Configuring SAP R/3 4.6C and 4.7 Enterprise for Use with Oracle Real Application Clusters

An Oracle White Paper
March 2005
# Configuring SAP R/3 To Use With Oracle Real Application Clusters

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>System Requirements and Assumptions</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Part I  Configure the Hardware</strong></td>
<td>6</td>
</tr>
<tr>
<td>Prerequisites: Checks on the existing database provided by SAP</td>
<td>6</td>
</tr>
<tr>
<td>Overview: Supported Configurations for SAP R/3</td>
<td>7</td>
</tr>
<tr>
<td>HP Tru64 Cluster</td>
<td>11</td>
</tr>
<tr>
<td>IBM AIX 5L with HACMP</td>
<td>12</td>
</tr>
<tr>
<td>Sun Cluster 3.1</td>
<td>17</td>
</tr>
<tr>
<td>Fujitsu PrimeCluster</td>
<td>20</td>
</tr>
<tr>
<td>Veritas Database Edition/Advanced Cluster (DBE/AC) for HP-UX PA-RISC and Sun Sparc Solaris</td>
<td>25</td>
</tr>
<tr>
<td>Linux</td>
<td>30</td>
</tr>
<tr>
<td>HP-UX 11i v2 PA-RISC and Itanium IA-64 with HP ServiceGuard Cluster</td>
<td>38</td>
</tr>
<tr>
<td>Microsoft Windows 2000</td>
<td>43</td>
</tr>
<tr>
<td><strong>Part II Upgrade the Database to Release 9.2</strong></td>
<td>44</td>
</tr>
<tr>
<td>2.1 Software Installation on Cluster Nodes</td>
<td>44</td>
</tr>
<tr>
<td>2.1.1 Creating of user ora&lt;dbsid&gt;</td>
<td>44</td>
</tr>
<tr>
<td>2.1.2 Configuration for r-commands</td>
<td>44</td>
</tr>
<tr>
<td>2.1.3 Environment Settings for user ora&lt;dbsid&gt;</td>
<td>45</td>
</tr>
<tr>
<td>2.1.4 Checking access permissions for user ora&lt;dbsid&gt;</td>
<td>45</td>
</tr>
<tr>
<td>2.1.5 Check for Required Disk Space</td>
<td>46</td>
</tr>
<tr>
<td>2.1.6 Checking Operating System Requirements</td>
<td>46</td>
</tr>
<tr>
<td>2.1.7 Run OUI to Install Oracle Software</td>
<td>47</td>
</tr>
<tr>
<td>2.1.8 Install latest Oracle Patchset and Patches</td>
<td>50</td>
</tr>
<tr>
<td>2.2 Database Upgrade to Release 9.2</td>
<td>51</td>
</tr>
<tr>
<td>2.2.1 Preparing the upgrade: Database Initialization file init&lt;dbsid&gt;.ora</td>
<td>51</td>
</tr>
<tr>
<td>2.2.2 Startup the database</td>
<td>51</td>
</tr>
<tr>
<td>2.2.3 Extending the SYSTEM Tablespace</td>
<td>52</td>
</tr>
<tr>
<td>2.2.4 Checking for Invalid Database Objects</td>
<td>52</td>
</tr>
<tr>
<td>2.2.5 Perform the upgrade</td>
<td>53</td>
</tr>
<tr>
<td>2.2.6 CATCLUST.SQL</td>
<td>53</td>
</tr>
<tr>
<td>2.3 Database Controlfile Parameters</td>
<td>54</td>
</tr>
<tr>
<td><strong>Part III  Adjusting the Environment</strong></td>
<td>57</td>
</tr>
<tr>
<td>3.1 Oracle Parameter Files</td>
<td>57</td>
</tr>
<tr>
<td>3.1.1 Modifying the Oracle Initialization Parameters for RAC</td>
<td>57</td>
</tr>
<tr>
<td>3.1.2 Oracle Password File</td>
<td>59</td>
</tr>
<tr>
<td>3.2 Network Configuration</td>
<td>60</td>
</tr>
<tr>
<td>3.2.1 Listener.ora</td>
<td>62</td>
</tr>
<tr>
<td>3.2.2 Tnsnames.ora</td>
<td>64</td>
</tr>
<tr>
<td>3.2.3 Sqlnet.ora</td>
<td>67</td>
</tr>
<tr>
<td>3.2.4 Create Database Links for SAP R/3</td>
<td>67</td>
</tr>
<tr>
<td>3.3 Database UNDO Management</td>
<td>68</td>
</tr>
<tr>
<td>3.3.1 Using Rollback – Segments</td>
<td>68</td>
</tr>
<tr>
<td>3.3.2 UNDO Tablespaces</td>
<td>69</td>
</tr>
<tr>
<td>3.4 Redo Log Groups and Members</td>
<td>70</td>
</tr>
<tr>
<td>3.5 SAP User Profiles</td>
<td>73</td>
</tr>
</tbody>
</table>
Preface

Oracle Real Application Clusters (RAC) will become a general available solution for the database backend in a SAP R/3 installation shortly. Using SAP with Oracle RAC provides high availability and scalability. The purpose of this document is to describe the best practices for migrating an existing SAP R/3 4.6D installation from a single instance Oracle database to a multiple instance Oracle RAC cluster database configuration.

The main focus of this white paper is on upgrading your single instance Oracle database to Oracle9i Release 2 Real Application Clusters and configuring your SAP R/3 system. This document covers the changes in the database structures and configuration as well as the modifications in the environment of the SAP system itself.

The preinstallation steps for setting up the underlying cluster hardware are also discussed as a prerequisite. However, because the cluster technology from the different hardware vendors is quickly evolving, this paper covers only examples from some vendors and, therefore, is not complete. The paper describes the recommended RAC configurations for a SAP installation that have been successfully tested. Please refer to the documentation of the respective platform and operating system vendor for details on the setup for your specific cluster hardware.

The chapters in this document provide a logical separation for the different aspects of the RAC migration:
- Hardware preparation
- Software installation
- Adoption of the environment

There are only a few dependencies in addition to the hardware setup of the cluster. Carefully planning the migration and preparing the modified files for the SAP R/3 environment and the Oracle instance profiles can reduce the amount of downtime.

For implementing Oracle RAC in an existing SAP R/3 system it makes no difference whether the original system is a 2-tier or a 3-tier configuration. In all examples and figures within this document, we refer to a 2-tier system.

System Requirements and Assumptions

A somewhat important expectation throughout this document is that for the completed Oracle9i RAC implementation, new or at least updated cluster hardware and operating systems will be used. Also, this paper assumes that the existing systems are running SAP R/3 4.6D with Oracle version 8.1.7 as the starting point. To optimally take advantage of the configuration described in this white paper, the new system should support cluster file system storage.

*Use of a cluster filesystem is the preferred storage solution for a SAP R/3 system. It is the implementation default defined by SAP. If a cluster filesystem is not available for a specific platform combination, or not desired for important reasons, check back with SAP on configuration details.*

The default location for the software installation of Oracle RAC is on a shared cluster filesystem. A shared $ORACLE_HOME is supported by Oracle on most platforms. In
this document, we mostly refer to an installation on a shared $ORACLE_HOME. This has been tested by SAP and is the preferred way for ease of installation, upgrade, administration, monitoring and supportability. The use of the SAP BR*Tools for a RAC installation is only supported by SAP when using a cluster filesystem. There are some caveats inherently given. The Oracle inventory will hold only one node. Tools like opatch will probably not detect that there is a cluster configuration with a shared $ORACLE_HOME. Updates and patches MUST be performed from the node holding the valid Oracle inventory. This has to be kept in mind.

Nevertheless it is possible to do a software installation on local filesystems on every node. The Oracle Universal Installer is capable to distribute the software and configuration files to all nodes. If your cluster filesystem or storage solution does not support a shared $ORACLE_HOME, the configuration of the SAP R/3 system differs from the examples given throughout this guide. You will find remarks on affected items.

On most platforms, the preferred location for $ORACLE_HOME is on a shared cluster filesystem. It is the reference implementation tested and certified by SAP for Oracle RAC and recommended as well.

The final assumption is for the datafiles and Oracle-managed files that belong to the database. For SAP installations the standard configuration places all database files on a filesystem. Most often these files will not reside on a cluster file system already in the original single instance system and will need to be copied or moved to a cluster file system. On the target system these files should have the same access path as in the original system.
Part I Configure the Hardware

Prerequisites: Checks on the existing database provided by SAP

Before the database files are transferred to the target system, verify that the Oracle database on the existing system is in the correct condition for the upgrade. Especially for this task, SAP provides two specific SQL scripts to perform some checks in advance of the Oracle9i database upgrade. The scripts are called CHECKS.SQL and PREMIG.SQL and they can be found on the first RDBMS CD from SAP for Oracle9i Release 9.2.0.

After these SQL scripts have been executed on the database installation of the original system, the database should be shut down. The files can then be copied or moved to the new cluster hardware for the RAC installation and the cluster file system.

To run the scripts, perform the following steps on the original database server as user ora<dbsid> as follows:

- Log on to the system as user ora<dbsid>
- cd <path to the SAP CD-ROM with RDBMS/SAP>
- Run sqlplus to connect to the database

```
sqlplus /nolog
SQL> connect / as sysdba
Connected.
@CHECKS.SQL
```

The script gives additional instructions regarding the upgrade, which must be followed before the upgrade on the new cluster system can be started. Note that the final instructions explain how to set the environment variable NLS_LANG. The value of NLS_LANG is required later on.

PREMIG.SQL is the second SQL script from SAP for checking requirements for the upgrade. This script can be found on the first Oracle RDBMS CD shipped together with the R/3 distribution in the directory SAP.

```
sqlplus /nolog
SQL> connect / as sysdba
Connected.
SQL> @PREMIG.SQL
```

This script does a backup of the database control files and storage adjustment for the system rollback segment.
Overview: Supported Configurations for SAP R/3

During the migration to Oracle 9i RAC within the SAP R/3 environment the original / source system configuration can be either 2-tier (SAP & Oracle database running on the same host) or 3-tier (SAP & Oracle database run on two different hosts). While, in principle this would also apply to the target system, an additional step for the 2-tier SAP / Oracle RAC system would be to install / configure the Standalone Enqueue Server (along with replication server) which is delivered by SAP.

The Oracle 9i RAC database provides for a highly available database due to the existence of the parallel/multiple instances. Hence within 3-tier configurations the failure of one database node would result in the SAP application servers failing over / reconnecting to the surviving database instance.

However if the SAP Central Instance (Message + Enqueue) runs on one of the database nodes itself (2-tier configuration), a failure of this node would automatically result in failure of database and the SAP application. In this case the locks held by the enqueue server would also be lost. This issue is addressed by installing the Standalone Enqueue Server on one node which then replicates the lock table (enqueues) to the other RAC node where the Enqueue replication server also runs.

Within the 3-tier environment since the SAP application / DB instances run on separate hosts the risk of the SAP application and the database failing at the same time is minimized. The Enqueue server still remains a SPOF even within 3-tier configurations running Oracle 9i RAC. It would therefore also make tremendous sense to use the SAP Replicated Enqueue server even within 3-tier SAP / RAC configurations although this is not mandatory.

Some important points to be considered for such a configuration:

1. The failover features for the Enqueue server / replication server has to be provided by the HA software. This means the startup / shutdown of the Enqueue server / replication server must be done by the HA software. The HA software is also responsible for detection of node failure which could either be software, hardware or network failure.

2. The HA Software used for the Replicated Enqueue must be compatible with the underlying clusterware / cluster file system software. What this means is that in event of failure there should be no conflict regarding choice of the ‘surviving node’ between the HA software for Enqueue replication and the software providing the clusterware / CFS functionality.
A typical configuration scenario for a mixed 2-tier / 3-tier SAP / RAC configuration with the enqueue replication service provided by SAP would look as under:
The next two figures show the supported configurations of Oracle9i RAC in a 2-tier and 3-tier SAP R/3 system.

Two Tier SAP R/3 and RAC

Three Tier SAP R/3 and RAC

All examples and figures within this document refer to a 2-tier system.
As of March 2005, some larger production installations are running using SAP R/3 4.6D / 4.7 Enterprise and Oracle9i RAC in a regular customer environment under real live conditions. Some restrictions apply to these installations, as not all tools, scripts, and utilities are already made completely aware of the different behaviors and new commands available for starting and stopping a clustered database. Adoption to Oracle 9i RAC is ongoing at SAP, so this may change in the future. For that, it is a good idea to check out the latest information regarding RAC certifications on SAP’s OSS system or the service marketplace.

SAP OSS note 527843 should be used as the entry point for RAC with SAP R/3.
This part describes the requirements for configuring HP Tru64 UNIX for Oracle9i Real Application Clusters.

Oracle9i Real Application Clusters requires HP Tru64 UNIX version 5.1. Additionally, in order to use the cluster file system some operating system patchkits are required. Use either 5.1 patchkits or 3 and the following HP patch kits:

- T64V5117-C0038308<unique id and mfg date>.tar
- TCV5117-C0009702<unique id and mfg date>.tar

or

- 5.1 patchkit 4 or higher.

Please also make sure to consult the latest product information from Oracle and HP on installing and administering Real Application Clusters.

For information regarding the installation and setup of Tru64 clusters check these external websites provided by HP:

- [http://tr64unix.compaq.com](http://tr64unix.compaq.com)
- [http://tr64unix.compaq.com/faqs/publications/pub_page/doc_list.html](http://tr64unix.compaq.com/faqs/publications/pub_page/doc_list.html)
- [http://tr64unix.compaq.com/faqs/publications/pub_page/cluster_list.html](http://tr64unix.compaq.com/faqs/publications/pub_page/cluster_list.html)

General information on known restrictions and prerequisites for HP Tru64 for use with SAP R/3 can be found in “SAP Basis Installation on UNIX: Oracle 46D Support Release”.

A database upgrade guide is also available by SAP that gives additional information on R/3 specific tasks as follows:

Database Upgrade Guide

*Upgrade to Oracle Version 9.2.0: UNIX*

Detailed information from Oracle Corporation is available from these sources:

*Oracle9i Release Notes Release 2 (9.2.0.1.0) for Compaq Tru64 UNIX (64-bit)*

*Oracle9i Installation Guide Release 2 for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*

*Oracle9i Administrator's Reference for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*

All these documents can be found on the Oracle Technology Network (OTN) under [http://otn.oracle.com/documentation/oracle9i.html](http://otn.oracle.com/documentation/oracle9i.html)
IBM AIX 5L with HACMP and GPFS

This part describes the requirements for configuring IBM AIX 5L, HACMP (High Availability Cluster Management Protocol) and GPFS (General Parallel File System) for Oracle9i Real Application Clusters.

The operating system release must be at least AIX 5.2 ML2. The OS kernel itself can run in 64-bit or 32-bit mode.
It is recommended to use HACMP 5.1 and GPFS 2.3.

All important APAR’s for RAC are part of maintenance level 2: The APAR’s in detail:

IY48488   AIX 5200-02 MAINTENANCE PACKAGE
IY45937   NOT ABLE TO CHANGE THE VALUE OF IPQMAXLEN USING NO COMMAND
IY37183   DYNAMIC TRACKING AND FAST I/O FAILURE OF FIBRE CHANNEL DEVICES

According to the GPFS Installation Guide, set

```
# no -o ipqmaxlen=512
```

This command must be executed after every reboot. Place an entry at the end of the file /etc/rc.net

For use of fibre channel adapter devices, set the attribute "fc_err_reco=fast_fail" for every device "fcsiXX":

```
# chdev -l 'fcsiXX' -a 'fc_err_reco=fast_fail'
```

Check that the attribute "dyntrk=no" is switched off.

GPFS v2.1 requires two additional APAR’s:

IY49503   BAD DATA FROM DIRECTIO WHEN REPLICATED DISKS DOWN
IY50460   GPFS I/O COMPLETION: UNEXPECTED DISK AVAILABILITY

Both APAR’s are part of the fileset

```
mmfs.gpfs.rtre.2.1.0.11  or higher.
```

To use HACMP, RSCT 2.2/AIX 5L must be installed on all nodes (Reliable Scalable Cluster Technology).

HACMP ES 4.5 and 5.1 (High Availability Cluster Management Protocol/Enhanced) are valid for the use with GPFS 2.1,2.2 and 2.3. Check for the correct versions.

The following additional OS packages are required for Oracle9i Release 2 (9.2.0.1.0):

- bos.adt.base
- bos.adt.libm
- bos.perf.perfstat
You should check the document *Oracle9i Release Notes Release 2 (9.2.0.1.0) for AIX-Based 5L Systems (64-bit)* on the Oracle Technology Network (OTN) under [http://otn.oracle.com/documentation/oracle9i.html](http://otn.oracle.com/documentation/oracle9i.html) before you start the installation of Oracle9i Real Application Clusters on AIX 5L.

General information on known restrictions and prerequisites on IBM AIX 5L for use with SAP R/3 can be found in “SAP Basis Installation on UNIX: Oracle 46D Support Release”.

A database upgrade guide is also available from SAP that gives additional information on R/3 specific tasks as follows:

Database Upgrade Guide
 Upgrade to Oracle Version 9.2.0: UNIX

Detailed information on setup and configuration of Oracle 9i RAC on top of the GPFS filesystem can be found in the following document:

*Deploying Oracle 9i RAC in IBM @server cluster 1600 with GPFS*

You can download this document by following the link given below:


The following part highlights some of the common pitfalls on configuring Oracle RAC on AIX 5L in conjunction with HACMP and GPFS. It is intended to be a short reminder for simple steps to follow for a hassle-free migration to a cluster solution.

The users ora<dbsid> and root must be members of the group hagsuser on every node. Make sure /etc/group is updated to reflect this as in the example given as follows:

```
/etc/group:  hagsuser:!:203:ora<dbsid>,root
dba:!:204:ora<dbsid>
oinstall:!:205:ora<dbsid>
```

Oracle requires you to use /usr/sbin/cluster/utilities/cldomain

Oracle software needs both read and write access permission on the HAGS socket /var/ha/soc/grpsvcsdsocket.<domain>.<domain>

The directory /var/opt/oracle has to be created on every node with read/write permission for the user ora<dbsid>. On AIX 5L, /var/opt does not exist by default. This directory path is required for the srvConfig.loc file, which the Installer creates during Real Application Clusters installation and configuration.

Either a shared raw device with about 100 MB space for storing the “Shared Configuration File Name” must be available, or the file on the cluster file system for holding the configuration data must already exist. The Oracle Universal Installer requires this “device” to be a raw device on AIX 5L if not already created.

*Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters* 13
To see all currently installed patches use the following command:

```
# /usr/sbin/instfix -i
```

You can verify the installed HACMP software with the "clverify" command.

```
# /usr/sbin/cluster/diag/clverify
```

At the "clverify>" prompt enter "software" then at the "clverify.software>" prompt enter "lpp". You should see a message similar to:

```
Checking AIX files for HACMP for AIX-specific modifications...
*/etc/inittab not configured for HACMP for AIX.
If IP Address Takeover is configured, or the Cluster Manager is to be started on boot, then /etc/inittab must contain the proper HACMP for AIX entries.
Command completed.
-------- Hit Return To Continue --------
```

Configure the cluster topology by the "smit hacmp" command:

```
# smit hacmp
```

Add the cluster definition:

```
# smit cm_config_cluster.add
```

Enter Cluster ID and Cluster Name.

Configure the nodes:

```
# smit cm_config_nodes.add
```

"Node Names" should be the hostnames of the nodes. All nodes participating in the cluster must be entered on this screen separated by a space.

Configure the network adapters:

```
# smit cm_config_adapters.add
```

Take care that the "Adapter IP Label" specified in this screen matches the entries in the "/etc/hosts" file. Otherwise the adapter will not map to a valid IP address and the cluster will not synchronize. The "Network Name" is an arbitrary name for the network configuration. All the adapters in this ether configuration should have the same "Network Name". This name is used to determine what adapters will be used in the event of an adapter failure.

All configurations are done from one node and then synchronized to the other participating nodes. The synchronization performs topology sanity checks as well as pushes the configuration data to each of the nodes in the cluster configuration. This task requires user equivalence for root on all nodes. Update the "/.rhosts" file accordingly by adding the user root to every node of the cluster. Be sure permissions on the "/.rhosts" file is 600.
Create volume groups to be shared concurrently on one node

    # smit vg

Select "Add a Volume Group". The "PHYSICAL VOLUME names" must be physical disks that are shared between the nodes. We do not want the volume group automatically activated at system startup because HACMP activates it. Also "Auto-varyon in Concurrent Mode?" should be set to "no" because HACMP varies it on in concurrent mode.

You must choose the major number to be sure the volume groups have the same major number in all the nodes (attention, before choosing this number, you must be sure it’s free on all the nodes).

To check all defined major number, type:

    # ls –al /dev/*

On this volume group, create all the logical volumes and file systems you need for the cluster database.

Use "importvg" to import the oracle_vg volume group on all of the other nodes.

On the first machine, type:

    # varyoffvg oracle_vg

On the other nodes, import the definition of the volume group using "smit vg" :

Add and configure a concurrent cluster resource group with these commands:

    # smit cm_add_grp
    # smit cm_cfg_res.select

To start and stop cluster nodes use

    # smit clstart.dialog
    # smit clstop.dialog

The command "clstat" can also be used to verify cluster health. The "clstat" program can take a while to update with the latest cluster information and at times does not work at all. Also you must have the "Startup Cluster Information Daemon?" set to "true" when starting cluster services. Use the following command to start "clstat":

    # /usr/es/sbin/cluster/clstat

One other way to check the cluster status is by querying the "snmpd" daemon with "snmpinfo":

    # /usr/sbin/snmpinfo -m get -o /usr/es/sbin/cluster/hacmp.defs -v ClusterSubstate.0

This should return "32":

    clusterSubState.0 = 32
If other values are returned from any node consult your IBM HACMP documentation or contact IBM support.

For all datafiles located on a GPFS filesystem, the Oracle 9i RAC database server automatically uses asynchronous DIRECT_IO for all IO – operations. This is the only supported operation mode of the database server and can’t be overwritten by init.ora parameters. As access to Oracle datafiles is possible for external programs during online database operation, the access to the datafile by these programs should bypass the GPFS pagepool. Otherwise the IO from the Oracle 9i RAC database server will be blocked as long as pagepool IO is ongoing and this may impose a performance penalty and introduce errors under rare conditions.

Use a hint for GPFS to bypass the pagepool and do DIRECT_IO when possible:

```
# mmchattr –D yes <filename>
```

The command must be executed for every Oracle datafile. The programs used to access Oracle datafiles during online operation of the database server must issue all IO operations page aligned and in multiple of the filesystem blocksize to allow GPFS the use of the hint. Otherwise the pagepool will be used still.

Best solution is to use only tools, which are DIRECT_IO aware. To give an example, use RMAN for online database backup. See also SAP note 815797.

Detailed information from Oracle Corporation is available from these sources:

*Oracle9i Release Notes Release 2 (9.2.0.1.0) for Sun Solaris (64-bit)*

*Oracle9i Installation Guide Release 2 for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*

*Oracle9i Administrator's Reference for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*

All these documents can be found on the Oracle Technology Network (OTN) under [http://otn.oracle.com/documentation/oracle9i.html](http://otn.oracle.com/documentation/oracle9i.html)
Sun Cluster 3.1

New with Sun Cluster 3.1 is the support of Sun QFS as a shared cluster filesystem for storing both Oracle and SAP binaries and all database files.

For SharedQFS you need Solaris 9 06/04 with the recommended patch cluster or above, QFS 4.3, Sun Cluster 3.1 Update 3 or above, Oracle 9.2.0.5 + Oracle patch 3566420

Differently, Sun GFS (Global File System) is supported for Oracle binary and archive logs only, but NOT for database files. We do not recommend the use of GFS.

Sun’s interconnect protocol, RSM, is supported with Oracle 9.2.0.5 and higher. Please see Metalink note 263699.1: The Use of RSM with Real Application Clusters and check the RAC/Sun certification matrix on recent updates. Also see the actual supported combinations of Sun servers, Sun or third-party storage products, Cluster interconnects, Public networks and Switch options in the certification matrix.

The use of RSM requires the following packages to be installed:

- SUNWrsmo
- SUNWrsmox (64bit)
- SUNWrsm
- SUNWrsmx (64bit)
- SUNWscrdt
- SUNWscrif

To see information about a package:

```
pkginfo -l <package name>
```

Use of RSM is enabled by setting following parameters in the init.ora (or spfile.ora) file:

```ini
_disable_sun_rsm=FALSE
_reliable_block_sends=TRUE
```

In addition, you must copy the new library `$ORACLE_HOME/lib/libskgxpu.so` over the existing `$ORACLE_HOME/lib/libskgxp9.so`

General information on known restrictions and prerequisites on Sun Solaris for use with SAP R/3 can be found in “SAP Basis Installation on UNIX: Oracle 46D Support Release”.

A database upgrade guide is also available from SAP that gives additional information on R/3 specific tasks as follows:

```
Database Upgrade Guide
Upgrade to Oracle Version 9.2.0: UNIX
```
The Sun Cluster 3.1 host system (node) installation process is completed in several major steps. The general process is:

- repartition boot disks to meet SunCluster 3.1.
- install the Solaris 9 Operating System on all nodes
- install the recommended patch cluster for Solaris 9 on all nodes
- install Sun Cluster 3.1 on the first cluster node
- install Sun Cluster 3.1 on the remaining nodes
- perform post installation checks and configuration

We recommend that you complete the Sun Cluster software installation on the first node. After that you can run `scinstall` in parallel on all remaining cluster nodes. The additional nodes are placed in install mode so they do not have a quorum vote.

Only the first node has a quorum vote. As the installation on each new node completes, each node reboots and comes up in install mode without a quorum vote. If you reboot the first node at this point, all the other nodes would panic because they cannot obtain a quorum. You can, however, reboot the second or later nodes freely. They should come up and join the cluster without errors.

Cluster nodes remain in install mode until you use the `scsetup` command to reset the install mode. You must perform post installation configuration to take the nodes out of install mode and also to establish quorum disk(s).

**Do not continue with the further preparations for installing and configuring RAC without checking that the basic cluster setup is correct.**

Install the Sun StorEdge QFS software as described in

*Sun StorEdge QFS and Sun StorEdge SAM-FS Installation and Configuration Guide.*

After that, create the Sun StorEdge QFS file systems.

**Create at least two filesystems, one for binaries, the other for the Oracle data files. This is important as the mount options for the filesystem will be different.**

Do a device id discovery and list the Sun Cluster Disk ID (DID) and the path to the devices:

```
# scdidadm –r
# scdidadm –L
```

From one node, use the format utility to layout the partitions on the devices you plan to use for QFS filesystems. After that, you must configure the newly created partitions into at least 2 Sun Storeedge QFS shared filesystems by placing the configuration entries to the file `/etc/opt/SUNWsamfs/mcf`. Repeat this on every node in the cluster.

On every node, add the appropriate mount options which are required to the file `/etc/opt/SUNWsamfs/samfs.cmd`.

**Set the options mh_write, qwrite and forcedirectio on all filesystems which hold Oracle datafiles. Do not set these options on filesystems intended for binaries.**
Validate the configuration on each node by the command

```
#/opt/SUNWsamfs/sbin/sam-fsd
```

Use the samd config command to signal that a new StorEdge configuration is available. Do this on each node.

```
#samd config
```

On all nodes, create the Sun StorEdge QFS shared hosts file for the filesystems. Be sure to specify the private interconnect names for the host IP addresses. There must be a separate file for every filesystem you want to create in the /etc/opt/SUNWsamfs/ directory.

From one node in the cluster, use the sammkfs –S command to create the Sun StorEdge QFS shared filesystem:

```
#sammkfs –S <filesystemname>
```

Create mountpoints for the newly created filesystems on every node and place an entry to /etc/vfstab to have the filesystems mounted automatically.

Mount the filesystems on the metadata server and check the configuration by use of the command `samsharefs`. Use this command to check the voluntary failover of the filesystems between the nodes.

Now configure the Sun Cluster resource type for all QFS filesystems created, and bring the resource groups online. Registration and configuration is done by use of the command `scrgadm`. Use `scswitch` to bring the resource groups online.

Detailed information from Oracle Corporation is available from these sources:

- *Oracle9i Release Notes Release 2 (9.2.0.1.0) for Sun Solaris (64-bit)*
- *Oracle9i Administrator's Reference for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*

All these documents can be found on the Oracle Technology Network (OTN) under [http://otn.oracle.com/documentation/oracle9i.html](http://otn.oracle.com/documentation/oracle9i.html)
Fujitsu PrimeCluster

This part describes the requirements for configuring Fujitsu PrimeCluster for Oracle9i Real Application Clusters.

For additional installation information on Network Appliance Filer, please see "Oracle9i RAC Release 2 - Installation on Solaris 8 with Fujitsu PRIMECLUSTER and Network Appliance Filer".

The Fujitsu PrimeCluster is capable to run the Solaris 8 or 9 operating system. For the list of supported servers, Fujitsu or third-party storage products, Cluster interconnects, Public networks, Switch options, check the RAC/Sun certification matrix for recent updates.

*SAP R/3 with Oracle RAC is tested and certified by SAP only in conjunction with the Network Appliance Filer storage solution and Veritas Advanced Cluster 3.5/4.0 (see chapter below).*

For additional installation information on NetApp Filer, please see "Oracle9i RAC Release 2 - Installation on Solaris 8 with Fujitsu PRIMECLUSTER and Network Appliance Filer"

*The PRIMECLUSTER Global File Services (GFS) component is not permitted. Note that at the time of writing Oracle has not certified use of the GFS with RAC*

Required software components to use SAP R/3 with RAC are

- Fujitsu PrimeCluster Cluster Foundation v. 4.0
  - SMAWrcadm
  - FSUNnet
  - SMAWskel
  - SMAWcf
  - SMAWvd
  - SMAWsf
  - FJSVcldbm
  - FJSVclapi
  - SMAWrefs

- Fujitsu PrimeCluster Web Based Admin View
  - FJSVwvcnf
  - FJSVwvbs
  - FJSVwvmpe

- Oracle Unix Distributed Lock Manager
  - SMAWrac

- Parallel Application Services (PAS)
  - SMAWpas

*Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters  20*
The following packages are required for Automatic Handling or Cluster Integrity issues (Split Brain):

- RCI Driver (CLDEV)
  - SMAWcldev

A volume manager is not required but is supported - such as Veritas (VxVM), Solstcie, or management through dkconfig.

The configuration will be completed using the Web-Based Admin View Tool GUI that will configure the basic cluster, the Cluster Integrity Monitor (CIM), and Cluster Interconnect Protocol (CIP). Web-Based Admin View should be installed on every node in the cluster and the Web-Based Admin View configuration must be done on all cluster nodes.

When setting up Web-Based Admin View in a 2-node system with no Cluster Console, you will have to assign one node as the Primary Management Server and one as the Secondary. When the Primary goes down the Secondary will take over. See PrimeCluster Installation guide for other configuration information.

Things to remember when configuring the cluster:

- Make sure all cluster nodes have the same patch levels
- Do not install any firmware-related patches without qualified assistance
- Always obtain the most current patch information
- Read all patch README notes carefully.

Specific Solaris and Fujitsu Operating System patches maybe required and it is recommended that Fujitsu be contacted to confirm the latest Solaris and Fujitsu patches that are required and supported for their hardware. To determine which patches have been installed, enter the following command:

```
$ showrev -p
```

It is required to have a dedicated set of NICs for the cluster interconnect and another, dedicated MIC (or multiple) for the connectivity to the NetApp filer. All database nodes can share the network segment to the NetApp filer.

At least Solaris 8 patches must be applied as described in SAP note #360438.

The following steps have to be repeated on every node in the cluster:

Make sure that the following option is set in the open boot prompt (OBP) prior to installing any software for the cluster:

```
ok setenv local-mac-address? true
```

The cluster interfaces must be configured using the following commands (e.g. with two Fujitsu GigaBit-Ethernet NICs), where **SID** needs to be replaced by the SAP System ID:
# cfconfig --s SIDCLUSTER /dev/fjge0 /dev/fjge1

Edit the file //etc/default/cluster.config and adapt the following lines:

CFCP “cfcp”
CHSH “cfsh”

After that the cluster foundation can be started:

# /etc/init.d/cf start

Check the interconnect configuration with the following command:

# cftool --l

All nodes of the cluster should be shown. The node names listed must be the names as they are returned by the `uname --n` command.

Monitor the traffic on the interconnect by using the `mipcstat` command:

# mipcstat -on
# mipcstat -S 5

Install the operating system-dependent (OSD) clusterware after the basic cluster is working properly. You must configure RAC to use the shared disk architecture of PrimeCluster. Conflicting access to the same data is controlled by means of the Oracle UNIX Distributed Lock Manager (UDLM).

The Fujitsu PrimCluster Cluster Foundation and PAS install contains the required udlm packages: Once installed, Oracle's interface with this, the Oracle UDLM, can be installed. The Oracle Unix Distributed Lock Manager (SMAWrac - supplied by Fujitsu and not Oracle) must be installed.

To check version information on any previously installed dlm package:

# pkginfo -l SMAWrac |grep PSTAMP

or

# pkginfo -l SMAWrac |grep VERSION
VERSION: rac_9201_64

You must apply the following steps to all cluster nodes.

Become super user. Change to the /var/spool/pkg/OracleRAC directory
Determine which file is required for your Oracle Version.

# cd /var/spool/pkg/OracleRAC
# ls
SMAWrac_9012_32_S8.tar SMAWrac_9013_32_S8.tar

Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters  22
Note there are a 64-bit and a 32-bit version for 9i Release 1 and 2. Ensure you choose the correct file for the Oracle version you will be installing. In this case we will be installing 64 bit Oracle 9.2.0.1.

Extract the Package from the required tar file:

```bash
# tar xvf SMAWrac_9201_64_S8.tar
```

Install the package by adding the package as root:

```bash
# pkgadd -d ./SMAWrac.ds SMAWrac
```

Note: this will create a directory - /opt/SMAM/SMAMrac/rac_9201_64

Run the prepare RAC script to place libskgxn2.so in to /opt/ORCLcluster/lib

```bash
# /opt/SMAM/SMAMWrac/rac_9201_64/prepare_rac.sh
```

Note: the file libskgxn2.so is used by the Oracle Universal Installer (OUI) to detect the cluster. If the correct file is not found in /opt/ORCLcluster/lib, then a Node Selection screen will not be presented by ./runInstaller and RAC will not be installed

For the database files you need to use the following mount options to access the network appliance filer:

```
rw,bg,hard,intr,rsize=32768,wsize=32768,proto=tcp,forcedirectio,noac,vers=3
```

*Warning: Do not use these mount options with executables!*

Enable the tcp option on the filer itself prior to mounting any datafiles:

```bash
# options nfs.tcp.enable on
```

Before you start with the installation of the Oracle software for 9i RAC, run a prepare script:

```bash
# /opt/SMAM/SMAMWrac/rac_9201_64/prepare_rac.sh
```

After completing the software installation with runInstaller, run the script to enable the configuration in the cluster:

```bash
# /opt/SMAM/SMAMWrac/rac_9201_64/enable_rac.sh
```

Detailed information from Oracle Corporation is available from these sources:

*Oracle9i Release Notes Release 2 (9.2.0.1.0) for Sun Solaris (64-bit)*

Oracle9i Administrator's Reference for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris

All these documents can be found on the Oracle Technology Network (OTN) under http://otn.oracle.com/documentation/oracle9i.html
Veritas Database Edition/Advanced Cluster (DBE/AC) for HP-UX PA-RISC and Sun Sparc Solaris

This part describes the requirements for configuring Veritas DBE/AC 3.5 for Oracle9i Real Application Clusters. The Veritas DBE/AC cluster solution is supported in combination with either the HP-UX - operating system for PA-RISC based systems or the Sun Solaris operating system for Sparc systems.

There is no requirement to install either an operating system or Oracle Universal Distributed Lock Manager (UDLM) to use DBE/AC with RAC.

Basic requirements for HP-UX are

- OS version >11.11 64bit. Oracle Version 9.x is a 64bit Application and is available for HP 64bit Operating systems only. Verify that the OS is running a 64bit Kernel. Only HP PA-RISC CPUs are currently supported.
- For HP servers, third-party storage products, Cluster interconnects, Public networks, Switch options, Memory, Swap & CPU requirements consult the operating system or hardware vendor and see the **RAC/HP certification matrix**.
- Private network connection can be any of the following (see the **RAC/HP certification matrix**):
  - HyperFabric
  - Ethernet or FDDI

Further requirements are described in the Release Notes of Veritas Database Edition / Advanced Cluster for Oracle9i RAC Release 3.5.

HP provides patch bundles at

http://www.software.hp.com/SUPPORT_PLUS

Individual patches can be downloaded from

http://itresourcecenter.hp.com/

To determine which operating system patches are installed, enter the following command:

```
# /usr/sbin/swlist -l patch
```

To determine which operating system bundles are installed, enter the following command:

```
# /usr/sbin/swlist -l bundle
```

Ensure that you are running the latest patch release. New patches will supercede patches listed. Oracle always recommends using the most current patch set from the OS vendor. You can also reference Document id 43507.1 in Oracle Metalink, which is a best effort to update customers on OS patches that are needed.

Basic requirements for Sun Solaris are

*Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters* 25

---

### Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters

**Basic requirements for Sun Solaris are**

- OS version >8.0 64bit. Oracle Version 9.x is a 64bit Application and is available for HP 64bit Operating systems only. Verify that the OS is running a 64bit Kernel. Only HP PA-RISC CPUs are currently supported.
- For HP servers, third-party storage products, Cluster interconnects, Public networks, Switch options, Memory, Swap & CPU requirements consult the operating system or hardware vendor and see the **RAC/HP certification matrix**.
- Private network connection can be any of the following (see the **RAC/HP certification matrix**):
  - HyperFabric
  - Ethernet or FDDI

Further requirements are described in the Release Notes of Veritas Database Edition / Advanced Cluster for Oracle9i RAC Release 3.5.

HP provides patch bundles at

http://www.software.hp.com/SUPPORT_PLUS

Individual patches can be downloaded from

http://itresourcecenter.hp.com/

To determine which operating system patches are installed, enter the following command:

```
# /usr/sbin/swlist -l patch
```

To determine which operating system bundles are installed, enter the following command:

```
# /usr/sbin/swlist -l bundle
```

Ensure that you are running the latest patch release. New patches will supercede patches listed. Oracle always recommends using the most current patch set from the OS vendor. You can also reference Document id 43507.1 in Oracle Metalink, which is a best effort to update customers on OS patches that are needed.
OS version 2.8 64-bit or higher. Verify that the OS is running a 64bit Kernel.

For Sun servers, third-party storage products, Cluster interconnects, Public networks, Switch options, Memory, Swap & CPU requirements consult the operating system or hardware vendor and see the RAC/Sun certification matrix

To determine which patches have been installed, enter the following commands:

```bash
$ showrev -p
```

Make sure all required patches as listed in the RAC/Sun certification matrix are applied to every node that forms the cluster.

General information on known restrictions and prerequisites for both operating systems for use with SAP R/3 can be found in “SAP Basis Installation on UNIX: Oracle 46D Support Release”.

A database upgrade guide is also available by SAP that gives additional information on R/3 specific tasks as follows:

Database Upgrade Guide  Upgrade to Oracle Version 9.2.0: UNIX

DBE/AC nodes might require patches in the following areas:

- HP-UX 11.11 Operating System (PA-RISC) patches
- Solaris Operating System (SPARC) Environment patches
- Storage Array interface firmware patches
- Storage Array disk drive firmware patches
- DBE/AC Maintenance Packs
- Veritas Volume Manager patches

Patch 113607-01 need not be applied as the patch was intended to provide cached Quick I/O (QIO) support with the Oracle Disk Manager (ODM) in a RAC environment, but this feature is currently not supported for the DBE/AC product (see Note:241060.1)

Some patches, such as those for Veritas Volume Manager cannot be installed until after the volume management software installation is completed.

Install DBE/AC Software

Setting the Shell Environment for root - ensure that a umask of at least 022 is used. If not, a "No Name for Node 0" may result on the installation as the /etc/llthosts* files require oracle read access.

Login as root and set the PATH and MANPATH variables as follows:

```bash
# PATH=/sbin:/usr/sbin:/usr/bin:/usr/lib/vxvm/bin:
/usr/lib/fs/vxfs:/opt/VRTSvxsfs/sbin:/opt/VRTSvcs/bin:
/opt/VRTSvcs/ops/bin:/opt/VRTSob/bin:$PATH:/opt/VRTS/bin

#export PATH
```
Run the install script:

```
# ./installDBAC
```

At the prompt, enter the names of the database nodes for installation. You need a Veritas license key for every node, as the install script requests these keys.

During the installation the program asks for the NICs to use for the private interconnect. At least 2 NICs are required on every node forming the cluster. The chosen NICs are exclusively reserved for the cluster interconnect.

The installation procedure configures the cluster manager (VCM), and optional SMTP email notification and SNMP trap notification. We recommend that you configure all components.

After reporting that the product is successfully installed, the utility tells you that you need to reboot all systems in the cluster:

Do NOT reboot the systems at this time. You must first run the vxinstall utility on each system before rebooting. After running installDBAC, verify that all the necessary packages are installed. Use the following command on each system:

```
# /opt/VRTSvcs/bin/chk_dbac_pkgs
```

Install the Cluster Manager GUI:

```
# cd dbedac/dbedac_software/database_ac_for_oracle9i/pkgs
# pkgadd -d.
```

choose 2 VRTSvcs

vea is the new GUI interface on UNIX and replaces vmsa from previous versions.

Configure VERITAS Volume Manager on all nodes:

Run vxinstall.

```
# vxinstall
```
Initialize one disk into rootdg. Root/boot disk encapsulation is not required.

Reboot the system when prompted or when done.

Verify Installation. All ports should be open on all nodes:

```
# /sbin/gabconfig -a
Port a gen 4a1c0001 membership 01
Port d gen 40100001 membership 01
Port f gen f1990002 membership 01
Port h gen d8850002 membership 01
Port o gen 243f0002 membership 01
Port q gen 28d10002 membership 01
Port v gen 1fc60002 membership 01
Port w gen 15ba0002 membership 01
# df -k
```

/dev/odm should be mounted

Verify that the same serial number for a LUN is returned on all paths to the LUN.

```
# vxfenadm -i /dev/rdsk/c2t13d0s2
```

To verify that the shared storage arrays support SCSI-3 persistent reservations and I/O Fencing, use the vxfentsthdw utility. Test at least one disk in each shared disk array. We recommend you test as many disks as possible in each array.

Note: Only supported storage devices can be used with I/O fencing of shared storage. Other disks, even if they pass the following test procedure, are not supported. Please see the VERITAS DBE/AC for Oracle9i RAC Release Notes for additional information about supported shared storage.

The utility, which you can run from one system in the cluster, tests the storage by setting SCSI-3 registrations on the disk you specify, verifying the registrations on the disk, and removing the registrations from the disk.

**Caution: The utility vxfentsthdw overwrites and destroys existing data on the disks.**

On one system, start the utility:

```
# cd /opt/VRTSvcs/ops/bin
# ./vxfentsthdw
```

Run the vxfentsthdw utility for each disk you intend to verify.

Create a disk group containing the coordinator disks named vxfencoorddg.

```
# vxdisksetup -i c1t9d0
# vxdisksetup -i c1t10d0
# vxdisksetup -i c1t11d0
# vxdg init vxfencoorddg <disks/luns>
# vxdg deport vxfencoorddg
# vxdg -t import vxfencoorddg
# vxdg deport vxfencoorddg
```

Create /etc/vxfendg and start fencing on all nodes in the cluster
# echo "vxfencoorddg" > /etc/vxfendg
#/etc/init.d/vxfen start

run gabconfig -a

should see port b membership and visible on other system.

After the Oracle software installation (runInstaller) is completed and any further Oracle patchsets applied some further library steps are required.

**Sun Solaris only:**

$ cp /opt/ORCLcluster/lib/9iR2/libskgxp92_64.so $ORACLE_HOME/lib/libskgxp9.so

or alternatively:

$ cp /opt/VRTSvcs/ops/lib/libskgxp92_64.so $ORACLE_HOME/lib/libskgxp9.so

$ cp /opt/ORCLcluster/lib/9iR2/libskgxn2_64.so $ORACLE_HOME/lib/libskgxn9.so

$ cd $ORACLE_HOME/lib
$ mv libodm9.so libodm9.so.old
$ ln -s /usr/lib/sparcv9/libodm.so libodm9.so

Note: If further Oracle patchsets are installed some of the libraries in use by vcsmm will be overwritten. To correct this you must always relink vcsmm after you relink Oracle. See [Note:230445.1](#) for further details.

To ensure that ODM is being used and the llt interconnect check in the alert logfile for the instances. For example:

Oracle instance running with ODM: VERITAS 3.5 ODM Library, Version 1.1
...
cluster interconnect IPC version:
VERITAS IPC 3.5MP1s+RP3a 10:36:03 Jun 24 2003
This part describes the requirements for Oracle9i Real Application Clusters running on the Linux platform.

"Important Note: Dell, Oracle and SAP request that customers do not use Dell Deployment CDs for the installation of Red Hat Linux, Oracle9i Real Application Clusters, and OCFS as part of ANY installation of SAP on Linux. SAP requirements for this kind of Oracle installation and the proper release versions are not currently maintained and supported as part of the Dell Deployment CDs install process, which, if used, could result in unsupported system configurations."

Please check the RAC/Linux certification matrix for information on currently supported hardware/software.

On Linux, the cluster software required to run Real Application Clusters is included in the Oracle distribution.

The Oracle Cluster Manager (ORACM) installation process includes some major tasks.

1. UNIX pre-installation tasks.
2. Configuring the shared disks
3. Run the Oracle Universal Installer to install the 9.2.0.1 ORACM (Oracle Cluster Manager)
4. Configure the hangcheck-timer.
5. Install version 2.2.0.18 of the Oracle Universal Installer
6. Run the 2.2.0.18 Oracle Universal Installer to patch the Oracle Cluster Manager (ORACM) to 9.2.0.4
7. Modify the ORACM configuration files to utilize the hangcheck-timer.
8. Start the ORACM (Oracle Cluster Manager)

For use with SAP R/3, the database archive files among with other directories have to reside on a filesystem, which is accessible from all nodes forming the cluster. The datafiles, controlfiles and redolog files may reside on raw devices.

You have to use either Oracle cluster filesystem for Linux (OCFS) or a NetApp Filer storage solution in conjunction with SAP R/3.

These steps need to be performed on ALL nodes.

First, on each node, create the Oracle group. Example:

```
# groupadd dba -g 501
```

Next, make the Oracle user's home directory. Example:

```
# mkdir -p /u01/home/oracle
```

On each node, create the Oracle user. Make sure that the Oracle user is part of the dba group. Example:

```
# useradd -c "orasid" -G dba -u 101 -m -d /u01/home/oracle -s /bin/csh oracle
```
On each node, create a mount point for the Oracle software installation on either a shared OCFS filesystem or on the NetApp filer. The oracle user should own this mount point and all of the directories below the mount point.

Depending on your Linux distribution, make sure inetd or xinetd is started on all nodes and that the ftp, telnet, shell and login (or rsh) services are enabled (see /etc/inetd.conf or /etc/xinetd.conf and /etc/xinetd.d). Example:

```bash
# more /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \ # unencrypted username/password pairs for authentication.
service telnet
{
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/sbin/in.telnetd
log_on_failure += USERID
disable = no
}
```

On the node from which you will run the Oracle Universal Installer, set up user equivalence by adding entries for all nodes in the cluster, including the local node, to the .rhosts file of the oracle account, or the /etc/hosts.equiv file.

Note: If you are prompted for a password, you have not given the oracle account the same attributes on all nodes. You must correct this because the Oracle Universal Installer cannot use the rcp command to copy Oracle products to the remote node's directories without user equivalence.

Verify operating system kernel parameters are set to appropriate levels. You will have to set the correct parameters during system startup, so include them in your startup script (startoracle_root.sh):

```bash
$ export SEMMSL=100
$ export SEMMNS=1000
$ export SEMOPM=100
$ export SEMMNI=100
$ echo $SEMMSL $SEMMNS $SEMOPM $SEMMNI > /proc/sys/kernel/sem
$ export SHMMAX=2147483648
$ echo $SHMMAX > /proc/sys/kernel/shmmax
```

Check these with:

```bash
$ cat /proc/sys/kernel/sem
$ cat /proc/sys/kernel/shmmax
```

You might want to increase the maximum number of file handles, include this in your startup script or use /etc/sysctl.conf:

```bash
$ echo 65536 > /proc/sys/fs/file-max
```
To allow your oracle processes to use these file handles, add the following to your oracle account login script (ex.: .profile)

```
$ ulimit -n 65536
```

Note: This will only allow you to set the soft limit as high as the hard limit. You might have to increase the hard limit on system level. This can be done by adding ulimit -Hn 65536 to /etc/initscript. You will have to reboot the system to make this active. Sample /etc/initscript:

```
ulimit -Hn 65536
eval exec "$4"
```

Create the directory /var/opt/oracle and set ownership to the oracle user.
Set the oracle user's umask to "022" in your ".profile" or ".login" file. Example:

```
$ umask 022
```

There is a verification script InstallPrep.sh available, which may be downloaded and run prior to the installation of Oracle Real Application Clusters. This script verifies that the system is configured correctly according to the Installation Guide. The output of the script will report any further tasks that need to be performed before successfully installing Oracle 9.x DataServer (RDBMS).

```
./InstallPrep.sh
```

For 9.2 Real Application Clusters on Linux, you can use either OCFS (Oracle Cluster Filesystem), RAW, or NFS (Red Hat and Network Appliance Only) for storage of Oracle database files.

For more information on setting up OCFS for RAC on Linux, see the following MetaLink Note:

Note 220178.1 - Installing and setting up ocfs on Linux - Basic Guide


For more information on setting up RAW for RAC on Linux, see the following MetaLink Note:

Note 246205.1 - Configuring Raw Devices for Real Application Clusters on Linux

For more information on setting up NFS for RAC on Linux, see the following MetaLink Note (Steps 1-6):

Note 210889.1 - RAC Installation with a NetApp Filer in Red Hat Linux Environment

Run the Oracle Universal Installer to install the 9.2.0.1 ORACM (Oracle Cluster Manager)

*These steps only need to be performed on the node that you are installing from (typically Node 1).*
If you are using OCFS or NFS for your shared storage, pre-create the quorum file and srvm file. Example:

```bash
# dd if=/dev/zero of=/ocfs/quorum.dbf bs=1M count=20
# dd if=/dev/zero of=/ocfs/srvm.dbf bs=1M count=100
# chown root:dba /ocfs/quorum.dbf
# chmod 664 /ocfs/quorum.dbf
# chown oracle:dba /ocfs/srvm.dbf
# chmod 664 /ocfs/srvm.dbf
```

Verify the Environment - Log off and log on as the oracle user to ensure all environment variables are set correctly. Use the following command to view them:

```bash
% env | more
```

Start the Oracle Universal Installer and install the RDBMS software - Follow these procedures to use the Oracle Universal Installer to install the Oracle Enterprise Edition and the Real Application Clusters software. Oracle 9i is supplied on multiple CD-ROM disks. During the installation process it is necessary to switch between the CD-ROMS. OUI will manage the switching between CDs.

Use the following commands to start the installer:

```bash
% cd /tmp
% /cdrom/runInstaller
```

Or cd to /stage/Disk1 and run ./runInstaller

Respond to the installer prompts as shown below:

At the "File Locations Screen", verify the destination listed is your ORACLE_HOME directory. Also enter a NAME to identify this ORACLE_HOME. The NAME can be anything.

At the "Available Products Screen", Check "Oracle Cluster Manager".

At the public node information screen, enter the public node names.

At the private node information screen, enter the interconnect node names.

Accept the default value (60000) for the watchdog parameter.

Enter the full name of the file or raw device you have created for the ORACM Quorum disk information.

**These steps need to be performed on ALL nodes.**

Some kernel versions include the hangcheck-timer with the kernel. You can check to see if your kernel contains the hangcheck-timer by running:

```
# /sbin/lsmod
```

Then you will see hangcheck-timer listed. Also verify that hangcheck-timer is starting in your /etc/rc.local file (on Red Hat) or /etc/init.d/boot.local (on United Linux). If you see hangcheck-timer listed in lsmod and in the rc.local file or boot.local, you can skip the next section.
If hangcheck-timer is not listed here and you are not using Red Hat Advanced Server, see the following note for information on obtaining the hangcheck-timer:

Note 232355.1 - Hangcheck Timer FAQ

If you are on Red Hat Advanced Server, you can either apply the latest errata version (> 12) or go to MetaLink – Patches and download patch 2594820.

The three RPMs will work for the e3 kernels in Red Hat Advanced Server 2.1 gold and the e8 kernels in the latest Red Hat Advanced Server 2.1 errata release. These RPMs are for Red Hat Advanced Server 2.1 kernels only.

Run the following command to install the module:

```
# rpm -ivh hangcheck-timer RPM name
```

If you have previously installed RAC on this cluster, remove or disable the mechanism that loads the softdog module at system start up, if that module is not used by other software on the node. This is necessary for subsequent steps in the installation process. This step may require log in as the root user. One method for setting up previous versions of Oracle Real Applications Clusters involved loading the softdog module in the `/etc/rc.local` (on Red Hat) or `/etc/init.d/boot.local` (on United Linux) file. If this method was used, then remove or comment out the following line in the file:

```
# /sbin/insmod softdog nowayout=0 soft_noboot=1 soft_margin=60
```

Append the following line to the `/etc/rc.local` file (on Red Hat) or `/etc/init.d/boot.local` (on United Linux):

```
# /sbin/insmod hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
```

Load the hangcheck-timer kernel module using the following command as root user:

```
#/sbin/insmod hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
```

Repeat the above steps on all Oracle Real Applications Clusters nodes where the kernel module needs to be installed.

Run `dmesg` after the module is loaded. Note the build number while running the command. The following is the relevant information output:

```
build 334adfa62c1a153a41bd68a787f8e0e9
```

The build number is required when making support calls.

Install Version 2.2.0.18 of the Oracle Universal Installer

First install the 2.2.0.12 OUI into your `ORACLE_HOME` by running the installer from the 9.2.0.1 cd or your original stage location for the 9.2.0.1 ORACM install.

Use the following commands to start the installer:

```
% cd /tmp
% /cdrom/runInstaller
```

Or cd to `/stage/Disk1` and run `./runInstaller`

At the "File Locations Screen", verify the destination listed is your `ORACLE_HOME`
directory and that the source directory is pointing to the products.jar from the 9.2.0.1 cd or staging location.
At the "Available Products Screen", check "Oracle 9i Database 9.2.0.1.0".
At the "Installation Types Screen", check "Custom".
At the "Available Database Components Screen", uncheck EVERYTHING (some options you cannot de-select initially but after de-selecting other options you can go back and de-select these options). This should only install the java runtime environment and the Oracle Universal Installer.
Run root.sh when prompted.

Next, download the 9.2.0.4 patchset from MetaLink - Patches:

Run the installer from the $ORACLE_HOME/bin directory:

```
$ cd $ORACLE_HOME/bin
$ ./runInstaller
```

At the "File Locations Screen", change the source location to the products.jar file in the 9.2.0.4 patchset location under Disk1/stage. Also verify the destination listed is your ORACLE_HOME directory.
On the "Available Products Screen", Check "Oracle Universal Installer 2.2.0.18.

Create the following symbolic link

```
$ cd $ORACLE_HOME/oui/bin/linux
$ ln -s libclntsh.so.9.0 libclntsh.so
```

Run the 2.2.0.18 Oracle Universal Installer to patch the Oracle Cluster Manager (ORACM) to 9.2.0.4

*These steps only need to be performed on the node that you are installing from (typically Node 1).*

Run the installer from the $ORACLE_HOME/bin directory:

```
$ cd $ORACLE_HOME/bin
$ ./runInstaller
```

At the "Available Products Screen", Check "Oracle9iR2 Cluster Manager 9.2.0.4.0".
At the public node information screen, enter the public node names.
At the private node information screen, enter the interconnect node names.

Modify the ORACM configuration files to utilize the hangcheck-timer.
*These steps need to be performed on ALL nodes.*
Modify the $ORACLE_HOME/oracm/admin/cmcfg.ora file:
Comment out the following lines:

```
# WatchdogSafetyMargin=5000
# WatchdogTimerMargin=60000
```

Add the following lines:

```
KernelModuleName=hangcheck-timer
```
Adjust the value of the MissCount line based on the sum of the hangcheck_tick and hangcheck_margin values. (> 210)

```
MissCount=210
```

Make sure that you can ping each of the names listed in the private and public node name sections from each node.

*Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters* 35
Verify that a valid CmDiskFile line exists in the following format:

*CmDiskFile=file or raw device name*

In the preceding command, the file or raw device must be valid. If a file is used but does not exist, then the file will be created if the base directory exists. If a raw device is used, then the raw device must exist and have the correct ownership and permissions. Sample cmcfg.ora file:

```
HeartBeat=15000
KernelModuleName=hangcheck-timer
ClusterName=Oracle Cluster Manager, version 9i
PollInterval=1000
MissCount=210
PrivateNodeNames=int-rac1 int-rac2
PublicNodeNames=rac1 rac2
ServicePort=9998
# WatchdogSafetyMargin=5000
# WatchdogTimerMargin=60000
CmDiskFile=/ocfs/quorum
HostName=int-rac1
```

Modify the SORACLE_HOME/oracm/bin/ocmstart.sh and comment out the following lines:

```
# watchdogd's default log file
# WATCHDOGD_LOG_FILE=$ORACLE_HOME/oracm/log/wdd.log

# watchdogd's default backup file
# WATCHDOGD_BAK_FILE=$ORACLE_HOME/oracm/log/wdd.log.bak

# Get arguments
# watchdogd_args=`grep '^watchdogd' $OCMARGS_FILE | sed -e 's+^watchdogd *++'`

# Check watchdogd's existance
# if watchdogd status | grep 'Watchdog daemon active' >/dev/null
# then
# echo 'ocmstart.sh: Error: watchdogd is already running'
# exit 1
# fi

# Backup the old watchdogd log
# if test -r $WATCHDOGD_LOG_FILE
# then
# mv $WATCHDOGD_LOG_FILE $WATCHDOGD_BAK_FILE
# fi

# Startup watchdogd
# echo watchdogd $watchdogd_args
# watchdogd $watchdogd_args
```

Comment out the watchdogd line in the SORACLE_HOME/admin/ocmargs.ora so that your file looks like this:

```
# Sample configuration file SORACLE_HOME/oracm/admin/ocmargs.ora
# watchdogd
```
oracm
norestart 1800

Make sure all of these changes have been made to all RAC nodes. More information on ORACM parameters can be found in the following note:

Note 222746.1 - RAC Linux 9.2: Configuration of cmcfg.ora and ocmargs.ora

These steps need to be performed on ALL nodes.

Cd to the $ORACLE_HOME/oracm/bin directory, change to the root user, and start the ORACM.

    $ cd $ORACLE_HOME/oracm/bin
    $ su root
    # ./ocmstart.sh
    oracm </dev/null 2>&1 >/u01/app/oracle/product/9.2.0/oracm/log/cm.out &

Verify that ORACM is running with the following:

    # ps -ef | grep oracm

You should see several oracm processes running.

All these documents can be found on the Oracle Technology Network (OTN) under http://otn.oracle.com/documentation/oracle9i.html
This part describes the requirements for Oracle9i Real Application Clusters running on HP-UX 11i v2 PA-RISC and Itanium IA-64 with HP ServiceGuard Cluster.

In this configuration, all Oracle datafiles reside on raw devices. Shared Oracle Home and the saparch directory holding archived redolog files have to be placed on a filesystem which is exported to other nodes via a HA-NFS package of HP ServiceGuard cluster.

The basic requirement for HP-UX is a OS version 11.23 or newer. This OS release is certified for Oracle 9i RAC with HP ServiceGuard SG/SGeRAC version 11.16. The operating system must be configured to run a 64-bit kernel. To determine if a 64-bit configuration is installed, use following command:

```
# /bin/getconf KERNEL_BITS
```

For HP servers, third-party storage products, Cluster interconnects, Public networks, Switch options, Memory, Swap & CPU requirements consult the operating system or hardware vendor and see the RAC/HP certification matrix.

See also the latest version of the Oracle9i Release Notes Release 2 (9.2.0.1.0) for HP 9000 Series HP-UX (64-bit)

Make shure that the operating system release version and all required patches match the latest information given there. The operating system information can by optained by:

```
# uname –a
```

HP provides patch bundles at http://www.software.hp.com/SUPPORT_PLUS
Individual patches can be downloaded from http://itresourcecenter.hp.com/

To determine which operating system patches are installed enter the following command:

```
#/usr/sbin/swlist -l patch
```

To determine which operating system bundles are installed enter the following command:

```
#/usr/sbin/swlist -l bundle
```

Ensure that you are running the lasted patch release. New patches will supercede patches listed. Oracle always recommends using the most current patch set from the OS vendor. You can also reference Document id 43507.1 in Oracle Metalink, which is a best effort to update customers on OS patches that are needed.

Private network connection can be any of the following (see the RAC/HP certification matrix):

HyperFabric or Gigabit Ethernet or FDDI
Note: Only UDP protocol is supported as interconnect protocol for Oracle 9i RAC with SAP R/3. HP’s HMP protocol must not be used.

For the heartbeat from one node to the other, a serial RS232 connection or a second ethernet connection can be used. If a second ethernet connection is used, this connection has to be configured as a standby network. This would be configured as a second ip address subnet.

If there is a standby network card, it will take over in the case of a problem with one of the primary network cards. This is only the case if the cards are the same card type.

For better I/O - performance on the raw devices holding the Oracle datafiles asynchronous I/O should be used. If not already there, the /dev/async character device must be created:

```
# /sbin/mknod /dev/async c 101 0x0
# chown oracle:dba /dev/async
# chmod 660 /dev/async
```

After that the async driver in the kernel must be configured using SAM:

```
SAM -> Kernel Configuration -> Kernel -> driver is called ‘asyncdsk’
```

You must generate the new kernel and reboot afterwards. If asynchronous I/O is enabled, two different kernel parameters needs to be configured using SAM:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_async_ports</td>
<td>number of ‘processes’ from init.ora + number of background processes. If max_async_ports is reached, subsequent processes will use synchronous I/O</td>
</tr>
<tr>
<td>aio_max_ops</td>
<td>2048 (default value). This parameter limits the maximum number of asynchronous I/O operations that can be queued at any time. The parameter should be monitored over time using glance.</td>
</tr>
</tbody>
</table>

As the Oracle datafiles reside on raw devices, the disk layout needs careful planning in advance. The LVM steps to configure all the disks (LUN’s) are mentioned here.

**Steps to configure the configure the disks and create logical volumes:**

Turn off automatic volume group activation to prevent shared volume groups from being activated automatically at boot time. If deactivated, the shared volume groups will be enabled with cluster startup. Modify the file /etc/lvmrc:

```
AUTO_VG_ACTIVATE = 0
```

A disk must be properly initialized before it is added into a volume group. Repeat the following command for every disk:

```
# pvcreate -f /dev/rdsk/cxtydz
```

Create the volume group directory with the character special file called group. If there is more than one volume group in the system, use the same minor number for all volume groups:
# mkdir /dev/vg_rac
# mknod /dev/vg_rac/group c 64 0x060000

Note: Create the volume group directory on all nodes in the cluster. Use the same minor number on all cluster nodes. The configuration of the volume group(s) needs to be imported on the other cluster nodes later. See below.

Create the PV-LINKS and extend the volume group:

    # vgcreate /dev/vg_rac /dev/dsk/c0t1d0 /dev/dsk/c1t0d0
    # vgextend /dev/vg_rac /dev/dsk/c1t0d1 /dev/dsk/c0t1d1

Repeat with vgextend until all disks needed for the volume group(s) are included.

Now the logical volumes for the oracle 9i database can be created:

    # lvcreate –i 10 –I 1024 –L 100 –n Name /dev/vg_rac

Where:
- `-i`: number of disks to stripe across
- `-I`: stripe size in kilobytes
- `-L`: size of logical volume in MB
- `-n`: name of logical volume = raw file

The SAP naming convention for use of raw devices should be followed for the RAC configuration also: It’s a common practise to have a directory `/sapraw` and create symploric links within this directory to all logical volumes. The link names represent the file names for the SAP R/3 database.

Keep in mind that for Automatic Undo Management one additional tablespace per instance is required. Also additional Online redo log files per instance are needed.

Verify that the volume groups are properly created and available:

    # strings /etc/lvmtab
    # vgdisplay –v /dev/vg_rac

Change permission and owner of the volume group and logical volumes:

    # chmod 777 /dev/vg_rac
    # chmod 660 /dev/vg_rac/r*
    # chown oracle:dba /dev/vg_rac/r*

**Steps to export and import the volume group to all nodes in the cluster:**

Deactivate the volume group:

    # vgchange –a n /dev/vg_rac

Create the map file for the volume group:

    # vgexport -v -p -s -m mapfile /dev/vg_rac

Copy the mapfile to all nodes in the cluster:
To import the volume group on the other nodes, import the mapfile on every node:

```
# vgimport -v -s -m mapfile /dev/vg_rac
```

To check whether the devices are imported, use

```
# strings /etc/lvmtab
```

Activation of the volume group and logical volumes on all nodes is done later after the configuration of the HP ServiceGuard Cluster.

**Steps to configure HP ServiceGuard Cluster**

Create a default cluster configuration file. This needs to be done one node only. It is important that every node in the cluster is specified in the command line with the –n option.

```
# cmquerycl –v –C /etc/cmcluster/rac.asc –n node_1 –n node_2
```

Edit the cluster configuration file /etc/cmcluster/rac.asc. It is assumed here that there is only one volume group /dev/vg_rac configured in the whole cluster and this volume group is also used to hold the lock volume group.

```
CLUSTER_NAME sap_rac_cluster_on_hp
FIRST_CLUSTER_LOCK /dev/vg_rac
FIRST_CLUSTER_PV /dev/dsk/c0t1d0
OPS_VOLUME_GROUP /dev/vg_rac
DLM_ENABLED NO
GMS_ENABLED NO
```

FIRST_CLUSTER_LOCK and FIRST_CLUSTER_PV are the cluster lock volume group and disk which are used as a quorum device. This volume group and disk must be accessible by all nodes. OPS_VOLUME_GROUP has an entry for all cluster aware volume groups. DLM_ENABLED and GMS_ENABLED are not needed for Oracle 9i RAC and should be set to NO.

Verify the cluster configuration file:

```
# cmcheckconf –v –C /etc/cmcluster/rac.asc
```

Activate the lock volume group. IMPORTANT: Do this on the configuration node ONLY.

```
# vgchange –a y /dev/vg_rac
```

Initialize the binary configuration file on the lock disk and distribute the configuration to all nodes in the cluster:

```
# cmapplyconf –v –C /etc/cmcluster/rac.asc
```

Deactivate the lock disk on the configuration node:
Basic cluster administration

To start the entire cluster:

```
# cmruncl –v
```

Or, on each node in the cluster, use:

```
# cmrunnode –v
```

Make all RAC volume groups and cluster lock volume groups sharable and cluster aware from the cluster configuration node:

```
# vgchange –S y –c y /dev/vg_rac
```

On all nodes, activate the volume group in shared mode in the cluster:

```
# vgchange –a s /dev/vg_rac
```

Check the cluster status with:

```
# cmviecl –v
```

Detailed information from Oracle Corporation is available from these sources:

- Oracle9i Release Notes Release 2 (9.2.0.1.0) for Sun Solaris (64-bit)
- Oracle9i Administrator's Reference for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris

All these documents can be found on the Oracle Technology Network (OTN) under [http://otn.oracle.com/documentation/oracle9i.html](http://otn.oracle.com/documentation/oracle9i.html)
Microsoft Windows 2000

This document is focused on various UNIX flavors only. For the configuration of SAP R/3 with Oracle RAC on a Microsoft Windows 2000 database server cluster, the following white paper is available:

*SAP R/3 4.x / Oracle RAC on Windows 2000
A “best practices guide”*

This guide gives a detailed description on all tasks, which are specific to the Windows 2000 operating system as well as the Oracle cluster file system (OCFS).

This document can be found either on the SAP Service Marketplace, the URL to the document is [http://service.sap.com/dbaora](http://service.sap.com/dbaora), or on the following Oracle Intranet Site [http://sapdev.de.oracle.com](http://sapdev.de.oracle.com) by following the link: Database->Real Application Clusters->Dokumentation.
Part II Upgrade the Database to Release 9.2

2.1 Software Installation on Cluster Nodes

The installation of the database software on the cluster nodes may require additional tasks. It is mandatory that you check the latest documentation on installing Oracle9i RAC.

Before starting the installation of the Oracle9i RAC software, the cluster software of the underlying operating system must be installed and operational. This is because the Oracle Universal Installer (OUI) requires a functional Cluster Manager (CM) for the detection of the nodes within the cluster, the automatic distribution (copying) of the software to the different nodes, and various other checks during the installation.

If the Oracle Universal Installer is started from the CD set delivered by SAP, a preconfigured response file for a single instance installation automatically kicks in which is not appropriate for a cluster configuration. Do not use this response file for Oracle software installation, as it will fail.

2.1.1 Creating of user ora<dbsid>

For a SAP R/3 system, the installation of the database software is performed as user ora<dbsid>. This user account is already created on an existing SAP R/3 system. Otherwise, it has to be created on every node in the cluster. For exact information on how to create a user account, consult the documentation from your operating system vendor. A comprehensive explanation on how to perform this task on the respective OS can also be found in the already mentioned document “Database Upgrade Guide: Upgrade to Oracle Version 9.2.0: UNIX” available from SAP. The user ora<dbsid> must be a member of the group dba.

2.1.2 Configuration for r-commands

The user ora<dbsid> and root must have the correct permissions to do a rlogin, rcp, and rsh to every node within the cluster without needing a password after the initial logon to one of the cluster nodes. This is achieved by a proper configuration of the files /etc/hosts and /etc/hosts.equiv as well as the file ~/.rhosts for the user root and ora<dbsid> in the respective home directories of these users. This is often a pitfall during the installation and may cause some tools to fail. By checking this carefully on every node in the cluster, the installation can proceed uninterrupted.
2.1.3 Environment Settings for user ora<dbsid>

The environment of the user ora<dbsid> should be set as follows:

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>&lt;hostname&gt;:0.0</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td>/oracle</td>
</tr>
<tr>
<td>ORACLE_SID</td>
<td>&lt;dbsid&gt;</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>/oracle/&lt;dbsid&gt;/920_64</td>
</tr>
<tr>
<td>NLS_LANG</td>
<td>&lt;refer to the output of script CHECKS.SQL&gt;</td>
</tr>
<tr>
<td>ORA_NLS33</td>
<td>$ORACLE_HOME/ocommon/nls/admin/data</td>
</tr>
<tr>
<td>SAPDATA_HOME</td>
<td>/oracle/&lt;dbsid&gt;</td>
</tr>
</tbody>
</table>

Depending on the operating system, the following variables need to be set:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX 5L</td>
<td>LIBPATH</td>
<td>$ORACLE_HOME/lib</td>
</tr>
<tr>
<td>Solaris,TRU64,Linux</td>
<td>LD_LIBRARY_PATH</td>
<td>$ORACLE_HOME/lib</td>
</tr>
<tr>
<td>HP-UX</td>
<td>SHLIB_PATH</td>
<td>$ORACLE_HOME/lib</td>
</tr>
</tbody>
</table>

The following environment variables should NOT be set:

- TWO_TASK
- ENV

All settings of the environment variables should be made permanent in the profile of the user ora<dbsid>. The PATH or path variable must also contain $ORACLE_HOME/bin.

2.1.4 Checking access permissions for user ora<dbsid>

As SAP uses the account ora<dbsid> for the upgrade process, this account needs write permission to the /oracle directory. This can be checked the following way on every node in the cluster:

As user ora<dbsid> (log on to the system as user ora<dbsid>), and enter the following commands:

```
touch /oracle/write_test
rm /oracle/write_test
```

If the user ora<dbsid> does not have write permissions, log on to the system as user root and issue the following commands:

```
chgrp dba /oracle
chmod 755 /oracle
```
2.1.5 Check for Required Disk Space

Depending on the actual operating system, the following space requirements must be observed to enable the upgrade process. Check that disk space in MB is available for the upgrade as follows:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>$ORACLE_HOME</th>
<th>/oracle/920 64</th>
<th>/tmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tru64</td>
<td>4000</td>
<td>3600</td>
<td>400</td>
</tr>
<tr>
<td>HP-UX 64-bit</td>
<td>3000</td>
<td>3000</td>
<td>400</td>
</tr>
<tr>
<td>Linux</td>
<td>2300</td>
<td>3000</td>
<td>400</td>
</tr>
<tr>
<td>Solaris 64-bit</td>
<td>3000</td>
<td>3000</td>
<td>400</td>
</tr>
<tr>
<td>AIX 5L</td>
<td>4000</td>
<td>3600</td>
<td>400</td>
</tr>
</tbody>
</table>

In addition, Oracle9i release 9.2.0 requires 160MB free space in the /oracle directory. [SAP note 939 922]

2.1.6 Checking Operating System Requirements

For all UNIX platforms, the Java Development Kit (JDK) version 1.3.1 or later must be installed on the system.

For more details on specific operating system requirements, including patches and hardware requirements consult the document Oracle9i Real Application Clusters Installation and Configuration Guide on the RDBMS CD1 for RAC. This documentation can also be found and downloaded from the Oracle Technology Network (OTN) at this URL:

http://otn.oracle.com/docs/products/oracle9i/index.html

Check out this URL for the latest updates also. Carefully read and follow the instructions given there for your specific hardware platform.
2.1.7 Run OUI to Install Oracle Software

Refer to the corresponding documentation on Oracle Universal Installer and how to start the installation for your platform and operating system. Only the Oracle software needs to be installed. No sample database, no network configuration, no additional wizard or agent is required. For a SAP installation, the software installation is performed as user ora<dbsid>. Log on to the system as user ora<dbsid>. Change to the directory where the Oracle software CD is mounted.

The Installer must present all available members of the cluster. **If the cluster node selection screen does not show all cluster members, do not start the installation!** Check the configuration of the cluster interconnect and the Cluster Manager before proceeding with the installation.

Example: Two-node cluster, where both nodes are displayed.

Choose only one node for software installation to a shared $ORACLE_HOME on the cluster filesystem. By selecting one node only, the installer does not copy the software to the remaining nodes.

If $ORACLE_HOME is on a local filesystem, all nodes forming the cluster must be selected.

**Note: SAP recommends a shared $ORACLE_HOME.**
For a RAC database installation, the information about the cluster configuration must be on a location accessible from all nodes in the cluster. The cluster configuration information resides in a “device” that can either be a raw device (disk) or a file located on a cluster file system. As all the database files and the Oracle control files are located on a cluster file system for a SAP R/3 installation, this cluster configuration device should also reside on a cluster file system. The recommendation and default for the name and the location for this file is $ORACLE_HOME/srvconfig. You cannot change the name for this device; however, you can specify a location.

In the screen displaying all available products, select the Enterprise Edition option 9.2.0.1.0. Check that the Oracle Real Application Clusters 9.2.0.1.0 option is selected. The next example shows a screenshot. In the final screen that provides a summary of all selected products and options, you can perform a last check. After this check, the installation of the Oracle database software can be started.

**Note:** If the location of the shared configuration file is on a cluster filesystem, this file has to be created in advance. Do a “touch <filename>” to create the file.
Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters
2.1.8 Install latest Oracle Patchset and Patches

After the basic software installation is completed, all newer patchsets and latest patches to the Oracle binaries can be installed already. As of the date of writing this document, patchset Oracle 9.2.0.4 is available on all supported platforms. Check Oracle OTN or Oracle Metalink for newer patchsets and use the newest patchset available for your platform.

Patchset binaries get installed using OUI. Specify the location of the products.jar file of the patchset in the OUI screen.

*Note: First upgrade OUI itself to the latest version before installing any other patch or patchset to the Oracle binaries.*

The minimum required version of OUI is always part of the patchset. There is no need to separately download just OUI software.

For the node selection the same recommendations as for the base installation apply: Advice is to use a shared Oracle home directory on a cluster filesystem for the software if possible. Use always the same node for running OUI. Perform the installation/upgrade just on one node. There is no need to distribute the binaries to the other nodes.

Apply any additional patches for your platform by using “opatch”. As this tool is capable to detect a RAC configuration in a cluster, don’t be confused if the tool reports that actual configuration is not a RAC system! It is because the Oracle Inventory holds just the one node, which was selected during software installation with OUI. Apply the patches anyway.

The current version of opatch also offers an option “-l” for locally patching on one node only. You may want to specify this option in addition.

If your $ORACLE_HOME is not on a shared cluster filesystem, you have to distribute patchset and patches to all nodes, of course.
2.2 Database Upgrade to Release 9.2

2.2.1 Preparing the upgrade: Database Initialization file init<dbsid>.ora

After the software is installed by OUI, the upgrade of the database itself to Oracle9i release 9.2 will be performed manually. The checks mentioned here are intended to accelerate the upgrade progress by making sure that all necessary prerequisites are met before the actual upgrade starts. Following the steps given in this part reduces the risk of errors during the upgrade process.

Remember that all the tasks for the upgrade to Oracle9i Release 9.2 must be done in a single instance environment.

Copy the file init<dbsid>.ora from the old (8.1.7) $ORACLE_HOME/dbs directory to the new $ORACLE_HOME/dbs directory. According to the naming rules used by most SAP installations, the new $ORACLE_HOME is /oracle/<dbsid>/920_64. If the ifile parameter is used in the old file, then any referenced file herein should be copied also.

Note that a check for the current parameter settings needs to be performed prior to the start of the instance. For a smooth, error free instance startup, all deprecated or no longer supported parameters should be commented out in the parameter file. The list below shows all parameters that are obsolete in Oracle release 9.2:

<table>
<thead>
<tr>
<th>seq process cache const</th>
<th>ops admin group</th>
</tr>
</thead>
<tbody>
<tr>
<td>row cache cursors</td>
<td>parallel min message pool</td>
</tr>
<tr>
<td>distributed transactions</td>
<td>close cached open cursors</td>
</tr>
<tr>
<td>max transaction branches</td>
<td>sort direct writes</td>
</tr>
<tr>
<td>distributed recovery connection hold time</td>
<td>sort write buffers</td>
</tr>
<tr>
<td>text enable</td>
<td>sort write buffer size</td>
</tr>
<tr>
<td>job queue interval</td>
<td>sort spacemap size</td>
</tr>
<tr>
<td>job queue keep connections</td>
<td>sort read fac</td>
</tr>
<tr>
<td>snapshot refresh processes</td>
<td>sort multiblock read count</td>
</tr>
<tr>
<td>snapshot refresh interval</td>
<td>always anti join</td>
</tr>
<tr>
<td>multi block read count</td>
<td>push join predicate</td>
</tr>
<tr>
<td>optimizer percent parallel</td>
<td>complex view merging</td>
</tr>
<tr>
<td>optimizer search limit</td>
<td>fast full scan enabled</td>
</tr>
<tr>
<td>parallel default max instances</td>
<td>parallel broadcast enabled</td>
</tr>
<tr>
<td>cache size threshold</td>
<td>always semi join</td>
</tr>
<tr>
<td>parallel server idle time</td>
<td>allow partial sn results</td>
</tr>
</tbody>
</table>

A list of these parameters can be found in the online documentation CD also.

2.2.2 Startup the database

After installing the Oracle software, the database can be started up on one node of the cluster using the executables of the new release.

Log on to the system as user ora<dbsid>. It is important that the environment of the user ora<dbsid> now reflects the changes to the new Oracle9i software release. Before starting the database with the new Oracle executables, these variables must be set or corrected as follows:
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE_HOME</td>
<td>/oracle/&lt;dbsid&gt;/920_64</td>
</tr>
<tr>
<td>ORACLE_SID</td>
<td>&lt;dbsid&gt;</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td>/oracle/&lt;dbsid&gt;</td>
</tr>
<tr>
<td>ORA_NLS</td>
<td>$ORACLE_HOME/ocommon/nls/admin/data</td>
</tr>
<tr>
<td>NLS_LANG</td>
<td>as set in the old system</td>
</tr>
</tbody>
</table>

Additionally, the $PATH variable should include the path to the Oracle executables. After the environment of the user ora<dbsid> is configured as shown, use SQL*Plus to bring up the database as follows:

```sql
sqlplus /nolog

SQL> connect / as sysdba
connected to an idle instance
SQL> startup migrate;
```

### 2.2.3. Extending the SYSTEM Tablespace

The free space in the SYSTEM tablespace must be at least 80MB for the upgrade. The check whether the actual amount of free space is sufficient can be obtained by issuing the following query from SQL*Plus as follows:

```sql
SQL> select sum(bytes), sum(blocks) from dba_free_space
    where tablespace_name = `SYSTEM';
```

If the size requirements are not met, preparations for additional space can be achieved by one of the following commands:

- Adding space by creating an additional file:

  ```sql
  SQL> alter tablespace system add datafile `<full-path-of-datafile>'
    size 80M autoextend on next 10M maxsize unlimited;
  ```

- Setting the AUTOEXEND feature on an existing file of the SYSTEM tablespace:

  ```sql
  SQL> alter database datafile `filename` autoextend on;
  ```

- Directly resizing an existing file:

  ```sql
  SQL> alter database datafile `filename’ resize <new filesize>;
  ```

### 2.2.4. Checking for Invalid Database Objects

Any invalid object within the database also remains invalid after the upgrade. The upgrade does not remove invalid objects. The following SQL statements can obtain a list of invalid objects:
SQL> spool /var/tmp/Check4InvObj.log
SQL> select count(*) from dba_objects where status = 'INVALID';
SQL> select comp_id, version, status from dba_registry;

If the output of the previous statements did not return 0, then invalid objects should be identified as follows:

SQL> select owner, object_name, object_type from dba_objects where status = 'INVALID';
SQL> spool off

2.2.5 Perform the upgrade

A SQL script performs the upgrade of the database to Oracle9i Release 9.2. Log on to the system as user ora<dbsid>. Change the current working directory to $ORACLE_HOME/rdbms/admin. The script u080107.sql runs the upgrade from Oracle version 8.1.7 to Oracle9i Release 9.2.

Note: The starting release for an upgrade to Oracle9i Release 9.2 must be Oracle Release 8.1.7 or later. This is normally true for a SAP R/3 installation of version 4.6D. If the starting release is not Oracle 8.1.7 for any reason, then first upgrade the database to Oracle Release 8.1.7!

sqlplus /nolog

SQL> connect / as sysdba
connected to an idle instance
SQL> startup migrate;

SQL> @u080107.sql

After this script has completed, the upgrade to 9.2 is finished. Now the changes for RAC can be done.

2.2.6 CATCLUST.SQL

Run $ORACLE_HOME/rdbms/admin/catclust.sql to create cluster related views in the data dictionary as follows:

SQL> @catclust.sql
2.3 Database Controlfile Parameters

For the planned RAC enabling of the existing database, the database parameters MAXINSTANCES, MAXLOGFILES and MAXLOGMEMBERS need special attention. These parameters are part of the Oracle control files and will have been specified during initial database creation. Most often, these parameters will not have changed since the first installation of the R/3 system.

This task of changing values within the controlfiles of the database is the only additional step, which typically needs to be performed for a RAC migration of the existing database. All the other steps are identical in the manual upgrade procedure for a single instance database.

MAXINSTANCES should be at least the number of nodes in the cluster, probably with some room for additional growth in the future.

MAXLOGFILES determines the highest number of redo logfile groups that can ever be created in the database. Keeping in mind the conventions used by SAP software for naming the logfile groups and members, this value should be set to a number greater than the highest GROUP value for any redo log file group.

MAXLOGMEMBERS specifies the maximum number of members, or identical copies, for a redo log file group. The default value for a SAP installation is 2.

If the current values are not sufficient for the planned configuration, a new set of controlfiles needs to be created. To do this, dump the current settings to a trace file as follows:

```
SQL> alter database backup controlfile to trace;
```

This generates a SQL statement together with the hints for generating a new set of controlfiles in the current database trace file. The location of the directory for this file can be determined by SQL*Plus as follows:

```
SQL> show parameter user_dump_dest
```

The default directory for this file is `/oracle/<SID>/saptrace/usertrace` if the naming conventions of SAP are used.

The most recent database trace file contains all necessary information. Save the relevant part of this trace file to a temporary file and modify the parameters to reasonable values.

This command gives a list:

```
ls -lr
```

The most recent log file is at the end of this listing. Copy the most recent trace file to a temporary file. Edit this file and correct, if necessary, the values for MAXLOGFILES, MAXLOGMEMBERS and MAXINSTANCES.
If the current parameters are already sufficient, then the creation of a new set of control files can be skipped. Otherwise, these parameters need to be adjusted. After applying the changes, follow the instructions given in the trace file for creating a new set of control files.

Example: Excerpt from a trace file from an early RAC installation at SAP

```sql
# Below are two sets of SQL statements, each of which creates a new control file and uses it to open the database. The first set opens the database with the NORESETLOGS option and should be used only if the current versions of all online logs are available. The second set opens the database with the RESETLOGS option and should be used if online logs are unavailable.
# The appropriate set of statements can be copied from the trace into a script file, edited as necessary, and executed when there is a need to re-create the control file.
#
# Set #1. NORESETLOGS case
#
# The following commands will create a new control file and use it to open the database.
# Data used by the recovery manager will be lost. Additional logs may be required for media recovery of offline data files. Use this only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "RAC" NORESETLOGS NOARCHIVELOG
  SET STANDBY TO MAXIMIZE PERFORMANCE
  MAXLOGFILES 255
  MAXLOGMEMBERS 3
  MAXDATAFILES 254
  MAXINSTANCES 50
  MAXLOGHISTORY 65535
LOGFILE
  GROUP 11 ("/oracle/RAC/origlogA/log_g11m1.dbf", "/oracle/RAC/mirrlogA/log_g11m2.dbf") SIZE 20M,
  GROUP 44 ("/oracle/RAC/origlogB/log_g44m1.dbf", "/oracle/RAC/mirrlogB/log_g44m2.dbf") SIZE 20M
-- STANDBY LOGFILE
DATAFILE
  "/oracle/RAC/sapdata1/system_1/system.data1",
  "/oracle/RAC/sapdata1/user1d_2/user1d.data2"
CHARACTER SET WE8DEC;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# Database can now be opened normally.
ALTER DATABASE OPEN;
```

---

Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters  55
Part III    Adjusting the Environment

3.1 Oracle Parameter Files

3.1.1 Modifying the Oracle Initialization Parameters for RAC

For proper operation in RAC cluster database environment, the database initialization parameters need to be changed. Within a single instance Oracle database, there is only one parameter file init<dbsid>.ora in the directory $ORACLE_HOME/dbs. When running a RAC cluster database configuration, every database instance needs a separate parameter file with name init<dbsid> or the default init.ora file for instance startup. In a SAP environment, SPFILE is now supported, and the modification to use it is rather simple to perform.

Note: With a shared $ORACLE_HOME, the server parameter file is on a shared location as well. If $ORACLE_HOME is on a local disk, you have to create a symbolic link to a shared location in the cluster. This storage location has to fulfill all prerequisites for high availability. A simple NFS mount point is not sufficient.

SAP uses $ORACLE_HOME/dbs for configuration files as well. Namely the configuration for the BR*SPACE toolset is located in this directory. Maintenance of these files as well as the maintenance of the Oracle server parameter file by SAP tools requires a shared location. A distributed (local to a node) configuration is not supported.

**Oracle and SAP recommend using a SPFILE in a RAC environment.**

The original initialization file init<dbsid>.ora must be copied from the old (Oracle 8.1.7) directory to the new directory $ORACLE_HOME/dbs. If the IFILE parameter is used in the old file, then any referenced file within should also be copied from the old location to the new location. Note that if this include mechanism was used for database initialization, then it should be avoided for the future. This recommendation is also from SAP and is valid for normal upgrades to Oracle 9i as well. This will ensure an easier transition to binary SPFILEs in future releases.

In the projected RAC environment, every Oracle database instance needs a private initialization file init<dbsid><thread>.ora. A good practice if you want to stay with instance specific files is to use this file only for the instance-specific settings and include the common parameters for all instances in a file named icommon.ora.

The instance-specific part of the initialization file should look like this:

```plaintext
instance_number = <thread>
thread = <thread>
instance_name = <dbsid><thread>
service_names = (<dbsid>, <dbsid><thread>
rollback_segments = ( ...)
ifile = <ORACLE_HOME>/dbs/icommon.ora
```

Note that the entry for the rollback_segments is specific to every instance. The rollback segments for the additional instances referenced in the instance profile must exist on the database. See section 3.4.1 Using Rollback – Segments on how to create and activate the rollback segments for new instances in the cluster.
Example: Assume the database SID is RAC. This is the first instance in the cluster as indicated by instance_number = 1. The thread number is 1 also. The instance_name is a unique name for every instance. It is composed from the old SID and the 3-digit instance_number. In this example, the name of the file is then initRAC001.ora. The rollback segments in this case are from the former single instance implementation.

```
instance_number = 1
thread = 1
instance_name = RAC001
service_names = (RAC, RAC001)
rollback_segments=(PRS_0,PRS_1,PRS_2, ... ,PRS_19)
ifile=/oracle/RAC/920_64/dbs/initcommon.ora
```

Hint: for automatic undo management using undo tablespaces, the entry rollback_segments has to be replaced by:

```
undo_management = auto
undo_tablespace=<tablespacename>
```

Also, an undo tablespace has to be created in the database for every instance. See section 3.4.2 UNDO Tablespace for an explanation of automatic UNDO management.

**Oracle and SAP recommend using UNDO tables in a RAC environment.**

The file icommon.ora contains all other initialization parameters, which are valid for all instances. In a RAC environment, after the upgrade, special attention is needed for the parameters shown in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>db_domain</code></td>
<td>WORLD</td>
<td>set to WORLD if no other value</td>
</tr>
<tr>
<td><code>compatible</code></td>
<td>9.2</td>
<td></td>
</tr>
<tr>
<td><code>cluster_database</code></td>
<td>true</td>
<td>Set this after the upgrade to 9i</td>
</tr>
<tr>
<td><code>max_commit_propagation_delay</code></td>
<td>1</td>
<td>block exchange behavior</td>
</tr>
<tr>
<td><code>remote_login_passwordfile</code></td>
<td>exclusive</td>
<td></td>
</tr>
<tr>
<td><code>remote_os_authent</code></td>
<td>true</td>
<td></td>
</tr>
<tr>
<td><code>local_listener</code></td>
<td>LISTENER</td>
<td></td>
</tr>
</tbody>
</table>

These parameters have all to be set explicitly in a RAC configuration for SAP R/3. As already mentioned in the section 2.2.1 Preparing the Upgrade: Database Initialization File init<dbsid>.ora, any deprecated parameters should be removed.

**Note:** Adjust the parameters shown in the previous table after the upgrade of the database. For the upgrade procedure itself, the old settings must be used.

Once the migration to a RAC enabled database is finished, you should create a single init.ora initialization file instead of multiple init<dbsid>.ora files. Instance specific values are tagged by the instance name in front of the parameter. The instance name is resolved from the ORACLE_SID environment variable. Refer to chapter 3.5 SAP User Profiles on how to establish a unique ORACLE_SID environment variable on every cluster node for user ora<dbsid>.

Example for an excerpt from a single init.ora file:
Any parameter which is valid for all instances is preceded by a “*”. You can omit the “*” for common parameters.

Remove any init<dsid>.ora or init<dsid><thread>.ora initialization files from the $ORACLE_HOME/dbs directory.

Create a binary SPFILE (with filename spfile.ora) by using sqlplus

```
connect "/ as sysdba"
SQL> startup nomount
SQL> create spfile from pfile;
SQL> alter database mount;
SQL> alter database open;
```

**Note: Do this after the migration to RAC succeeded.**

A binary SPFILE is a prerequisite for using Oracle Enterprise Manager in a RAC environment.

### 3.1.2 Oracle Password File

In the table shown above, the use of an Oracle password file is indicated by the parameter `remote_login_passwordfile`. A password file is probably not used in the old single instance SAP R/3 configuration, but it is now necessary for the RAC environment. The database software to identify the users with DBA and OPER privileges will use this file. To create a password file, log on as user ora<dsid>. If any other user account is used, the password file will be unusable. The file itself resides in the same directory as the database initialization files which is $ORACLE_HOME/dbs.

Use this command to create the file:

```
orapwd file=<fname> password=<password> entries=<users>
```

The parameter file specifies the filename and is mandatory. For the command line parameter `password` the password of user SYS has to be given. The parameter `entries` are optional and denote the number of distinct users with DBA and OPER privileges.
3.2 Network Configuration

Correct setup of the network configuration inside and outside the cluster nodes is vital for the smooth and error free operation of the SAP R/3 system as well as for the various tools for administration. Depending on the underlying operating system, there are several options for the file location of the Oracle network configuration files. The assumption within this document is that these files are located on a cluster file system. This is especially true if the $ORACLE_HOME/network/admin directory is on a clustered file system. In this case, all the nodes in the cluster have access to the same directory for the Oracle network configuration. The basic approach is to make this directory available to present individual copies of single files to every node in the cluster. This is done by creating a symbolic link back to a directory on a local disk or by using cdsl (context dependent symbolic link, on HP Tru64) on every node. Therefore, every node has a local copy of the files within this directory. The same applies for trace and log files.

Note: Skip creating symbolic links if $ORACLE_HOME is local (not shared)

The next table shows which files and directories must reside on a local file system on every cluster node. A generic solution for all UNIX systems is to create symbolic links from the cluster file system to the local file system. HP Tru64 provides a feature called context dependent symbolic links that can be used as a preferred method on such systems.

<table>
<thead>
<tr>
<th>file or directory</th>
<th>symbolic link to local file</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ORACLE_HOME/network/admin/tsnames.ora</td>
<td>/cluster/oracle/network/admin/tsnames.ora</td>
</tr>
<tr>
<td>$ORACLE_HOME/network/log</td>
<td>/cluster/oracle/network/log</td>
</tr>
<tr>
<td>$ORACLE_HOME/network/trace</td>
<td>/cluster/oracle/network/trace</td>
</tr>
</tbody>
</table>

Example for creating the symbolic links from a local disk to the network configuration in a shared Oracle home directory:

mkdir /cluster/oracle/network/admin
mkdir /cluster/oracle/network/log
mkdir /cluster/oracle/network/trace
chmod 664 /cluster/oracle/network/admin
chmod 664 /cluster/oracle/network/log
chmod 664 /cluster/oracle/network/trace
chown ora<dbsid>:dba /cluster/oracle/network/admin
chown ora<dbsid>:dba /cluster/oracle/network/log
chown ora<dbsid>:dba /cluster/oracle/network/trace
cp $ORACLE_HOME/network/admin/tsnames.ora /cluster/oracle/network/admin

vi /cluster/oracle/network/admin/tsnames.ora
make all necessary changes

ln -s /cluster/oracle/network/admin/tsnames.ora
$ORACLE_HOME/network/admin/tsnames.ora
ln –s /cluster/oracle/network/log  $ORACLE_HOME/network/log
ln –s dba /cluster/oracle/network/trace $ORACLE_HOME/network/trace
Repeat this on all nodes in the cluster.
3.2.1 Listener.ora

On every cluster node that runs an Oracle database instance, there is also a listener process to satisfy the connection requests from the clients. The listener process is not started automatically at instance startup. It must be started either manually as user ora<dbname> via the command:

```
lsnrctl start
```

or by a script during system startup.

The connection method used in a SAP environment is called “client side load balancing”, where a client chooses the listener and database instance to connect to that is specified in an address list in tnsnames. The local listener running on a cluster node only has knowledge of the local instance, as the database instances do not register remotely. The following figure shows the connection setup.

![Connection setup diagram](image)
Every node in the cluster can share the listener.ora configuration file. There is no need to have a separate file for every listener process. In the simplest case, the listener.ora file should look like the example given. The following example shows a configuration with four instances. The SID_NAME entries correspond to the instance names.

```
LISTENER=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
   )
 )
SID_LIST_LISTENER =
 (SID_LIST =
   (SID_DESC =
     (ORACLE_HOME = /oracle/RAC/920_64)
     (SID_NAME = RAC001)
   )
   (SID_DESC =
     (ORACLE_HOME = /oracle/RAC/920_64)
     (SID_NAME = RAC002)
   )
   (SID_DESC =
     (ORACLE_HOME = /oracle/RAC/920_64)
     (SID_NAME = RAC003)
   )
   (SID_DESC =
     (ORACLE_HOME = /oracle/RAC/920_64)
     (SID_NAME = RAC004)
   )
 )
```

Note that the host entry is left blank in this example. In this case, the value for HOST is substituted automatically by the local hostname of the cluster node.

Note: If the listener configuration file is maintained using network configuration tools, then the ADDRESS_LIST has to specify every HOST explicitly. Otherwise, the tools may fail. For example:

```
LISTENER=
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)(HOST=node_1)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=node_2)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=node_3)(PORT=1521))
     (ADDRESS=(PROTOCOL=tcp)(HOST=node_n)(PORT=1521))
   )
 )
SID_LIST_LISTENER =
 (SID_LIST =
   . . . . . . . .
 )
```
3.2.2 Tnsnames.ora

The SAP clients inside and outside of the cluster require an individual setup for the network configuration. The reason for this is to establish load balancing for the database instances in the cluster. The order of ADDRESS entries in the ADDRESS_LIST determines how the SAP clients try to connect to the database. The setup shown here gives an example for node_1 and node_2 of an n-way cluster. The failover in this case is a round robin behavior only if the Oracle instance should fail for any reason. In general, a configuration for the SAP clients should keep in mind that all work processes from one SAP instance connect to the same RAC instance on the cluster. In a 2-tier environment, this should be the same cluster node for SAP and RAC.

You need to specify the <dbsid> as a service that is presented by all database instances in the cluster. This is accomplished by the parameters given to the CONNECT_DATA section. A database instance registers itself at the listener process providing one or more service names. The given service names have been specified in the init.ora file or spfile.ora. Section „3.1.1 Modifying the Oracle Initialization Parameters for RAC“ mentioned this. This service entry is used by the SAP work processes to connect to the database on any instance and necessary for the failover to work.

Additionally, the tools from SAP like sapdba, brconnect, brbackup …. (up to SAP Rel.4.6C) as well as the new BR*SPACE toolset (available with Rel 6.20) require to connect to a specific database instance for administrative purpose (e.g. starting and stopping a single database instance). For access to a specific instance, the instance name is used in the CONNECT_DATA section. In a RAC configuration with SAP R/3, the instance name is always the database name and the thread number.

The LISTENER entry is needed because SAP uses port 1527 by default. If you don’t specify a LISTENER entry for the local_listener parameter in file init.ora or spfile.ora, service registration will not work as Oracle uses port 1521 as the default value. If you don’t give a hostname in the ADDRESS section, the local hostname is automatically substituted. By leaving this entry blank, all database instances are able to use the same value for the local_listener parameter.
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = )(PORT = 1527))
  )

<dbsid> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = tcp)(HOST = <node_1>)(PORT = 1527))
      (ADDRESS = (PROTOCOL = tcp)(HOST = <node_2>)(PORT = 1527))
      ...
      (ADDRESS = (PROTOCOL = tcp)(HOST = <node_n>)(PORT = 1527))
    )
    (LOAD_BALANCE=OFF)
    (FAILOVER=ON)
  )
  (CONNECT_DATA =
    (SERVER=DEDICATED)
    (SERVICE_NAME=<dbsid>.WORLD)
    (FAILOVER_MODE=)
    (METHOD=BASIC)
  )
)

<dbsid>001 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node_1)(PORT = 1527))
    )
    (CONNECT_DATA =
      (SID = <dbsid>001)
    )
  )
)

<dbsid>002 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node_2)(PORT = 1527))
    )
    (CONNECT_DATA =
      (SID = HP1002)
    )
  )
)

<dbsid>00n =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node_n)(PORT = 1527))
    )
    (CONNECT_DATA =
      (SID = <dbsid>00n)
    )
  )
Example: local tnsnames.ora on node_2

LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = node_2)(PORT = 1527))
  )

<dbname> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=tcp)(HOST=node_2)(PORT=1521)
      (ADDRESS=(PROTOCOL=tcp)(HOST=node_3)(PORT=1521)
      ...
      (ADDRESS=(PROTOCOL=tcp)(HOST=node_n)(PORT=1521)
      (ADDRESS=(PROTOCOL=tcp)(HOST=node_1)(PORT=1521)
      (LOAD_BALANCE=OFF)
      (FAILOVER=ON)
    )
    (CONNECT_DATA =
      (SERVER=DEDICATED)
      (SERVICE_NAME=<dbname>.WORLD)
      (FAILOVER_MODE =
        (TYPE=SELECT)
        (METHOD=BASIC)
      )
    )
  )

<dbname>001 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node_1)(PORT = 1527))
    )
  (CONNECT_DATA =
    (SID = <dbname>001)
  )

<dbname>002 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node_2)(PORT = 1527))
    )
  (CONNECT_DATA =
    (SID = HP1002)
  )

<dbname>00n =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node_n)(PORT = 1527))
    )
  (CONNECT_DATA =
    (SID = <dbname>00n)
  )

Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters  66
The file tnsnames.ora has to be maintained separately on every node in the cluster. Log on to every node and go to the directory with the local copy of the file. The recommended location is /cluster/oracle/network on UNIX in general or cdsl on HP Tru64. Change the order of the HOST entries in the ADDRESS_LIST so that load balancing can occur.

3.2.3 Sqlnet.ora

The file sqlnet.ora needs no special attention for RAC. It can be left as is, since it is the same for a single instance database. Example:

```
################
# Filename......: sqlnet.ora
# Created........: created by SAP AG, R/3 Rel. >= 4.0A
# Name..........:
# Date...........:
################
AUTOMATIC_IPC = ON
TRACE_LEVEL_CLIENT = OFF
DEFAULT_DOMAIN = WORLD
```

Set AUTOMATIC_IPC = ON if there are SAP R/3 instances running on the database cluster nodes also. In a Three-Tier environment this parameter has no effect.

3.2.4 Create Database Links for SAP R/3

To a certain extent, SAP R/3 software is already prepared for a clustered database solution! To enable some monitoring or administrative tasks on a cluster database from within the R/3 system, database links between the instances must be defined.

Use SQL*Plus to create these links between the instances. In the using-clause of the create statement, the net service name for this instance is expected as it is specified in the tnsnames.ora network configuration file. The link name itself is the instance name of the local instance. The statement has to be repeated for all instances.

```
create public database link instance_name_1 using instance_name_1;
create public database link instance_name_2 using instance_name_2;
                   ...                               ...
create public database link instance_name_n using instance_name_n;
```

SAP transaction SM66 (Global Workprocess Overview) pops up a window in the SAPGUI if the link definition was issued wrong.
3.3 Database UNDO Management

3.3.1 Using Rollback – Segments

As the RAC database, like the original single instance database, operates using rollback segments, these rollback segments need to be specific for every instance in the cluster. Additionally, the rollback segments themselves should be contained in a separate tablespace for every instance.

First, the additional tablespaces need to be created. The names for these tablespaces extend the naming conventions used by SAP: In a single instance installation the name is PSAPROLL. For every additional database instance in the cluster, the name for the additional tablespace should be PSAPROLL_<instance_number>.

The same applies to the name of the rollback segments for these additional tablespaces. The single instance name is PRS_0, PRS_1 ..., PRS_n and so on. For all additional rollback segments for the database cluster, the naming convention is PRS_0_xxx, PRS_1_xxx ... PRS_n_xxx, where xxx is replaced by the instance_number. The following example shows how a SQL script can achieve this task. This example adds two tablespaces PSAPROLL_002 and PSAPROLL_003 to the database.

```
connect / as sysdba
create tablespace 'PSAPROLL_002' datafile
'/oracle/<dbsid>/sapdataX/roll_002_1/roll_002.data1' size 650M
reuse default storage (initial 1M next 1M PCTINCREASE 0) online;
create rollback segment 'PRS_0_002' storage (initial 1048576 next
1064960 minextents 2 maxextents 300) tablespace 'PSAPROLL_002';
create rollback segment 'PRS_1_002' storage (initial 1048576 next
1064960 minextents 2 maxextents 300) tablespace 'PSAPROLL_002';
create rollback segment 'PRS_2_002' storage (initial 1048576 next
1064960 minextents 2 maxextents 300) tablespace 'PSAPROLL_002';
............................
create rollback segment 'PRS_n_002' storage (initial 1048576 next
1064960 minextents 2 maxextents 300) tablespace 'PSAPROLL_002';
create tablespace 'PSAPROLL_003' datafile
'/oracle/<dbsid>/sapdataX/roll_003_1/roll_003.data1' size 650M
reuse default storage (initial 1M next 1M PCTINCREASE 0) online;
create rollback segment 'PRS_0_003' storage (initial 1048576 next
1064960 minextents 2 maxextents 300) tablespace 'PSAPROLL_003';
............................
create rollback segment 'PRS_n_003' storage (initial 1048576 next
1064960 minextents 2 maxextents 300) tablespace 'PSAPROLL_003';
```

Repeat this for as many instances as required.

Note: Using rollback segments for undo management requires that these private rollback segments must be specified in the instance-specific part of the database initialization file.
Do not change the name of the first tablespace PSAPROLL. If this name changes, additional adjustments within SAP R/3 software are necessary. Errors may occur during execution of SAP R/3 by some rather seldom used R/3 programs.

3.3.2 UNDO Tablespaces

New with Oracle9i Release 2 is the use of special UNDO tablespaces for automatic undo management, making rollback segments obsolete.

*Oracle and SAP recommend using UNDO tablespaces in a RAC environment.*

Preparation for the use of automatic undo management for a SAP installation should be done by following certain naming conventions:

Name of the tablespaces should be:
PSAPUNDO, PSAPUNDO_002, .. , PSAPUNDO_n, depending on the number of instances in the cluster. Preferably, the datafiles for these tablespaces should be located on separate disks.

Creating the tablespaces can also be done as in the following example SQL script:

```sql
create undo tablespace PSAPUNDO datafile '/oracle/<dbsid>/sapdataX/undo/undo.data1 ' size 1000m reuse;
create undo tablespace PSAPUNDO_002 datafile '/oracle/<dbsid>/sapdataX/undo_002/undo_002.data1 ' size 1000m reuse;
create undo tablespace PSAPUNDO_00n datafile '/oracle/<dbsid>/sapdata4/undo_00n/undo_00n.data1 ' size 1000m reuse;
```

Note: Leveraging automatic undo management also requires changes to the initialization file for the instance. The entries for the rollback segments need to be deleted or commented out. Adding the lines:

```sql
undo_management = auto
undo_tablespace = PSAPUNDO
```

to the instance-specific initialization files is required.

Use of automatic undo management is mutually exclusive with the use of rollback segments for the instances in the cluster. It is not allowed to use a mix of both methods. *Either none or all instances in the cluster must be configured to use automatic undo tablespaces.*

Note: If you decide to use UNDO tablespaces, you have to drop all the tablespace(s) holding the old rollback segments. You cannot set these tablespaces in offline mode, as the database server may hang.
3.4 Redo Log Groups and Members

In a standard SAP installation, there are four groups of Oracle transaction log files (redo log files). By default, each group contains one original and one mirrored redo log file. If mirroring of redo log files is done by hardware or operating system support, then each group will consist of one original redo log file only. The default layout in a single instance database installation is organized the following way:

- **GROUP 101 (redo1)**
  /oracle/<dbsid>/origlogA/log_g101m1.dbf
  /oracle/<dbsid>/mirrlogA/log_g101m2.dbf

- **GROUP 102 (redo2)**
  /oracle/<dbsid>/origlogB/log_g102m1.dbf
  /oracle/<dbsid>/mirrlogB/log_g102m2.dbf

- **GROUP 103 (redo3)**
  /oracle/<dbsid>/origlogA/log_g103m1.dbf
  /oracle/<dbsid>/mirrlogA/log_g103m2.dbf

- **GROUP 104 (redo4)**
  /oracle/<dbsid>/origlogB/log_g104m1.dbf
  /oracle/<dbsid>/mirrlogB/log_g104m2.dbf

The log files are periodically written from redo log log_g101m?.dbf to redo log log_g104m?.dbf. The redo log that is written and the redo log that is archived always belong to a different set: GROUP 101 and GROUP 103 belong to set A, GROUP 102 and GROUP 104 belong to set B.

In an Oracle RAC configuration with multiple instances, every database instance needs its own group of redo log files. For these additional log file groups, the naming conventions are changed slightly. The second instance (thread2) should use a redo log set A containing GROUP 21 and GROUP 23, and a set B containing GROUP 22 and GROUP 24. The third instance will then use set A and B and so on.

Creating additional redo log files for the new database instances can be performed with a simple SQL script as shown in the following example for an Oracle RAC solution with four database instances:

```
alter database add logfile thread 1 group 11
  (´/oracle/<dbsid>/origlogA/log_g11m1.dbf´,
   ´/oracle/<dbsid>/mirrlogA/log_g11m2.dbf´) size 200M reuse;
alter database add logfile thread 1 group 12
  (´/oracle/<dbsid>/origlogB/log_g12m1.dbf´,
   ´/oracle/<dbsid>/mirrlogB/log_g12m2.dbf´) size 200M reuse;
alter database add logfile thread 1 group 13
  (´/oracle/<dbsid>/origlogA/log_g13m1.dbf´,
   ´/oracle/<dbsid>/mirrlogA/log_g13m2.dbf´) size 200M reuse;
alter database add logfile thread 1 group 14
  (´/oracle/<dbsid>/origlogB/log_g14m1.dbf´,
   ´/oracle/<dbsid>/mirrlogB/log_g14m2.dbf´) size 200M reuse;
alter database add logfile thread 2 group 21
  (´/oracle/<dbsid>/origlogA/log_g21m1.dbf´,
   ´/oracle/<dbsid>/mirrlogA/log_g21m2.dbf´) size 200M reuse;
alter database add logfile thread 2 group 22
  (´/oracle/<dbsid>/origlogB/log_g22m1.dbf´,
   ´/oracle/<dbsid>/mirrlogB/log_g22m2.dbf´) size 200M reuse;
alter database add logfile thread 2 group 23
  (´/oracle/<dbsid>/origlogA/log_g23m1.dbf´,
   ´/oracle/<dbsid>/mirrlogA/log_g23m2.dbf´) size 200M reuse;
```
alter database add logfile thread 2 group 24
  (`/oracle/<dbsid>/origlogB/log_g24m1.dbf`
  , `'/oracle/<dbsid>/mirrlogB/log_g24m2.dbf'`) size 200M reuse;
alter database add logfile thread 3 group 31
  (`/oracle/<dbsid>/origlogA/log_g31m1.dbf`
  , `'/oracle/<dbsid>/mirrlogA/log_g31m2.dbf'`) size 200M reuse;
alter database add logfile thread 3 group 32
  (`/oracle/<dbsid>/origlogB/log_g32m1.dbf`
  , `'/oracle/<dbsid>/mirrlogB/log_g32m2.dbf'`) size 200M reuse;
alter database add logfile thread 3 group 33
  (`/oracle/<dbsid>/origlogA/log_g33m1.dbf`
  , `'/oracle/<dbsid>/mirrlogA/log_g33m2.dbf'`) size 200M reuse;
alter database add logfile thread 3 group 34
  (`/oracle/<dbsid>/origlogA/log_g34m1.dbf`
  , `'/oracle/<dbsid>/mirrlogB/log_g34m2.dbf'`) size 200M reuse;
alter database add logfile thread 4 group 41
  (`/oracle/<dbsid>/origlogA/log_g41m1.dbf`
  , `'/oracle/<dbsid>/mirrlogA/log_g41m2.dbf'`) size 200M reuse;
alter database add logfile thread 4 group 42
  (`/oracle/<dbsid>/origlogB/log_g42m1.dbf`
  , `'/oracle/<dbsid>/mirrlogB/log_g42m2.dbf'`) size 200M reuse;
alter database add logfile thread 4 group 43
  (`/oracle/<dbsid>/origlogA/log_g43m1.dbf`
  , `'/oracle/<dbsid>/mirrlogA/log_g43m2.dbf'`) size 200M reuse;
alter database add logfile thread 4 group 44
  (`/oracle/<dbsid>/origlogB/log_g44m1.dbf`
  , `'/oracle/<dbsid>/mirrlogB/log_g44m2.dbf'`) size 200M reuse;

The new threads need to be enabled before they can be used.

```
SQL> alter database enable public thread 1;
SQL> alter database enable public thread 2;
SQL> alter database enable public thread 3;
SQL> alter database enable public thread 4;
```

After the files and groups have been created and enabled, the old set of redolog files can be dropped. Query v$log to see which is the current logfile group:

```
SQL> select group#, archived, status from v$log;
GROUP# ARC STATUS
------------ ---------------------
101 YES INACTIVE
102 NO  CURRENT
103 YES INACTIVE
104 YES INACTIVE
11 NO  UNUSED
12 NO  UNUSED
13 NO  UNUSED
... ...
44 NO  UNUSED
```

Do as many logfile switches as needed so that the current log is in the newly created group. Query v$log again if in doubt.

```
SQL> alter system switch logfile;
SQL> alter system switch logfile;
SQL> alter system switch logfile;
SQL> alter system switch logfile;
```
After this is accomplished, the old logfile groups can be dropped safely:

```
SQL> alter database drop logfile group 101;
SQL> alter database drop logfile group 102;
SQL> alter database drop logfile group 103;
SQL> alter database drop logfile group 104;
```

Don’t forget to delete the logfiles of group 101-104 at OS-level!

In the given example, it is assumed that the logfiles are not mirrored by hardware.

SAP recommends that the members of a logfile group reside on different disks for performance reasons. The preceding figure shows the optimal configuration where every set of redo logs resides on a separate disk or disk volume.
3.5 SAP User Profiles

In a SAP R/3 system, two UNIX users are defined. The environment of the user <sapsid>adm remains unchanged for a RAC configuration. The changes to the new ORACLE_HOME, NLS settings, and so forth apply to this user also, the same way as they do in a standard upgrade scenario to a newer Oracle software release. The user <sapsid>adm is the designated user to run the R/3 system. For this user account, it is not required to recognize the different database instances. This user needs only a view to the database in a transparent glance without knowing a specific instance.

All database related tasks are performed by the UNIX user ora<dbsid>. This user has to be aware of the different database instances and, therefore, some changes to the environment are required (compared to the environment of a single instance implementation of this user).

.sapenv.csh, .sapenv.sh

The user environment for the SAP user <sid>adm requires no change for RAC. All modifications made here reflect the upgrade of the database to Oracle9i Release 9.2.

Adjustments are for the variables ORACLE_HOME, ORA_NLS and so on.

.dbenv.sh, .dbenv.csh

The designated database administrator for a SAP R/3 system, ora<dbsid>, needs special treatment of the environment variables ORACLE_SID and THREAD, depending on the cluster node. The value of ORACLE_SID determines the init<dbsid><thread>.ora profile for database instance startup on the specified node. It has to be unique on every node in the cluster, and, therefore, the .dbenv.sh or .dbenv.csh files in the home directory for the user ora<dbsid> should set with these environment variables depending on the cluster node. As the environment setup from SAP for the user ora<dbsid> already distinguishes the filenames depending on the hostname, the changes for a node-specific setting are easy to establish.

The following excerpt from the .cshrc file for the C-Shell shows how it works:
Using this for the database environment, there must be a unique .dbenv_<hostname>.csh file in the home directory of the user ora<dbsid>. Within this file, only the node-specific THREAD number for the instance on that node is specified. The rest of the environment setup stays in a common file .dbenv.csh, which is included.

Example for csh:

File .dbenv_<nodename>.csh

```
# Oracle RDBMS Environment
setenv THREAD 001
source .dbenv.csh
```

The common file .dbenv.csh contains all the rest as follows:
Note that with this setup the home directory for the user ora<dbsid> can reside on a cluster file system also.

3.6 Server Control (SRVCTL) for Cluster Instance Administration

Oracle Server Management uses the SRVCTL utility (installed on each node) to manage configuration information that is used by some Oracle tools. For example, SRVCTL serves as a single point of control between the Oracle Intelligent Agent and the nodes. To use SRVCTL, the configuration information for the database must exist. This is accomplished by the following command which should be executed on one node as user ora<dsid>:

```
    srvctl add db -d <dbsid> -o <oracle_home_directory>
```

This command adds the database to the configuration repository. After that the instances are added to the configuration by the following command:

```
    srvctl add instance -d <dbsid> -i <instance_name> -n <node_name>
```

Repeat this command for every database instance and node.

Some environment variables for proper database / instance startup and shutdown are required also. The timezone setting of user ora<dsid> must be set in the configuration file to synchronize the system time information of the database server and the SAP R/3 application servers:

```
    srvctl setenv -d <dbsid> -t TZ=<timezone_for_R/3>
```
Part IV Upgrade SAP Software and Tools

4.1 BR*Tools Configuration

4.1.1 New Version of BR*Tools

BR*Tools is the name for the suite of programs and utilities from SAP to administer the Oracle database, perform backup and recovery, as well as many other reorganization tasks. These tools operate independent from the R/3 system. They do not rely on nor require any library or executable from the R/3 system itself. For this reason it is possible to use the BR*Tools of a newer SAP R/3 version together with the 4.6D executables of the R/3 system. Starting with SAP release 6.20, these tools are all built by SAP using Oracle9i and later. For an Oracle RAC configuration, replace these tools with the latest version shipped by SAP. The latest version is available on SAP’s service marketplace or through OCS.

As with all the other executables for a SAP installation, the executables of the BR*Tools toolset are located in the directory /usr/sap/<SID>/exe/run. BR*Tools consist of the following set of files:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sapdba</td>
<td>(no longer supported with WEB AS 6.40)</td>
</tr>
<tr>
<td>Brspace</td>
<td>(replacement for sapdba, also with GUI)</td>
</tr>
<tr>
<td>brconnect</td>
<td></td>
</tr>
<tr>
<td>brrestore</td>
<td></td>
</tr>
<tr>
<td>brbackup</td>
<td></td>
</tr>
<tr>
<td>brrestore</td>
<td></td>
</tr>
<tr>
<td>brarchive</td>
<td></td>
</tr>
<tr>
<td>brrecover</td>
<td></td>
</tr>
</tbody>
</table>

After the replacement of these files with the most current version available, the file permissions and ownership of the files should be checked and corrected as needed.

*SAP recommends use of the newest version of BR*Tools from the SAP Web Application Server Release 6.40.*

This version (Web AS 6.40) has built-in capabilities to administer Oracle RAC instances. It provides also comprehensive services to leverage almost all of the new features of Oracle 9i.

4.1.1 Configuration files

The configuration files for the BR*Tools are located in the $ORACLE_HOME/dbs directory. In a RAC environment with SAP R/3, these tools should only be used by user ora<dbsid>. This is the standard recommendation; however, with Real Application Clusters, this is not only recommended, it is necessary. The first thing to do is to make instance-specific copies of the original configuration files. The technique used here to establish an instance-specific configuration file for the BR*Tools is the same as already used to set up the Oracle database initialization files. By copying the file init<dbsid>.sap from the originating SAP R/3 system to init<dbsid><thread>.sap for as many instances as required, the user ora<dbsid> recognizes the correct config file for the cluster node by the naming schema.

For every newly created configuration file init<dbsid><thread>.sap, the corresponding entries in that file for the parameter parallel_instances must be set. In the following example the <dbsid> is RAC. There are four parallel instances in the
cluster. The configuration file is for the first member in the cluster. The name of the file is then initRAC001.sap.

```
# Oracle instance string to the primary database
# [primary_db = <inst_str> | LOCAL]
# no default
# primary_db = <inst_str>

# description of parallel instances for Oracle Parallel Server
# parallel_instances = <instance_desc> | (<instance_desc_list>)
# <instance_desc_list> -> <instance_desc>[,<instance_desc>...]
# <instance_desc>      -> <Oracle_sid>:<Oracle_home>@<inst_str>
# <Oracle_sid>         -> Oracle system id for parallel instance
# <Oracle_home>        -> Oracle home for parallel instance
# <inst_str>           -> Oracle instance string to parallel instance
# Do not include the local instance in the parameter definition!
# default: no parallel instances

# example for initC11.sap:
# parallel_instances = (C11_002:/oracle/C11@C11_002,
#                       C11_003:/oracle/C11@C11_003)
#                       
# example for initC11_002.sap:
# parallel_instances = (C11:/oracle/C11@C11,
#                       C11_003:/oracle/C11@C11_003)
parallel_instances = (RAC002:/oracle/RAC/901_64@RAC002,
                     RAC003:/oracle/RAC/901_64@RAC003,
                     RAC004:/oracle/RAC/901_64@RAC004)
#
# EOF
```

To set up the configuration correctly for every file, locate the parameter entry parallel_instances. Add a line for all the parallel instances configured in the cluster, but do not include the local instance for the local node!

*For detailed configuration information on BR*Tools in a RAC environment see SAP note 711665.*
4.2 SAP R/3: Changes and Modifications

4.2.1 Truncate table ddlog, Recreate DDLOG_SEQ sequence

SAP R/3 uses an Oracle database sequence number to synchronize the buffers between R/3 application servers. The name of this sequence is `ddlog_seq`. The synchronisation relies on the assumption that a higher sequence number is created by an event which happened later in time than any event that created a sequence number before and has a lower sequence number associated therefore. To guarantee this behaviour, the `ddlog_seq` has to be recreated with an ORDER option. The sequence is created with the NOORDER option by default. The table `ddlog` must also be truncated.

**CAUTION: All SAP R/3 application servers have to be stopped before!**

Log on to the database as user sapr3 by sqlplus:

```
SQL> connect sapr3/<password>
SQL> truncate table ddlog;
SQL> drop sequence ddlog_seq;
SQL> create sequence ddlog_seq minvalue -2147483640 maxvalue 2147483640 increment by 1 cache 50 order nocycle;
```

After that, SAP R/3 can be restarted. See also SAP note 743555.

4.2.2 New report RSORASNP

SAP has modified the report RSORASNP for use with Oracle RAC. This report is part of the performance collector of R/3 to gather database snapshot data. Download the SAP R/3 support-package from the SAP service marketplace. Depending on the actual SAP R/3 release, following support packages are available:

<table>
<thead>
<tr>
<th>Software-Component</th>
<th>Release</th>
<th>Packagename</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP_BASIS</td>
<td>46C</td>
<td>SAPKB46C48</td>
</tr>
<tr>
<td>SAP_BASIS</td>
<td>46D</td>
<td>SAPKB46D36</td>
</tr>
<tr>
<td>SAP_BASIS</td>
<td>610</td>
<td>SAPKB61030</td>
</tr>
<tr>
<td>SAP_BASIS</td>
<td>620</td>
<td>SAPKB62039</td>
</tr>
<tr>
<td>SAP_BASIS</td>
<td>640</td>
<td>SAPKB64002</td>
</tr>
</tbody>
</table>

See also SAP note 717442.
4.2.3 Change of view SAP_V$LOCK_GLOBAL and V$LOCK

From earlier OPS implementations, SAP created and used a UNION JOIN via database links to query the data for the views SAP_V$LOCK_GLOBAL and V$LOCK. The view definition is recreated every time just before the views are used, taking the actual number of active instances into account. This is no longer necessary with Oracle 9i RAC, as the GV$ - view provide a faster and easier way to gather the data across all instances without reconstructing a UNION JOIN condition for active instances. The views are used by the SAP R/3 reports RSORACMP, RSORAOPS, RSORAMAV and RSORASC2. These reports can be changed, dropping the recreation of the views and use a view definition valid for an arbitrary number of instances, as given below:

```sql
SQL> create or replace view "SAPR3"."SAP_V$LOCK_GLOBAL"
    2 ("HOSTNAME","PID","S_ADDR","S_KADDR","ID1","ID2","SID",
    3 "ADDR","KADDR","TYPE","REQUEST","TIME_WAITED")
    4 as
    5 (select A.MACHINE, A.PROCESS, A.TADDR, A.LOCKWAIT, C.ID1,
    6 C.ID2, C.SID, C.ADDR, C.KADDR, C.TYPE, C.REQUEST, E.TIME_WAITED / 100
    7 from SYS.GV$SESSION A, SYS.GV$LOCK C,
    8 SYS.V$SESSION_WAIT W, SYS.V$SESSION_EVENT E
    9 where
    10 C.TYPE IN ('TM','TX') and
    11 A.SID (+)= C.SID and
    12 W.SID = A.SID and
    13 E.EVENT = W.EVENT and
    14 A.INST_ID (+)= C.INST_ID);

SQL> create or replace view "SAPR3"."V$LOCK"
    2 ("ADDR","KADDR","SID","TYPE","ID1","ID2",
    3 "LMODE","REQUEST","CTIME","BLOCK")
    4 as
    5 (select
    6 "ADDR","KADDR","SID","TYPE","ID1","ID2",
    7 "LMODE","REQUEST","CTIME","BLOCK"
    8 from GV$LOCK;

The user SAPR3 must have permission to select the columns of the underlying base tables or views, otherwise there will be an error on querying the view from SAP R/3. Grant the SELECT permission on the base tables or views to user SAPR3.

Note: This is a performance optimization, it is not a strict requirement for running Oracle 9i RAC with SAP R/3. The solution using UNION JOIN is still valid.
4.2.4 Parameter MAX_COMMIT_PROPAGATION_DELAY

This parameter and its recommended value has been mentioned already in chapter 3.1.1 Modifying the Oracle Initialization Parameters for RAC, but because of the repeatedly arising questions about the effect on different settings the explanation following here seems to be appropriate.

To guarantee read consistency, all changes to the database must be recognized by all instances in the cluster. Oracle 9i RAC uses the system change number (SCN) to accomplish this: The SCN is propagated to all instances in the cluster. With Oracle 9.2 RAC there are two algorithm implemented:

**Broadcast-On-Commit**

The highest local SCN is send via broadcast to all instances immediately during a commit. With this implementation, read consistency is ensured for every kind of workload. If max_commit_propagation_delay < 100, broadcast-on-commit will be used.

**Lamport algorithm**

The local SCN is not send immediately with every change aka commit to the database. There is a possible delay up to 3 seconds until the changes are visible to all instances. The Lamport algorithm is used if max_commit_propagation_delay > 99.

The default and maximum value of max_commit_propagation_delay is 700, meaning up to 7 seconds possible delay for SCN propagation.

*In a SAP R/3 installation, broadcast-on-commit has to be used. The parameter max_commit_propagation_delay must be set to 1 on all platforms.*

On some platforms, a value of 0 for this parameter uses a platform specific implementation for the SCN propagation and can result in a negative performance impact.

Whether the right algorithm is used can be checked in the alertlog – file of the instance. See example below:

![Example Log Entries](https://example.com/example.png)

See SAP note 794361 also.
4.2.5 Workload distribution of SAP R/3 application servers to database instances

In chapter 3.2 Network Configuration, all the modifications and changes to the Oracle network configuration is explained. The tnsnames.ora configuration files given as an example there show how the clients on the cluster nodes connect to the database instance on that node and how the failover is configured. As mentioned in the Preface, all configuration examples refered to a 2-tier system. In a larger 3-tier configuration, SAP R/3 application servers run on an arbitrary number of additional computing nodes which are not a part of the cluster.

For the deployment of pure SAP R/3 application servers with one ore more SAP R/3 instances together with Oracle 9i RAC the following recommendations for configuration and operation should be followed.

To establish a good workload distribution to the available database instances, all SAP R/3 instances must be distributed in server groups. The number of server groups should match the number of configured database instances in the cluster. A server group must be connected in total with all SAP R/3 instances (and all the workprocesses of the instances) to the same Oracle RAC database instance. A distribution of the R/3 workprocesses to different database instances is not allowed and may lead to errors during the processing in the update task of R/3.

Distribution of SAP R/3 instances and application servers to multiple server groups is achieved by use of SAP transaction SM14. It is very important for the system configuration that the update task is performed locally within the server group and therefore a sufficient number of workprocesses for the update task (type vb and vb2) are provided.

It is possible to have a further distribution in pure dialog and update instances within a server group, but with respect to the high availability capabilities it’s a good advise to have a local update task within a SAP R/3 instance.

In case of an unplanned outage of one of the available database instances, all active workprocesses from one SAP R/3 server group should reconnect to one of the remaining database instances in a predefined way. The failover of a database session is performed automatically without any user intervention either by Oracle TAF or by an automatic reconnect of the SAP R/3 workprocess.

The predefined order for the failover or reconnect is determind by the Oracle network configuration file. It has to be assured that all members of a SAP R/3 server group use the same Oracle network configuration in the file tnsnames.ora. A new distribution of SAP R/3 application servers to SAP R/3 server groups has to be considered in the network configuration as well.

For a sheduled outage of a database instance, all SAP R/3 instances connected to this database instance should be stopped in advance. After the stop of the database instance the SAP R/3 instances can be restarted.

A restart of a database instance requires that all SAP R/3 instances, which are configured to use this database instance as the preferred instance, have to be stopped before the database instance is started. After the database instance has successfully joined the other members, the SAP R/3 instances can be started again. Only by this
procedure it is made sure that all workprocesses of one server group are connected properly to the right database instance.
Summary

Many of the activities described in this document for enabling SAP R/3 4.6 to use Oracle9i Real Application Clusters are more or less the same as for a simple upgrade of the database to Oracle9i Release 9.2. Of course, in the case of a single-instance Oracle database there is no cluster hardware configuration. You can think of this cluster setup as switching the database server hardware (and probably operating system software as well) for the new Oracle release. In this case the effort for installing the software and upgrading the database is nearly the same with or without the cluster solution. The real effort in enabling SAP R/3 4.6 is replacing some files from SAP (BR*Tools) and modifying the environment for the user ora<dbsid> to see the different instances on the cluster nodes. For the R/3 user <sid>adm and for the R/3 system itself all changes to the database configuration are mostly not visible at all. Additionally, the changes to the database (logfile, rollback segments, etc.) are rather straightforward to achieve by a few SQL statements. The most critical issue in managing a RAC enabled R/3 system is the correct setup of the network configuration. If this configuration is performed properly for all SAP clients to the database from within and outside the cluster, then running a clustered solution for SAP R/3 is easy to do.
Appendix A

Example Schedule for a GoingLive Migration

Overview

With this part of the document we want to provide a “Best Practices” schedule for a migration project of an existing SAP R/3 4.6 system towards a RAC system. The focus on this template is on minimizing the required downtime of the production system as well as lowering the risk of unexpected obstacles.

The simplified idea for this schedule is the use of a standby database of your production or test system. This standby database will be the target for the RAC enabling and will become the main or original database after the migration.

The assumption is of course that for the RAC cluster new hardware and storage subsystems will be used. In most cases this assumption meets the actual situation in a customer environment as the currently used hardware for the database cannot be easily upgraded to a cluster and most often the database files needs to be transported to a cluster file system also.

Please keep in mind that this schedule covers only homogenous database migration. This especially requires that the original hardware used for the database system and the hardware for the cluster use the same operating system and processor architecture.

If there is a switch on the underlying operating system and/or the processor architecture for the RAC cluster, it is strongly recommended that you perform a heterogeneous system migration in advance. This should be considered as a separate subproject. After the migration to the new operating system and/or processors plan for a fair amount of time to obey the new system for any critical issues. Let all the systems management operations stabilize. Tools and services for performing a heterogeneous system copy are available from SAP and Oracle Support.

There are two main advantages by proceeding according to the shown schedule:

1. Production systems are critical to customers in terms of downtime and failure. If, for any reason, something serious goes wrong and the cause for the failure cannot be corrected in a short time, just bring up the original system again. There is no loss of data and you can start over again.

2. The critical phase of the migration can be prepared in detail and therefore the actual downtime of the SAP system is rather low compared to all other activities.

And it is proven. The first migration of a productive SAP R/3 system was done according to this plan. The whole migration to a RAC enabled and fully configured SAP R/3 system was done in less than five hours including an upgrade from Oracle release 8.1.7 to 9.2.0.4.
1. Setup the new hardware

Install all necessary software components on the new hardware for the cluster. Connect the storage subsystems to the cluster. Setup the Interconnect between the nodes.

Refer to Part I of this document for a brief discussion on RAC specific issues for the setup of your platform of choice. This part shows somewhat minimum conditions the assembly for RAC requires. Please make sure to meet all prerequisites. As this guide is not intended to cover all platform specific tasks or configuration options, follow the instructions from the vendor for the complete setup. Also check the latest notes from SAP on this and make sure that the latest information available from Oracle Metalink is at hand.

Complete the layout of the storage. The cluster filesystems should be configured to imprint the layout of the original installation with respect to the access path for all datafiles and configuration files. Reserve additional space for multiple UNDO Tablespaces, Redo – Log Files, and new tablespaces for automatic segment space management (ASSM).

2. Install existing Oracle / SAP Software on the cluster

Do not copy the software from your production environment. It is strongly advised to use the original distribution and installation procedures from SAP and Oracle. Only this guaranties that all dependencies are fulfilled, and all entry to system files are made correctly.

Install the Oracle database software and the SAP R/3 software on one of the cluster nodes. Both, Oracle and SAP software, should reside on the cluster filesystem. Make sure that all nodes in the cluster have access to the installed files.

Check the Installation Guides from SAP for the installation procedure. Do a complete installation: Not only the software, but also the database load image from the distribution set. This database is for test purposes and will be deleted later on.

Choose the same user / group for user ora<SID> and <SID>adm as on the original system.

Check that the initial setup is correct by logging on to the SAP system on the cluster node.

3. Install Oracle Software Release 9.2 on the cluster nodes

After completing the initial setup, install the Oracle database software for release 9.2

Please refer to “Part II Upgrade the Database to Release 9.2” of this document on how to install the software on the cluster nodes and adjust the environment of the database user ora<sid>

Create a new home directory for user ora<sid>. Copy the content from old home directory to the new one. Change ORACLE_HOME to /oracle/<SID>/920_64.
Install the Oracle database software by running OUI. Install the latest Patchset 9.2.0.4 by running OUI. Install any additional patches using opatch.

4. Prepare Database Initialization Files

Copy the file $ORACLE_HOME/dbs/init<sid>.ora from the original system to the cluster.

Eliminate all deprecated parameters as described in chapter “2.2.1 Preparing the upgrade: Database Initialization file init<SID>.ora”.

This file will be used for the upgrade of the database to Oracle release 9.2 in a single instance environment.

After the corrections have been made, copy the file init<sid>.ora to a file init<sid>RAC.ora. Modify this file for use with the cluster instances. By doing this now, you just have to switch to this file for cluster database initialization during the hot phase of the migration. Follow the hints given in “PART III Adjusting the Environment” in this document.

5. Prepare the Network Configuration for the cluster

Adopt the changes to the files listener.ora and tnsnames.ora as described in chapter “3.2 Network Configuration” of this guide.

6. Prepare scripts for the upgrade and migration to the cluster

Create a directory “scripts” in $ORACLE_HOME/dbs. This directory holds some useful scripts for the upgrade and migration to the cluster.

1. Extending the SYSTEM Tablespace

See chapter 2.2.3 for additional explanations!

Example: Script for adding an additional file: add_system_datafile.sql

```sql
alter tablespace system add datafile '<full-path-of-datafile>'
size 80M autoextend on next 10M maxsize unlimited;
```

2. Use of UNDO Tablespaces for the RAC database instances

See chapter 3.3 Database UNDO Management for additional explanations!

Example: Script for adding UNDO Tablespaces: add_undo_tblspace.sql

Use the example given in chapter 3.3 as template for this script.

3. Additional REDO logfiles for the database instances

Example: Script for adding Logfile groups: add_logfile_groups.sql
Use the example given in chapter 3.4 “Redo Log Groups and Members” as template for this script.

7. Create standby database on the cluster

Change environment for ora<SID> to 817

# vi /etc/passwd
orap10:015byJE8.VJ.:3200:201:SAP Database Administration
User:/oracle/<SID>/806_64:/usr/bin/csh

# su – ora<SID>
# env
  ORACLE_SID=<sid>
  ORACLE_HOME=/oracle/<SID>/817_64
# exit
# vi /etc/oratab
  <SID>:/oracle/<SID>/817_64:N

If you tested the installation of the original software from steps 2 and 3, stop all SAP and Oracle processes:

  stopsap
  /usr/sap/<SID>/SYS/exe/run/saposcol –k
  su – ora<SID>
  lsnrctl stop
  svrmgrl –l
    shutdown

Delete any datafiles from previous testing:

  # cd /oracle/<SID>
  # rm –R sapdata*
  # rm –R origlog*
  # rm –R mirrlog*
  # rm –R saplog*
  # rm –R sapbackup
  # rm –R saparch
  # rm –R sapcheck
  # rm –R sapreorg
  # rm –R saptrace

Restore DB from last Full-Offline-Backup of the Source System

  # su – ora<SID>
  # cd /oracle/<SID>/sapbackup
  # ftp old_database_server
       user:<sid>, pw:…
  cd /oracle/<SID>/sapbackup
  dir *.aff
  get <timestamp>.aff
  by
  # view <timestamp.aff>

Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters  87
Oldest online log sequence = ?
-> BRBACKUP terminated successfully?
# brrestore –m all –r init<SID>.utl.clone –b <timestamp.aff>
-> BRRESTORE terminated successfully?

# cd /oracle/<SID>/saparch
# ftp old_database_server
   user:orap10, pw:…
   cd /oracle/<SID>/saparch
   get arch<SID>.log
by
# brrestore –a <oldest-log>-<newest-log> -r init<SID>.utl.clone
   (<newest-log> from arch<SID>.log of source system)

Create Standby database and start recovery

Create and copy standby controlfile(s)

Source system:

# su – ora<SID>
# svrmgrl
   connect internal
   alter database create standby controlfile as ‘/tmp/cntrl<SID>.dbf’;
exit
# exit

Standby system:

# su – ora<SID>
# cd /oracle/<SID>/sapdata1
# mkdir cntrl
# cd cntrl
# ftp old_database_host
   user:ora<SID>, pw:…
   cd /tmp
   get cntrl<SID>.log
by
# mkdir /oracle/<SID>/origlogA/cntrl
# mkdir /oracle/<SID>/saparch/cntrl
# cp –p cntrl<SID>.dbf /oracle/<SID>/origlogA/cntrl
# cp –p cntrl<SID>.dbf /oracle/<SID>/saparch/cntrl

Mount Standby Database and perform Recovery

# svrmgrl
   connect internal
   startup nomount;
   alter database mount standby database;
   recover standby database;
      auto
exit
After completing all the steps above, the preparation phase finished. The following steps describe the tasks to perform during the critical phase when the original database server will be replaced by the new cluster.

8. Shutdown productive system and activate standby database

1. **Stop all SAP and Oracle processes on PROD**

   # su – <SID>adm
   # stopsap
   # exit
   #/usr/sap/<SID>/SYS/exe/run/saposcol –k

2. **Archive current Redologs**

   # su – ora<SID>
   # svrmgrl
   connect internal
   alter system archive log current;
   exit

3. **Transfer all ArchiveLogs to Standby System and perform Recovery, Standby System:**

   # su – ora<SID>
   # cd /oracle/<SID>/saparch
   # ftp old_database_host
   user:ora<SID>, pw:…
   cd /oracle/<SID>/saparch
   bin
   prompt
   mget *.dbf
   by

   # svrmgrl
   connect internal
   recover standby database;

4. **Activate Standby Database**

   (in svrmgrl)
   alter database activate standby database;
   shutdown
   startup

9. **Migrate to Oracle9i**

   1. Extend system tablespace

      # su – ora<sid>
      # svrmgrl
      connect internal
      @?/dbs/scripts/add_system_datafile.sql
2. Check for invalid objects

```
# su – ora<sid>
# svrmgrl
   connect internal
   set echo on
   spool /oracle/P10/logs/Chk4InvObj.log
   select count(*) from dba_objects where status='INVALID';
   select owner, object_name, object_type
   from dba_objects where status='INVALID';
   spool off;
```

2. Execute script PREMIG.SQL (delivered by SAP)

```
# su – ora<SID>
# svrmgrl
   connect internal
   @?/dbs/scripts/PREMIG.SQL
```

3. Stop database

```
# su – ora<SID>
# svrmgrl
   connect internal
   shutdown
```

4. Change environment of ora<SID> from 817 to 920

```
# vi /etc/passwd
ora<sid>:0l5bbyJE8.VJ.:3200:201:SAP Database Administration
User:/oracle/<SID)/920_64:/usr/bin/csh

# vi /oracle/<SID)/920_64/.dbenv.csh
#setenv THREAD threadnumber_of_this_node
#if ( $THREAD == NOPS ) then
    # set DBSID = <SID>
#else
    # set DBSID =<SID>${THREAD}
#endif

set DBSID = <SID>

# su – ora<SID>

# env
ORACLE_SID=<SID>
ORACLE_HOME=/oracle/<SID)/920_64
LD_LIBRARY_PATH=/usr/sap/>SID>/SYS/exe/run:/oracle/<SID)/920_64/lib
# exit
```

5. Adjust init<SID>.ora compatible parameter

Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters  90
10. Database Structure Upgrade to Oracle 9.2

```
# su – ora<SID>
# cd /oracle/<SID>/920_64/rdbms/admin
# sqlplus /nolog
connect / as sysdba
spool /oracle/<SID>/logs/upgrade.log
set echo on
startup migrate
@u080107.sql
spool off

select comp_id, version, status from dba_registry;
<table>
<thead>
<tr>
<th>COMP_ID</th>
<th>VERSION</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG</td>
<td>9.2.0.3.0</td>
<td>VALID</td>
</tr>
<tr>
<td>CATPROC</td>
<td>9.2.0.3.0</td>
<td>VALID</td>
</tr>
</tbody>
</table>

spool /oracle/<SID>/logs/cmpdbmig.log
@cmpdbmig.sql
spool off

dbms_stats.delete_schema_stats('SYS');

spool /oracle/<SID>/logs/catpatch.log
@catpatch.sql
spool off

dbms_stats.gather_schema_stats('SYS');

shutdown immediate
startup

spool /oracle/<SID>/logs/utlrp.log
@utlrp.sql
spool off

spool /oracle/<SID>/logs/Chk4InvObj.log2
select count(*) from dba_objects where status='INVALID';
spool off

shutdown immediate
exit
```

11. POST Migration Steps
1. Execute script POSTMIG.SQL

   # su – ora<SID>
   # cd /oracle/<SID>/scripts
   # sqlplus /nolog
   connect / as sysdba
   @POSTMIG.SQL

2. Start the Oracle Listener

   # lsnrctl start

3. Change oratab entry

   # vi /etc/oratab
   <SID>:/oracle/<SID>/920_64:N

4. Execute sapdba_role.sql

   # cd /oracle/<SID>/scripts
   # sqlplus /nolog
   connect / as sysdba
   startup
   exit
   # sqlplus /nolog @sapdba_role.sql <SID> UNIX

5. Update the Oracle Database Statistics

   # brconnect –c –u <user>/<pw> -f stats –t all

12. Migrate single instance to 9i RAC

1. Create new controlfile if necessary

   # su – ora<SID>
   # sqlplus /nolog
   connect / as sysdba
   alter database backup controlfile to trace
   shutdown
   exit

   # cd /oracle/<SID>/saptrace/usertrace
   # ls -ltr
   # cp -p <tracefile> /oracle/<SID>/scripts/create_controlfile.sql
   # cd /oracle/<SID>/scripts
   # vi create_controlfile.sql
   MAXLOGFILES 255
   MAXINSTANCES 50

   >>>delete all comments!!

   # sqlplus /nolog
   connect / as sysdba
2. Create additional UNDO tablespaces

```sql
# su – ora<SID>
# sqlplus /nolog
connect / as sysdba
@ ?/dbs/scripts/add_undo_tblspace.sql
exit
```

3. Create additional online redologs

```sql
# su – ora<SID>
# sqlplus /nolog
connect / as sysdba
@ ?/dbs/scripts/add_logfile_groups.sql;
```

```sql
select group#, archived, status from v$log;
```

<table>
<thead>
<tr>
<th>GROUP#</th>
<th>ARC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>CURRENT</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>CURRENT</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>CURRENT</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>UNUSED</td>
</tr>
</tbody>
</table>

```sql
alter system switch logfile;
alter system switch logfile;
alter system switch logfile;

```sql
select group#, archived, status from v$log;
```

<table>
<thead>
<tr>
<th>GROUP#</th>
<th>ARC</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>INACTIVE</td>
</tr>
</tbody>
</table>

```sql
alter database drop logfile group 1;
```
alter database drop logfile group 2;
alter database drop logfile group 3;
alter database drop logfile group 4;

select group#, archived, status from v$log;

GROUP# ARC STATUS
---------- --- ----------------
11 YES INACTIVE
12 YES INACTIVE
13 NO CURRENT
14 YES UNUSED
21 YES INACTIVE
22 NO CURRENT
23 YES UNUSED
24 YES UNUSED

delete logfiles of group 1-4 at OS-level

4. Create public database links

# su – ora<SID>
# sqlplus /nolog
connect / as sysdba
show parameter domain;
select * from global_name;
alter database rename global_name to <SID>.world;

create public database link '<SID>001' using "<SID>001";
create public database link '<SID>002' using "<SID>002";

select db_link, host from dba_db_links;
exit

5. Stop all Oracle processes

# su – ora<SID>
# sqlplus /nolog
connect / as sysdba
shutdown
exit

6. Change environment for user ora<SID>

# vi /oracle/<SID>/920_64/.dbenv.csh
# Entry for 9i RAC implementation

setenv THREAD threadnumber_of_this_node
if( $THREAD == NOPS ) then
    set DBSID = <SID>
else
    set DBSID = <SID>$\{THREAD\}
endif

#set DBSID = <SID>     >>> comment out!!
7. Create oracle password file

```
# su – ora<SID>
# cd /oracle/<SID>/920_64/dbs
# rm orapw (if exists)
# orapwd file=/oracle/<SID>/920_64/dbs/orapw password=xxx
entries=100

# sqlplus /nolog
connect / as sysdba
startup
grant sysoper, sysdba, connect, dba, resource to system;
```

8. Start instances on other nodes

```
# su – ora<SID>
# sqlplus /nolog
connect / as sysdba
startup
select * from gv$instance;
```

9. shutdown all instances

10. Cleanup

```
# cd /oracle/<SID>/920_64/dbs
# rm init<SID>.sap
# rm init<SID>.ora

# cd /oracle/<SID>/saptrace/background
# mv * OLD

# cd /oracle/<SID>/saparch
# mv arch<SID>.log OLD
```

13. Post-Migration Steps

1. Copy additional filesystems from source system.

2. Install SAP Licenses

3. Redefine R/3 Printer (optional): SAP transaction spad

4. Adjust Transport Management System: SAP transaction stms

5. Adjust SAP-Logon-Groups: SAP transaction smlg

6. Reactivate cancelled Jobs: SAP transaction sm37
Appendix B

Use of advanced Oracle Features

Oracle strongly recommends to use the following features for a RAC database. A general note from SAP (OSS note 598678) is available explaining the usefulness of these new features in Oracle 9i for all SAP installations. Especially for Oracle RAC together with SAP R/3 review SAP OSS note 527843. This note keeps the actual status of the ongoing RAC certification for all supported hardware platforms by SAP.

Automatic UNDO Management

UNDO tablespaces specific for every database instance in a RAC environment are superior compared to the traditional rollback segments in one tablespace only. Beside of performance benefits the administration of rollback segments is much easier using automatic UNDO. See SAP OSS note 600141 for detailed explanation to R/3 users.

Locally Managed Tablespaces LMTS

Locally managed tablespaces use a bitmap to manage the free space in a tablespace. LMTS is a prerequisite for using ASSM (see below). See SAP OSS note 214995 for a detailed explanation on LMTS for use with SAP R/3. This note discusses all recommendations on how to migrate existing tablespaces to LMTS, which restrictions apply in a SAP environment, and which tools from SAP already support the use of LMTS.

Locally Managed System Tablespace

For a new installation such a migration from a different database or operating system Oracle recommends using a locally managed system tablespace. The benefits of LMTS apply to the SYSTEM tablespace as well. Remember: If the SYSTEM tablespace is locally managed, all other tablespaces have to be locally managed also.

Automatic Segment Space Management ASSM

In a RAC environment the number of potential contentions decreases due to ASSM. Tuning of storage parameters PCTUSED/FREELISTS/FREELISTGROUPS is no longer necessary. ASSM scales by itself with the number of concurrent users as well as with the number of instances. See SAP OSS note 620803 for implementing ASSM in a R/3 system.

Dynamic SGA

Leveraging a dynamic SGA may result in some performance advantages in a RAC cluster. Main benefit is the dynamic resizing of caches within a database instance. Most important is the optimized use of the library cache, as reparsing of statements involves additional overhead to the GES and GCS resources that in turn increase the
communication traffic between instances. Administration is made easier as the different pools respond automatically to changing workloads in an instance.

**Automatic PGA**

A RAC database may benefit from automatic PGA as main memory on the cluster node is used more efficiently. In addition, the administration overhead for maintaining parameters `sort_area_size`, `hash_area_size`, `bitmap_merge_size` and `create_bitmap_area_size` is eliminated. On the other hand, determining the correct size for the new parameter `pga_aggregate_target` requires some monitoring till a best fitting value is explored. See SAP OSS note 619876 on using automatic SGA with SAP R/3 and how to determine correct sizes for instance parameters.

**Default Temporary Tablespace**

Main advantage is the avoidance of allocating temporary segments (e.g. for sorts) to the SYSTEM tablespace. In a SAP R/3 environment with MCOD, using a default temporary tablespace for all users makes the administration tasks easier to handle. See SAP OSS note 683075 for a detailed discussion of the benefits for SAP R/3 users.

**Server Parameter File (SPFILE)**

There is no performance gain by using a SPFILE `spfile.ora` instead of a PFILE `init.ora`. The advantage is in a better manageability of the cluster instances. It is a prerequisite for integrating a RAC database to the Oracle Enterprise Manager. See SAP OSS note 601157 on how to create and maintain a SPFILE. If you are already using the BR*SPACE toolset, you are able to do all maintenance tasks regarding a SPFILE and parameters contained with these tools.
Appendix C

Network and Interconnect

If it comes to Oracle 9i RAC together with SAP R/3, the FAQ’s are mostly about network issues and the interconnect of the cluster. Although this actually is important for the overall performance, it is not the only one. Nevertheless this part is intended to give some hints on the topic.

Fujitsu Primecluster

Primecluster inter-node communication uses the Internode Communication Facility (ICF) module. This module is part of the Cluster Foundation (CF) and is not available to user-level resources. Applications can access the cluster interconnects via the Cluster Interconnect Protocol (CIP). CIP implements a TCP/IP communication stack for user applications. Oracle’s IPC layer uses CIP for RAC.

Use the “cftool” utility to configure the private interconnect between the nodes.

Review the hostname-to-IP resolution in the cluster configuration file /etc/cip.cf

Use “ndd” to set the following parameters to the maximum value (OS maximum):

```
udp_xmit_hiwat
udp_recv_hiwat
```

HP Tru64

The interconnect is established by the memory channel between the nodes. The protocol used is RDG. The init.ora parameter CLUSTER_INTERCONNECTS is ignored when using RDG. For tuning efforts, it is best to use operating system facilities to set up the memory channel for maximum performance.

IBM AIX (HACMP)

Oracle RAC uses UDP IPC protocol for the interconnect. Oracle chooses up to three network interfaces for the connection. A network is selected by Oracle if it is configured as “service” by HACMP. “Private” network connections are preferred over “Public” network connections. If there is more than one, (“Private”, then “Public”), selection is made on performance: Switch > FDDI > Ethernet.

Even if there is more than one network selected, traffic will go only over one connection. The remaining are for high availability only, not for load balancing (TNFF). If you want to use load balancing, you have to configure this by HACMP: Create a service which combines more than one network interfaces. “Jumbo-Frames” are supported by Oracle RAC.

Alternatively you can set the CLUSTER_INTERCONNECTS parameter in the init.ora file, specifying all the interfaces you want to use. Note: This will disable TNFF. We do not recommend this.
Use "/usr/sbin/cluster/utilities/cllsif –S" to determine the networks configured for your system.

Use “no” to tune these network parameters:

```
udp_recevspace
udp_sendspace
```

Set both parameters to a high value or (better) set it to the maximum permitted value.

**LINUX**

On Linux, UDP IPC protocol is used for RAC. The network address selection depends on the parameter given for the “HostName” in the OCMS configuration file ($ORACLE_HOME/oracm/admin/cmcfg.ora).

Tuning can be done by editing these files:

```
/proc/sys/net/core/rmem_default
/proc/sys/net/core/rmem_max
/proc/sys/net/core/wmem_default
/proc/sys/net/core/wmem_max
```

**SUN Solaris**

On Sun Cluster there are two options for the interconnect. You can use the SCI-PCI interconnect, or Sun Firelink (a.k.a. Wildcat) hardware on Sun Fire servers.

Leveraging RSM requires Oracle patch 2642824.

The IP address is taken from the cluster configuration file for this node. Under normal circumstances there is no need to set the CLUSTER_INTERCONNECTS parameter in the init.ora file.

Use the utility “ndd” to set the following parameters to the OS maximum value:

```
udp_xmit_hiwat
udp_recv_hiwat
```
Veritas DBE/AC

All NIC’s which have been specified during cluster setup (installDBAC) are used in parallel. The low-latency transport daemon (lltd) from Veritas is used for internode communication in the cluster.
Tuning may be done by adopting the flow control values. Under normal circumstances this should not be necessary. Check the Veritas logfiles for warnings on communication issues.
Do not set LLT flow control values below the defaults in /etc/llttab. The current defaults are

\[
\begin{align*}
\text{lowwater} & = 1100 \\
\text{highwater} & = 1200 \\
\text{window} & = 400
\end{align*}
\]

HP-UX (UDP Protocol)

The UDP protocol is not tunable.
Appendix D

References and Notes

Oracle9i Real Application Clusters Administration
Oracle9i Real Application Clusters Concepts
Oracle9i Real Application Clusters Deployment and Performance
Oracle9i Real Application Clusters Setup and Configuration
Oracle9i Real Application Clusters Documentation Online Roadmap
Oracle9i Installation Guide Release 2 for UNIX Systems: AIX-Based Systems, Compaq
Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris
Oracle9i Real Application Clusters on HP-UX
Tips for Installing and Configuring Oracle9i Real Application Clusters on Red Hat Linux
Advanced Server
Oracle9i Real Application Clusters on Compaq Tru64
Oracle9i Release Notes for Sun Solaris (64-bit)

RAC/IBM AIX certification matrix
RAC/HP certification matrix
RAC/Linux certification matrix
RAC/Tru64 Certification Matrix
RAC/Sun certification matrix
RAC/Veritas on Sun certification matrix

Fujitsu PrimeCluster documentation including PrimeCluster 4.0 Concepts,
PrimeCluster 4.0 Installation Guide, and PrimeCluster 4.0 System Administration and
Configuration Guide

Oracle9i RAC Installation with a NetApp Filer in Fujitsu Siemens PrimeCluster
Environment.

Sun Cluster 3.0 Release Notes
Sun Cluster 3.0 Installation Guide, Part No. 806-1419-10
Sun Cluster 3.0 System Administration Guide, Part No. 806-1423-10
Sun Cluster 3.0 Hardware Guide, Part No. 806-1420-10
Sun Cluster 3.0 Concepts, Part No. 806-1424-1
Sun Cluster 3.0 Error Messages Manual, Part No. 806-1426-10
Sun Cluster 3.0 Data Services Installation and Configuration Guide, Part No. 806-1421-10

Veritas Database Edition™ 3.5 Advanced Cluster for Oracle9i RAC Release Notes
Maintenance Pack 1 & 2 for Solaris.
Veritas Database Edition™ 3.5 Advanced Cluster for Oracle9i RAC Installation and
Configuration Guide for Solaris.
Veritas Database Edition™ 3.5 Advanced Cluster for Oracle9i RAC Participant Guide,
Training Manual 100-001877, Veritas Education.
Veritas Datasheet: Veritas Database Edition™ 3.5 Advanced Cluster for Oracle9i RAC
Veritas White Paper: VERITAS Database Edition/Advanced Cluster

Note:201019.1 - AIX: Quick Start Guide - 9.2.0 RDBMS Installation
Note:77346.1 - Overview of HACMP Classic and / or HACMP/ES
Note:177087.1 - Metalink COMPAQ: Quick Start Guide - 9.0.x RDBMS Installation
Note:154360.1 - HP: Quick Start Guide - 9.0.x RDBMS Installation
Note:201370.1 - LINUX Quick Start Guide - 9.2.0 RDBMS Installation
Configuring SAP R/3 4.6D To Use With Oracle Real Application Clusters
Configuring SAP R/3 4.6D for Use with Oracle Real Application Clusters
March 2005
Author: Kurt Brög
Contributing Authors: Jan Klokkers, Cathryn Shea

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
www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various Product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2005 Oracle Corporation
All rights reserved.