was detected over a 10 second
sample period.

Select the second cluster transport adapter:

    1) e1000g2
    2) e1000g3
    3) Other

Option:  3

What is the name of the second cluster transport adapter (help)?  ibd5
```

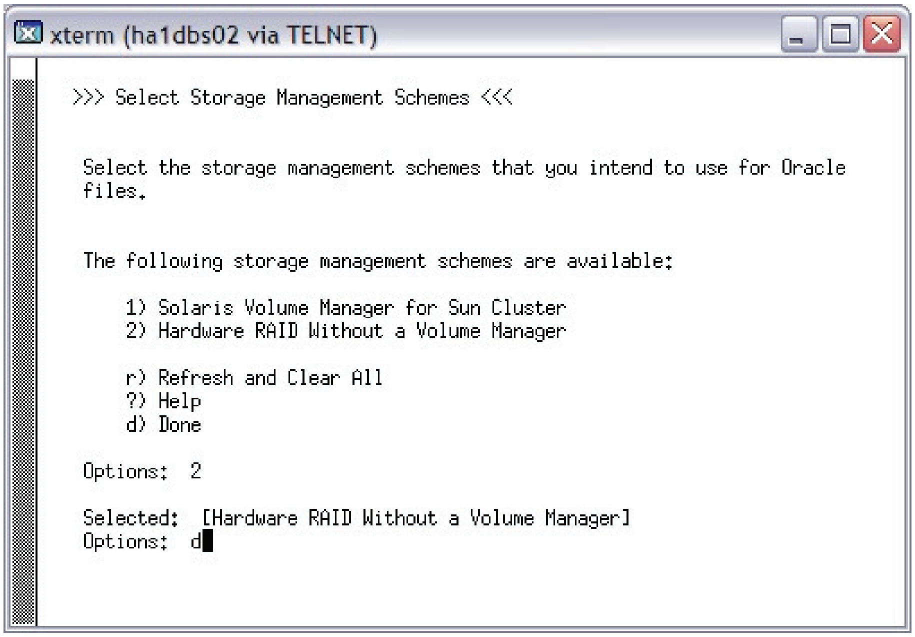
- Next, answer `yes` to disable automatic device selection. It will be configured manually in a later stage. In addition, agree to an automatic reboot of the node by `scinstall`, and answer `no` to interrupt cluster creation for `sccheck` errors.
- Review the configuration responses and proceed with the configuration.

- Cluster configuration and creation begins. Once complete, the node is rebooted and started in cluster mode.

Step A.1.3 — Configure Solaris Cluster Resource Group and Data Services Support for Oracle RAC

The following steps explain how to configure Solaris Cluster resource group and data services support for Oracle RAC.

- Run the `/usr/cluster/bin/clsetup` utility.
- Select Option 3, `Data Services`.
- Select Option 4, `Oracle Real Application Clusters`.
- Select Option 1, `Oracle RAC Create Configuration` and select the `RAC Framework Resource Group`.
- Verify prerequisites and press `Enter` to continue.
- Section Option 2, `Hardware RAID Without a Volume Manager` for the storage management scheme, as Oracle ASM will be used for Oracle datafiles.



```
 >>> Select Storage Management Schemes <<<


  Select the storage management schemes that you intend to use for Oracle
  files.


  The following storage management schemes are available:

      1) Solaris Volume Manager for Sun Cluster
      2) Hardware RAID Without a Volume Manager

      r) Refresh and Clear All
      ?) Help
      d) Done

  Options:  2

  Selected:  [Hardware RAID Without a Volume Manager]
  Options:  d
```

- Review the Sun Cluster objects and select `Done`.
- Select item c, `Create Configuration`.
- The Oracle RAC framework resource group and data services are created and enabled with online status. Run the `scstat –g` command to verify operation.

Task A.2 — Adding the Second Node to the Existing Cluster

The following steps explain how to add the second node to the cluster created in the previous steps.

Step A.2.1 — Install the Solaris Cluster Software

Install and configure the Solaris Cluster software on the second node by running `installer` and selecting the appropriate components to be installed. The steps outlined in task A.1.1 can be used as a guide.

Step A.2.2 — Configure the Cluster by Adding a New Node

Adding a new node and configuring the cluster requires the following steps.

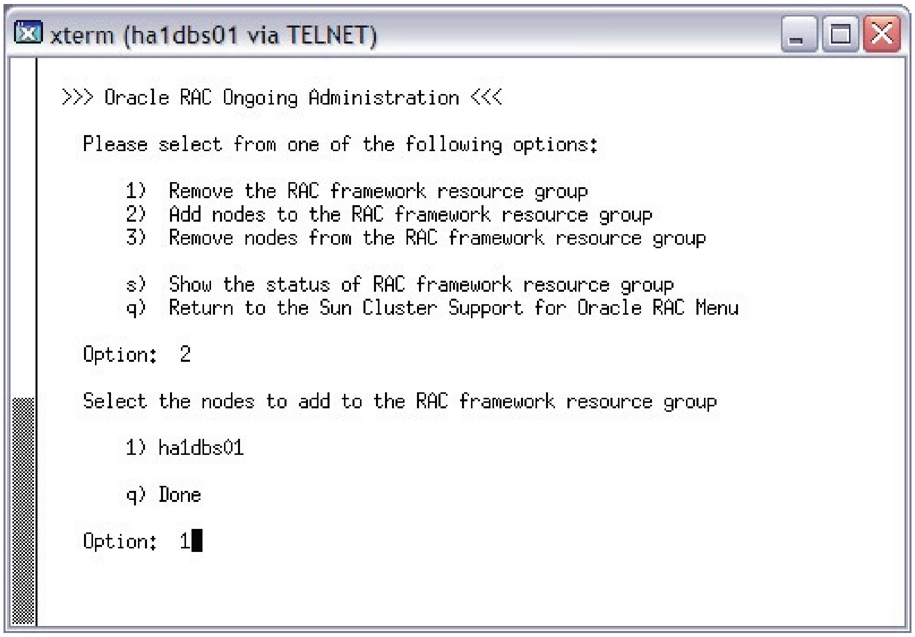
- Execute the `scinstall` script.
- Select Option 1, `Create a new cluster or add a cluster node`.
- Select Option 3, `Add this machine as a node in an existing cluster`.
- Choose `Typical` mode and enter the name of the sponsoring node. Since this task adds a node to an existing cluster, a sponsoring node must be specified. In a two-node cluster, the sponsoring node is usually the first node in the cluster. In the example below, the sponsoring node is node `ha1dbs02`.
- Enter the name of the cluster to join, such as `MAA-X64`.
- Answer `Yes` to `sccheck` and proceed.
- Answer `No` to `Autodiscovery of Cluster Transport`.
- Identify the cluster transport adapters, or cluster interconnects, by selecting `Other` and entering the name of the first cluster transport, `ibd3`.
- Select and input the second transport adapter, `ibd5`.
- Answer `yes` to disable automatic quorum device selection as it will be configured manually in a later stage. Agree to the automatic reboot of the node by `scinstall` and say `no` to `Interrupt cluster creation for sccheck errors`.
- Review the configuration responses and proceed with configuration.
- Cluster configuration and creation begins. Once completed, the node is rebooted and joined in cluster mode.

Step A.2.3 — Configure Solaris Cluster Resource Group and Data Service Support for Oracle RAC

The following steps outline the procedure for configuring Solaris Cluster resource group and data service support for Oracle RAC.

- Run the `clsetup` utility.
- Select Option 3, `Data Services`.
- Select Option 4, `Oracle Real Application Clusters`.
- Select Option 2, `Oracle RAC Ongoing Administration`.

- Select `Add Nodes to the ORACLE RAC Framework Resource Group` and select the name of the node to be added. The following example specifies node `ha1dbs01`.



xterm (ha1dbs01 via TELNET)

```
>>> Oracle RAC Ongoing Administration <<<

  Please select from one of the following options:

      1)  Remove the RAC framework resource group
      2)  Add nodes to the RAC framework resource group
      3)  Remove nodes from the RAC framework resource group

      s)  Show the status of RAC framework resource group
      q)  Return to the Sun Cluster Support for Oracle RAC Menu

  Option:  2

  Select the nodes to add to the RAC framework resource group

      1) ha1dbs01

      q) Done

  Option:  1
```

- Verify the information presented and press `Enter` to proceed with the update.

- The cluster should identify both nodes as members, `ha1dbs02` and `ha1dbs01`. Oracle RAC framework resource group and data service should be created and enabled with online status. Run the `scstat` command to verify service status.

Step A.2.4 — Identify and Configure the Quorum Devices

The following steps describe how to identify and configure the quorum devices.

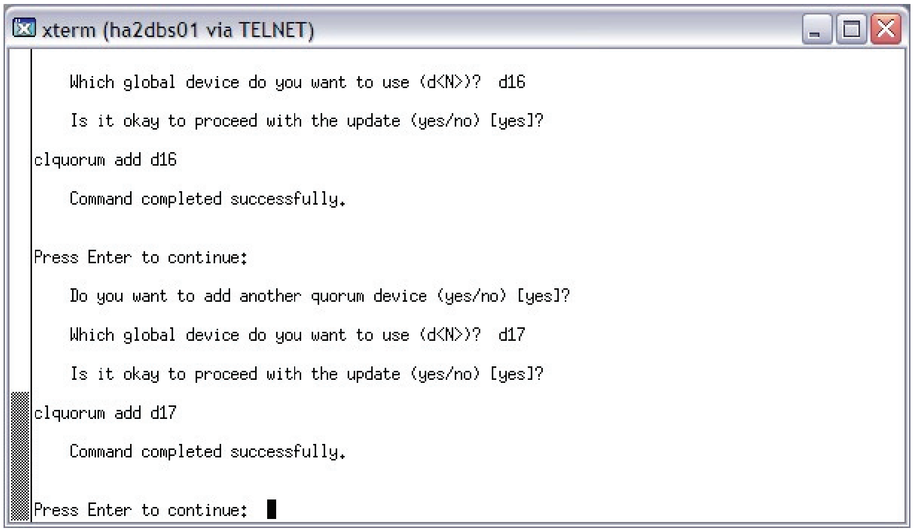
- Run the `clsetup` command.

- Select the `Quorum` option.

- Select `Directly attached shared disk`.

- Specify the global devices to be used as quorum. The device names are in the format `dxx` and are assigned by the Solaris Cluster software. Run the `/usr/cluster/bin/scdidadm –i` command to obtain a mapping between shared physical devices and global devices.

- Add additional quorum devices, if desired. The example below specifies two quorum devices, `d16` and `d17`.



```
    Which global device do you want to use (d<N>)?  d16

    Is it okay to proceed with the update (yes/no) [yes]?

clquorum add d16

    Command completed successfully.


Press Enter to continue:

    Do you want to add another quorum device (yes/no) [yes]?

    Which global device do you want to use (d<N>)?  d17

    Is it okay to proceed with the update (yes/no) [yes]?

clquorum add d17

    Command completed successfully.


Press Enter to continue:
```

Task A.3 — Creating the Cluster on the Disaster Recovery Site

The following steps explain how to create the cluster on the disaster recovery site.

Step A.3.1 — Install the Solaris Cluster Software

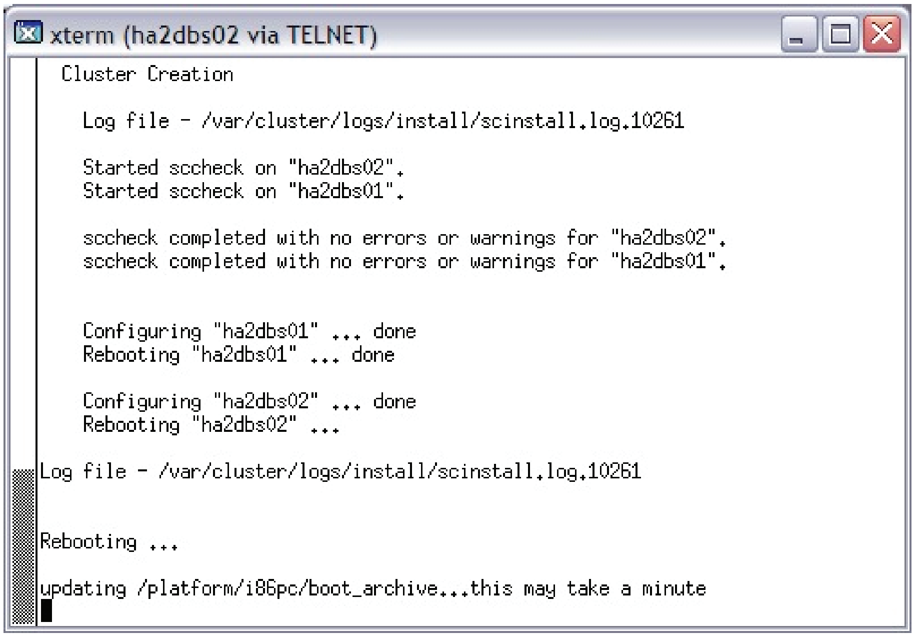
Install and configure the Solaris Cluster software on both nodes at the disaster recovery site by running the `installer` and selecting the appropriate components to be installed. See the steps outline in task A.1.1 for details.

Step A.3.2 — Create and Configure the Cluster with Two Nodes

The following steps create and configure the cluster with two nodes.

- Run `scinstall` on node 1 of the cluster.
- Select Option 1, `Create a new cluster or add a cluster node` followed by Option 1, `Create a new cluster`.
- Choose `Typical` mode and enter the name of the cluster to be created, such as `MAA-X64-S2`.
- Answer `Yes` to `sccheck` and proceed.
- Enter the name of the second node, such as `ha2dbs02`. Type `Ctrl-D` to finish the listing. Verify the names of the nodes to be included in the cluster and continue.
- Identify the cluster transport adapters, or cluster interconnects, by selecting `Other` and entering the name of the first cluster transport, `ibd3`.
- Select and input the second transport adapter, `ibd5`.
- Answer `No` to `Autodiscovery of Cluster Transport`.
- Answer `yes` to disable automatic quorum device selection as it will be configured manually in a later stage. Agree to the automatic reboot of the node by `scinstall` and say `no` to `Interrupt cluster creation for sccheck errors`.

- Review the configuration responses and proceed with the configuration.
- Cluster configuration and creation begins. Once complete, all nodes are rebooted to form a cluster, as illustrated in the screenshot below.

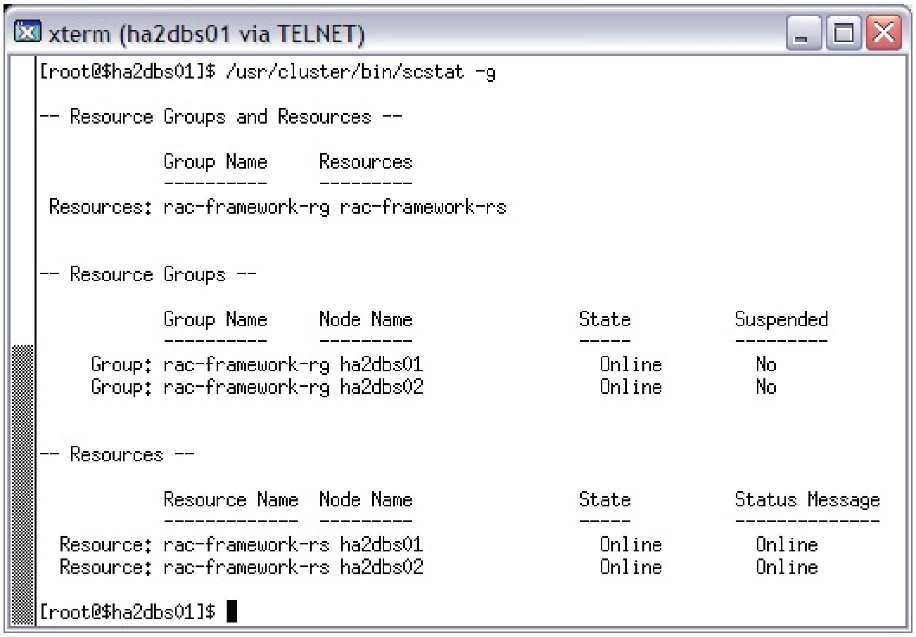


```
 Cluster Creation

   Log file - /var/cluster/logs/install/scinstall.log.10261

   Started sccheck on "ha2dbs02".
   Started sccheck on "ha2dbs01".

   sccheck completed with no errors or warnings for "ha2dbs02".
   sccheck completed with no errors or warnings for "ha2dbs01".


   Configuring "ha2dbs01" ... done
   Rebooting "ha2dbs01" ... done

   Configuring "ha2dbs02" ... done
   Rebooting "ha2dbs02" ...

Log file - /var/cluster/logs/install/scinstall.log.10261


Rebooting ...

updating /platform/i86pc/boot_archive...this may take a minute
```

Step A.3.3 — Configure Solaris Cluster Resource Group and Data Service Support for Oracle RAC

The following steps explain how to configure Solaris Cluster resource group and data service support for Oracle RAC. The steps below are identical to Step A.1.3 earlier in this section.

- Run the `/usr/cluster/bin/clsetup` utility.
- Select Option 3, `Data Services`.
- Select Option 4, `Oracle Real Application Clusters`.
- Select Option 1, `Oracle RAC Create Configuration` and select the `RAC Framework Resource Group`.
- Verify prerequisites and press `Enter` to continue.
- Section Option 2, `Hardware RAID Without a Volume Manager` for the storage management scheme, as Oracle ASM will be used for Oracle datafiles.
- Review the Sun Cluster objects and select `Done`.
- Select item c, `Create Configuration`.
- The Oracle RAC framework resource group and data services are created and enabled with online status. Run the `scstat –g` command to verify operation.



```
[root@$ha2dbs01]$ /usr/cluster/bin/scstat -g

-- Resource Groups and Resources --

            Group Name      Resources
            ----------      ---------
 Resources: rac-framework-rg rac-framework-rs


-- Resource Groups --

            Group Name      Node Name                State        Suspended
            ----------      ---------                -----        ---------
     Group: rac-framework-rg ha2dbs01                  Online       No
     Group: rac-framework-rg ha2dbs02                  Online       No


-- Resources --

            Resource Name   Node Name                State        Status Message
            -------------   ---------                -----        --------------
  Resource: rac-framework-rs ha2dbs01                  Online       Online
  Resource: rac-framework-rs ha2dbs02                  Online       Online

[root@$ha2dbs01]$ 
```

Step A.3.4 — Identify and Configure Quorum Devices

The following steps can be used to identify and configure quorum devices.

- Run the `/usr/cluster/bin/clsetup` utility and select the `Quorum` option.
- Select `Directly attached shared disk`.
- Specify the global devices to be used as the quorum. The device names are in the form `dxx` and are assigned by the Solaris Cluster software. Use the `/usr/cluster/bin/scdidadm -l` command to obtain a mapping between the shared physical devices and global devices.
- Add additional quorum devices, if desired. The testing effort configured two quorum devices `d16` and `d17`.

Appendix B — Storage Systems

This section describes the storage systems used in the tested MAA configuration.

Sun StorageTek 6540 Array

The goal of the Maximum Availability Architecture is to achieve high availability by being able to prevent, tolerate, and recover quickly from outages. The production database storage for the Oracle E-Business Suite 11.5.10.2 database is a Sun StorageTek 6540 array. The Sun StorageTek 6540 array prevents and tolerates outages by incorporating an N+1 configuration on all components. Equipped with four host interfaces per controller that run at 4 Gb/second, a 1 GB data cache per controller, and 2 GB of battery-backed cache, the Sun StorageTek 6540 array provides the connectivity needed for clustered environments. Each site uses a separate Sun StorageTek 6540 storage array.

The arrays used during the testing effort provided 3.5 TB of usable RAID-5 storage, with a storage partition option used to mask the LUNs to different nodes in the cluster. Multiple McData SAN switches provided the multiple paths between the hosts and storage arrays. Figure B-1 depicts the database tier and back-end storage. Each SAN switch is connected to each node in the cluster. Each Sun StorageTek 6540 array includes two controllers, each connected to each SAN switch.

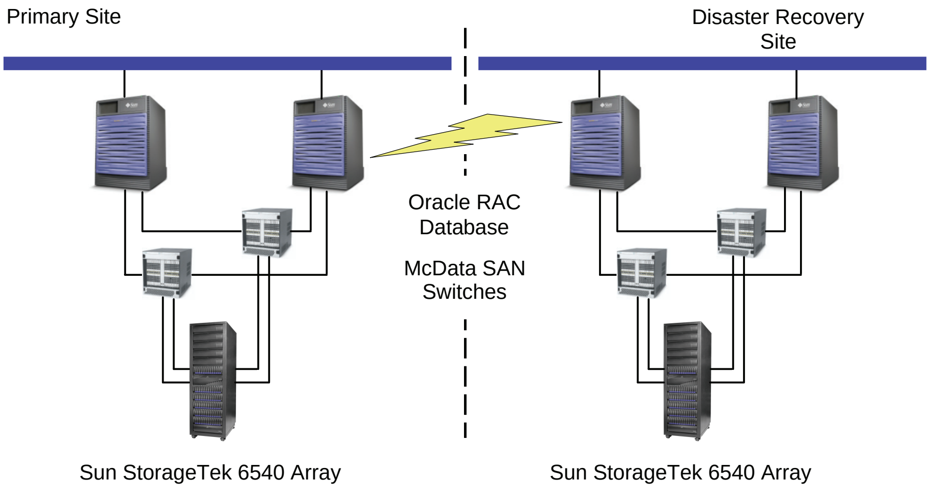


Figure B-1. The database tier and back-end storage configuration

The Sun StorageTek 6540 array prevents outages by tolerating component failures and environmental failures. Based on a completely redundant hardware architecture, no single component failure causes loss of data or access to data. The Sun StorageTek 6540 array architecture also facilitates protection against non-array failures, including redundant power supplies which may be connected to redundant power distribution systems, redundant controllers that may be connected to redundant storage area networks, and tight integration with the Solaris OS through MPXIO multipath software to prevent Fibre Channel network failures from causing loss of access to data. In the event of significant site failures, such as a complete loss of all power to the enterprise, batteries protect the cache contents of the array for up to seven days. Component replacement, firmware updates, and hardware upgrades may be done with the array online to further reduce downtime and improve data availability.

Caching Parameters

The cache block size is a global parameter for the storage subsystem. Set the cache block size closest to the typical I/O size — set the cache block size to a smaller size for transactional workloads, and to a larger size for sequential I/O. This can be easily changed at any time to optimize for a particular workload during a specific time period.

LUN-Specific Cache Parameters

Use of a performance monitor to view the cache hit percentages and read/write mix for each LUN of interest while the application is running can assist in the fine tuning of the following cache parameters.

Cache Flush — Write data in the controller cache is flushed to disk based on the demand for cache space. Two global parameters control flushing: Start Flushing and Stop Flushing. Writes to disk begin when the percentage of cache containing unwritten data exceeds the level specified in Start Flushing and stops when the percentage hits the Stop Flushing mark. Setting Start Flushing too low can cause excessive write activity and reduce overall performance. Setting Stop Flushing close to the Start Flushing value can result in a brief flushing operation to maintain a certain amount of free space. Start with the default values and experiment with different values for a given workload.

Media Scan — The Media Scan cache parameter enables the media scan option to kick off a periodic scan of the surface of all LUN disks. While the impact is minimal, extra reads do represent a finite workload, and should not be used when maximum performance is the primary objective.

Write Cache and Write Cache Mirroring — Enabling Write Cache on a LUN generally improves performance for applications with a significant write content. However, caching writes does introduce a small risk of data loss in the unlikely event of a controller failure. To eliminate the chance of data loss from a controller failure, the Write Cache Mirroring option can be enabled for a LUN, so that each write is contained in the cache of both controllers.

Segment Size

Segment size refers to the amount of data written to one disk in a volume group (VG) before writing to the next disk in the VG. For example, in a RAID-5 4+1 VG with a segment size of 128 KB, the first 128 KB of an I/O request is written to the first disk, the next 128 KB to the second disk, and so on. For a RAID-1 2+2 VG, 128 KB is written to each of the two disks. In both examples, if the I/O size is larger than (# disks * 128 KB), writes continue back at the first disk in the VG in the same pattern until the entire write request is satisfied.

For very large I/O requests, the optimal segment size for a RAID volume group is one that distributes a single host I/O request across all data drives. The formula is as follows:

```
LUN stripe width ÷ # of data drives = LUN segment size
```

For RAID 5, the number of data drives is the number of drives in the volume group minus one. For RAID 1, the number of data drives is the number of drives plus two. Examples for each type are shown below.

```
RAID 5, 4+1 w/ 64KB segment size =>(5-1)*64KB = 256 KB stripe width
RAID 1/0, 2+2 w/ 64KB segment size =>(2)*64KB = 128 KB stripe width
```

For small I/O requests, the segment size should be large enough to minimize the number of segments (disks in the LUN) that need to be accessed to satisfy the IO request (minimize segment boundary crossings). For IOPS environments, set the segment size so the stripe width is at least as large as the median I/O size.

When using a volume manager to collect multiple storage subsystem LUNs into an LVM volume group, the I/O stripe width is allocated across all the segments of all the data drives in all the LUNs. The formula used above becomes:

```
LUN segment size =
   LVM I/O stripe width ÷ (# of data drives/LUN * # of LUNs/VG)
```

Refer to the vendor documentation for a particular LVM to learn the terminology and understand how data in each I/O is allocated to each LUN in a logical volume group.

General Best Practices

- Choose faster disks. Note that 15K RPM drives provide 25 percent more performance than a 10K RPM drive.
- Choose low capacity drives to reduce seek times.
- Avoid multiple volumes on one volume group. This practice distributes data across a physical disk anad increases seek time penalties.
- Add more drives to the configuration for a linear increase in performance, up to the point of controller saturation. More disks generally mean more spindles to service I/O requests.
- Separate random and sequential workloads on different physical disks.
- Configure the entire capacity of a volume group with a single LUN.
- Stripe volume groups and the LUNs they contain across all drive modules to distribute the back-end traffic across all drive side loops and to maximize data availability.

The Sun StorageTek 6540 array further improves data availability through advanced data duplication and replication features. Snapshot Copy quickly duplicates data, and makes the data available for other business requirements including backup, test, and development processes. Local Mirroring duplicates data and augments Snapshot Copy by creating a completely independent copy of the data. Remote Replication over local or wide area networks (WANs) duplicates data on an alternate Sun StorageTek 6540 array to make the data available after a local disaster. When combined with Oracle availability features, such as Flashback Database, RMAN and Oracle Data Guard, Snapshot Copy, Local Mirroring, and remote mirroring provide a complete and comprehensive foundation to ensure data is available.

In addition to protecting data and ensuring data availability, the Sun StorageTek 6540 array can also ensure that data is only available to specific systems. Sun StorageTek Storage Partitioning controls access to data so that different computing systems supported by the same storage array cannot access or modify the other's data.

Sun StorageTek 5320 NAS Appliance

The Sun StorageTek 5320 NAS Appliance is a midrange storage solution ideally suited for medium to large enterprises with large, rapidly growing file system storage requirements and the need to reduce system management costs and application deployment time while consolidating storage, improving availability, and sharing information. The Sun StorageTek 5320 NAS Appliance accomplishes this through clustering, multipath I/O, hardware RAID, and a simple to use, intuitive management interface. Purpose-built for simple operation, the Sun StorageTek 5320 NAS Appliance combines high performance with low maintenance requirements to create a versatile, NAS solution for mixed NFS, CIFS, and iSCSI environments.

Advanced local and remote data replication features further improve data availability by reducing backup windows and speeding the deployment of copies of production data for alternate business processing. With dedicated storage and availability in single NAS Filer Head as well as a no single point of failure cluster solution, the Sun StorageTek 5320 NAS Appliance provides a comprehensive, flexible, powerful, and highly scalable solution for today's shared storage requirements. Many applications are supported, including:

- *E-mail servers* — The Sun StorageTek 5000 NAS Appliance Family provides cross-platform file sharing and support for e-mail applications, such as SendMail and Lotus Notes. Microsoft E-mail tools, such as Exchange, are supported via iSCSI.

- *Storage consolidation* — The cross-platform file sharing support built into Sun StorageTek 5000 NAS Appliance Family aids in reducing the need for organizations to dedicate large storage resources to a single server.

- *Imaging and graphics* — Huge graphic files demand maximum performance that can only be achieved with a filer that is optimized for storage services. The Sun StorageTek 5000 NAS Cluster system can be best utilized for this purpose.

An optional feature, the Sun StorageTek Compliance Archiving System couples any product of the StorageTek 5000 NAS Appliance Family with the Compliance Archiving Software to provide compliance-enabling features for authenticity, integrity, ready access, and security. The compliance archiving software is designed from the ground up in consultation with information management compliance and enterprise content management industry experts to help address the most stringent requirements for electronic storage media retention and protection.

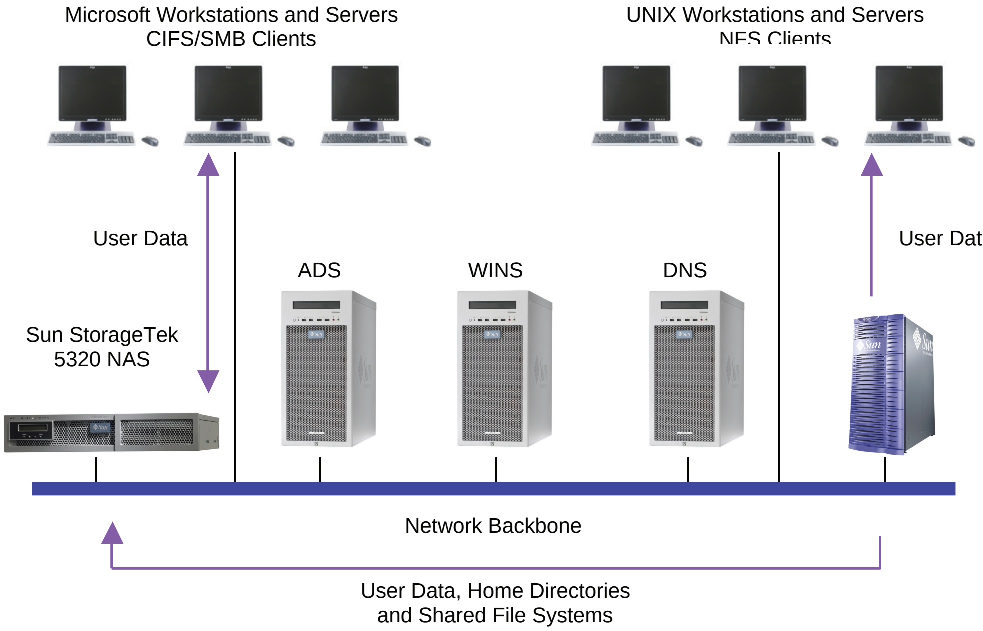


Figure B-2. The Sun StorageTek 5320 NAS Appliance enables storage to be shared among a wide a range of applications.

Appendix C — 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 10*g* 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 Guides for Oracle Database 10*g* Release 2:
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 10*g* 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 10*g* 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 Real Application Clusters and Automatic Storage Management
http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=362135.1

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

Oracle*MetaLink* Note 216205.1- Database Initialization Parameters for Oracle Applications 11i
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

Sun Cluster Software Installation Guide for Solaris OS
http://docs.sun.com/app/docs/doc/819-2970

Solaris Cluster 3.2 Documentation Collection
http://docs.sun.com/app/docs/prod/sun.cluster32#hic

Solaris 10 Operating System
http://docs.sun.com/app/docs/prod/solaris.10

Transitioning Oracle E-Business Suite to the Maximum Availability Architecture on the Solaris Operating System
September 2007
Authors: Jeffrey Wright, Brandon Hoang, Khader Mohiuddin, Richard Exley, Andrew Babb, Glen Co Ong

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