Managing ServiceGuard Extension for SAP

Manufacturing Part Number: B7885-90018

June 2003

Legal Notices

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Corporate Offices:

Hewlett-Packard Development Co., L.P.
3000 Hanover St.
Palo Alto, CA 94304

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Use of this manual and flexible disc(s), compact disc(s), or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

EMC^2, Symmetrix, and SRDF are trademarks of EMC Corporation. Windows NT is a trademark of Microsoft Corporation. SAP and R/3 are trademarks of SAP AG.

Contents

Contents

Contents

Printing History

Table 1 **Editions and Releases**

Printing Date	Part Number	Edition	SGeSAP Release	Operating System Releases
June 2000	B7885-90004	Edition 1	B.03.02	HP-UX 10.20 HP-UX 11.00
March 2001	B7885-90009	Edition 2	B.03.03	HP-UX 10.20 HP-UX 11.00 HP-UX 11i
June 2001	B7885-90011	Edition 3	B.03.04	HP-UX 10.20 HP-UX 11.00 HP-UX 11i
March 2002	B7885-90013	Edition 4	B.03.06	HP-UX 11.00 HP-UX 11i
June 2003	B7885-90018	Edition 5	B.03.08	HP-UX 11i

The printing date and part number indicate the current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The part number changes when extensive technical changes are incorporated.

New editions of this manual will incorporate all material updated since the previous edition.

HP Printing Division:

Business Critical Computing Business Unit
Hewlett-Packard Development Co., L.P.
19111 Pruneridge Ave.
Cupertino, CA 95014

About this Manual...

This document describes how to configure and install highly available R/3 systems on HP-UX 11.x using ServiceGuard. It refers to the HP product B7885BA – MC/SG Extension for SAP. To get further information regarding SAP and High Availability in general, please refer to the SAP documents:

- *SAP WAS in Switchover Environments*
- *SAP BC High Availability*

To understand this document you have to have knowledge of the basic ServiceGuard concepts and commands. Experience in the Basis Components of SAP R/3 will also be helpful.

This manual consists of three chapters:

- Chapter 1, “Understanding ServiceGuard Extension for SAP,” describes ServiceGuard Extension for SAP (SGeSAP) concepts. It also explains the current approach and why it was taken. It can be used like a FAQ list. If you have technical questions about the installation, especially during the scoping phase, refer to this section to get the information you need.
- Chapter 2, “Step by Step Installation Guide,” describes the installation of SGeSAP step-by-step down to the HP-UX command level.
- Chapter 3, “SAP Supply Chain Management,” contains a summary of how to configure and setup SAP Supply Chain Management using a failover LiveCache.
- Chapter 4, “SGeSAP Administration,” contains a summary of how to switch off SGeSAP. This is useful for very complex WAS reconfiguration tasks like *putting* the WAS-System.

Table 2 lists abbreviations used in this manual:

Table 2 Abbreviations

Abbreviation	Meaning
<SID>, <sid>	System ID of the SAP WAS system in uppercase/lowercase

Table 2 Abbreviations (Continued)

Abbreviation	Meaning
<INSTNR>	instance number of the SAP WAS system
<pri_host>, <sec_host>, <local_host>	names mapped to the local hostnames (output of uname -n)
<primary>, <secondary>, <local>	names mapped to local IP addresses of the client LAN
<relocdb>, <relocci>, <relocdbci>	names mapped to relocatable IP addresses of MC/SG packages in the client LAN
<primary_s>,<secondary_s>, <local_s>	names mapped to local IP addresses of the server LAN
<relocdb_s>, <relocci_s>, <relocdbci_s>	names mapped to relocatable IP addresses of MC/ServiceGuard packages in the server LAN

Related Documentation

The following documents contain additional related information:

- *Managing Systems and Workgroups* (Part Number B2355-90157)
- Managing ServiceGuard (Part Number B3936-90026) for HP-UX 10.20
- *Managing MC/ServiceGuard* (Part Number B3936-90065) for HP-UX 11.0 and 11i
- *Managing Highly Available NFS* (Part Number B5125-90001)
- *ServiceGuard Extension for SAP Release Notes* (For your version of the product)

1 Understanding ServiceGuard Extension for SAP

HP ServiceGuard Extension for SAP (SGeSAP) extends ServiceGuard's powerful failover capabilities to SAP mySAP environments. It continuously monitors the health of each SAP Web Application Server (WAS) node and automatically responds to failures or threshold violations. It can also minimize planned downtime when performing SAP upgrades. ServiceGuard protects the SAP WAS central instance and database by defining them in ServiceGuard packages. Protected WAS components include the R/3 kernel and the [associated] J2EE engine.

This chapter provides an overview of how SGeSAP works. Topics are as follows:

- Overview of SGeSAP
- Application Servers
- SAP Application Server Packages
- Highly Available NFS
- Networking
- Directory Structures
- SAP WAS Local Executables
- ServiceGuard Package Concepts
- Server Consolidation
- SGeSAP Product Files
- Cluster Consistency Monitor for SGeSAP Environments
- Metropolitan Clusters and SGeSAP

Overview of SGeSAP

SAP WAS allows a great amount of flexibility in setup and configuration. The SGeSAP Extension Scripts preserve much of this flexibility through the use of two integration models:

- One Package Configuration Model
- Two Package Configuration Model

One-Package Configuration Model

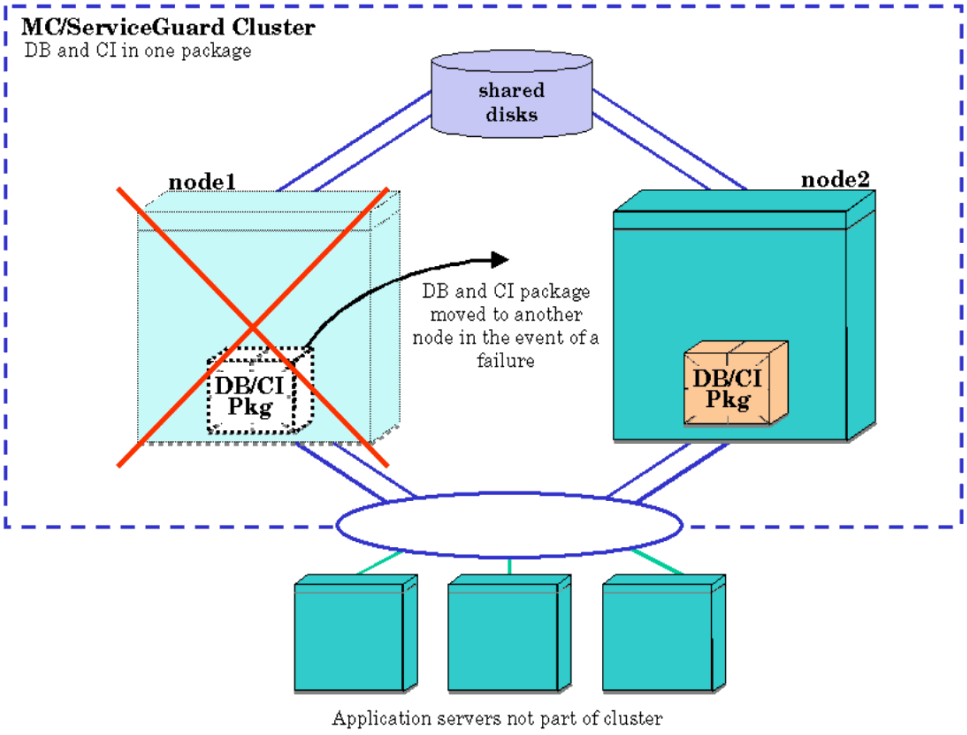
In a one-package configuration, both the database (DB) and central instance (CI) run on the same node at all times and are configured in one ServiceGuard package. If a J2EE Engine is part of the central instance, it is part of the package. Other nodes in the ServiceGuard cluster function as backups for the primary node (on which the system runs during normal operation).

If the primary node fails, the database and the central instance fail over and continue functioning on an adoptive node. The process of failover results in downtime that can last several minutes, depending on the work in progress when the failover takes place. The main portion of this downtime is needed for the recovery of the database.

After failover, the system runs without any manual intervention needed. The application servers are not part of the cluster but can either stay up or be restarted during failover.

A sample configuration in Figure 1-1 shows *node1* with a failure, which causes the package containing the database and central instance to fail over to *node2*.

Figure 1-1 One-Package Failover Scenario



Use the one-package model for all configurations where you can put the database and central instance on one node and you have available an equal sized node as a backup. During normal operation, the backup node can be used as an idle standby, application server, or test system.

Two-Package Configuration Model

If you are planning to distribute the database and central instance between two nodes, use the two-package model. The SAP WAS functionality is separated into two ServiceGuard packages, one for the

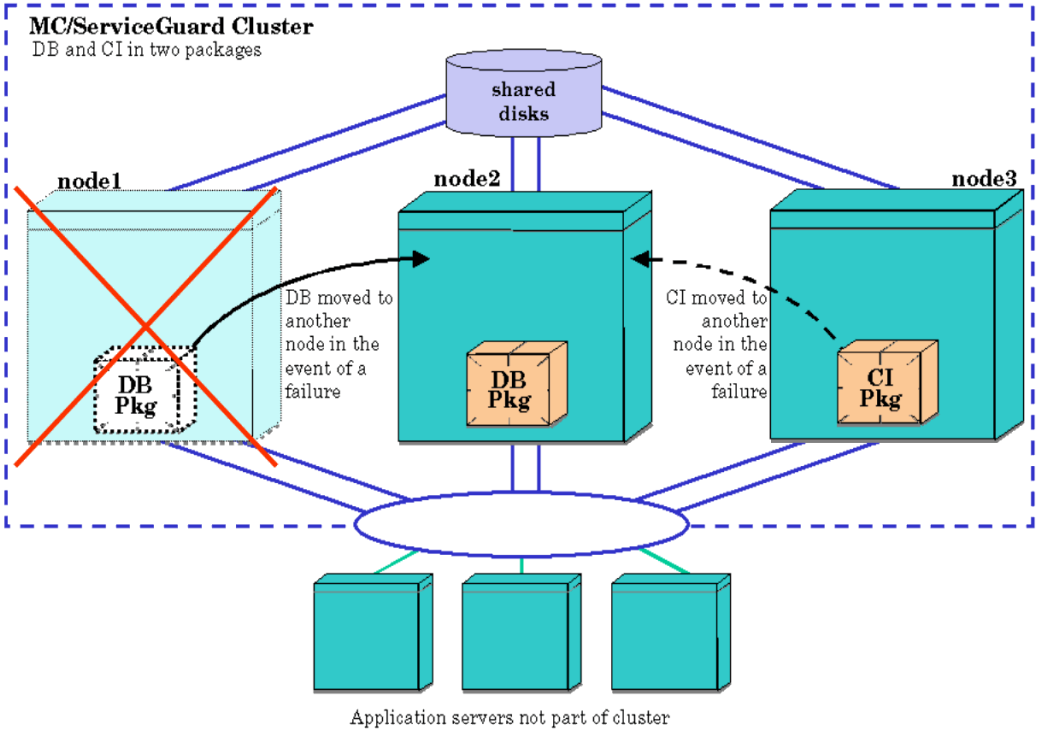
database (DB) and the other for the SAP WAS central instance (CI). If a J2EE Engine is part of the setup, it becomes part of the CI package. The database package contains the filesystems for the NFS mount points.

The cluster can be configured so that the two nodes back up each other, or so that one or more dedicated hosts back up the nodes running the SAP WAS packages as illustrated in Figure 1-2.

If either *node1* or *node2* fails, the package can fail over to *node3*. If a failover to a node that is not idle takes place (for example if *node3* were used as an application server), SGeSAP, if requested, can bring down running instances to free the resources needed to get the system up.

Under normal conditions, all backup hosts can be used to run application servers or instances of different test or development systems, or they can be idle. If needed, additional application servers inside and outside of the cluster can be restarted automatically. It is possible to define more than one highly available SAP WAS system in one cluster.

Figure 1-2 Two-Package Failover Scenario with Third Node as Standby



Dedicated NFS Package

In complex, consolidated environments it is an option to use a dedicates HA NFS package. This SAPNFS package is specialized to providing access to shared filesystems that are needed by more than one mySAP component. Typical filesystem served by SAPNFS would be the common SAP transport directory or the global SAPDB executable directory. The SAPDB client libraries are part of the global SAPDB executable directory and access to these files is needed by APO and Livecache at the same time.

SGeSAP setups are designed in a way that avoids HA NFS shared filesystems with heavy traffic if possible. For many implementations, this gives the option to use one SAPNFS package for all HA NFS needs in the SAP consolidation cluster without the risk to create a serious performance bottleneck. Usually, HA NFS is delivered in a distributed fahion with each database package serving its own filesystems. By consolidating this into one package, all NFS serving capabilities can be removed from the database packages.

Application Servers

The database and the SAP WAS central instance are always running on nodes that are protected by ServiceGuard.

Other SAP WAS services can be run on HP-UX application server hosts. These hosts do not need to be part of the ServiceGuard cluster. Even if the additional SAP WAS services are run on nodes in the ServiceGuard cluster, they are not protected by ServiceGuard. A combination of Windows NT/HP-UX application servers is technically possible but additional software is required to access HP-UX filesystems or HP-UX-like remote shells from the Windows NT system.

All application servers different from the central instance are called *additional application servers*. An additional application server that runs on a cluster node is called an *internal application server*. External application servers run on HP-UX- or Windows NT-hosts that are not part of the cluster. Even if application servers are external to the cluster, they are affected by package startup and shutdown.

Run the SAP WAS services dialog, update, batch, spool, and gateway, on additional application servers. Do not run the message or enqueue services on additional application servers. Set up one message server and one enqueue server for each SAP WAS system.

The message server, enqueue server, database and NFS server are all single points of failure (SPOF). To maintain high availability, all these SPOFs for the SAP WAS system must be eliminated by configuring them in ServiceGuard cluster nodes.

In standard SAP WAS scenarios the SPOFs database, NFS, message and enqueue server are all protected. It is highly recommended that you also protect at least one instance of each additional SAP WAS service in the ServiceGuard cluster.

For all application server nodes outside the ServiceGuard cluster, consider the following for each of the SAP WAS services:

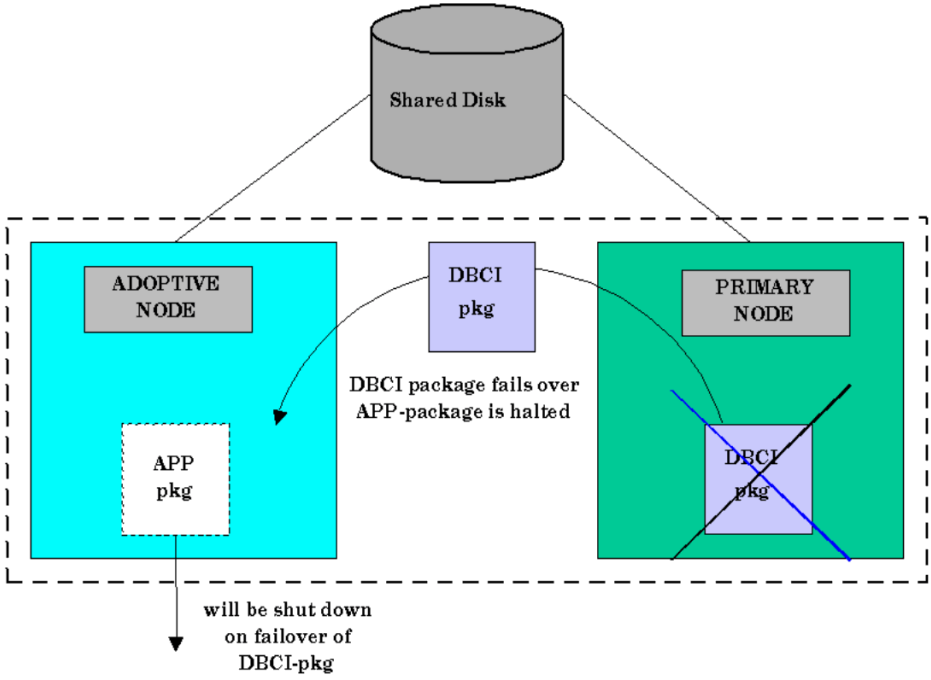
- `Dialog`—Logon using `saplogon` for an application server groups instead of `saptemu/sapgui` for each individual application server. When logging on to an application server group with two or more application servers, you not need a different login procedure even if one of the application servers of the group fails. Also, using login groups provides workload balancing between application servers.
- `Update`—Configure an update service on a node only for application services running on the same node. This ensures that the remaining SAP WAS servers, running on different nodes, are affected if an outage occurs on the update server. However, if the update server is configured to be responsible for application servers running on different nodes, any failure of the update server will lead to subsequent outages at these nodes. Configuring the update service on the high available central instance is recommended. Consider using local update servers only if performance issues require it.
- `Batch`—Do not specify a *destination host* when defining a batch job. This allows the batch scheduler to choose a batch server that is available at the start time of the batch job. If you must specify a destination host, specify the batch server running on the highly available central instance.
- `Spool`—Print requests stay in the system until the node is available again and the spool server has been restarted. These requests could be moved manually to other spool servers if one spool server is unavailable for a long period of time. An alternative is to print all time critical documents through the highly available spool server of the central instance.
- `Gateway`—You don't have to worry about the gateway processes `gwrd` and `gwwr` with SAP WAS 3.0 or higher. Do not confuse this with the SNA gateway.

SGeSAP monitors the hardware within the cluster (CPU, Network cards, etc), but it does not monitor the health of the SAP WAS processes or the database. This means that ServiceGuard will not switch the packages to another node if you have problems with SAP WAS or the database.

SAP Application Server Packages

With SGeSAP Version 3.03 it is possible to configure an SAP Application Server within a ServiceGuard package. For example, Figure 1-3 illustrates that in a 2 node cluster it is common to configure the adoptive node to run as an additional application server.

Figure 1-3 Adoptive Node as Application Server



Whenever a failover of the DBCI-package happens, this application server package will be shut down to free up resources for the DBCI-package. When the failed primary node is afterwards repaired the application server package can be easily brought up on this repaired node. The advantage is that the DBCI package can remain running on the former adoptive node without additional downtime for the application.

An additional example for the need of this SAP Application Server package is to protect dedicated batch servers against hardware failures.

Refer to Chapter 2, page 79 to get detailed information on how to configure Application Server Packages.

Highly Available NFS

The SAP WAS system requires the Network File System (NFS) to be highly available. Otherwise the NFS is one of the four single points of failure (SPOF) in the SAP WAS environment. SGeSAP works with HP's Highly Available NFS product, the NFS Toolkit (B5140BA). Refer to the *ServiceGuard NFS Toolkit Versions A.11.00.04 and A.11.11.01 Release Notes*.

The design of the HA-NFS toolkit has changed with the introduction of ServiceGuard 11.13. See the *ServiceGuard Version A.11.13 Release Notes*. With this ServiceGuard version a new control script support for additional products was introduced. The package control script template now provides special code for use with additional products, including MetroCluster with Continuous Access/CA, MetroCluster with EMC SRDF, and the HA-NFS toolkit. Refer to the additional product's documentation for details about how to create a package using the hooks that are provided in the control script template.

The HA-NFS product is a separate product with a set of configuration and control files that must be customized for the SGeSAP environment. SGeSAP uses the NFS-Toolkit interface to make the NFS server in an SAP WAS environment highly available. The detailed description of how to set up SGeSAP is now based on the new HA-NFS concept. This means that all installation steps in Chapter 2 of this document are based on the configuration setup for SGeSAP with ServiceGuard Version 11.13 and HA-NFS toolkit 1.1 for 11.x.

Networking

Communication with the highly available components takes place by using special IPaddresses, called *relocatable IP addresses*. They are handled by ServiceGuard. Think of them as addresses of a package rather than host machine addresses.

Some of the SAP WAS processes require that hostnames are not longer than eight characters and do not contain special characters.

One additional IP address is the minimum address requirement for the one package concept, two IP addresses are needed for the two package solution (one for each package). If you are planning to use a frontend LAN and a server LAN as suggested by SAP WAS, double the number of IP addresses, that is, use two or four.

You can reduce the number of additional IP addresses with the two package concept if you do not want to access the database directly through the frontend LAN. You can save another IP address, if you do not have frontends that directly connect to the central instance, but only access your SAP WAS system through different application servers.

The reason for relocatable IP addresses on both networks is, that a switchover must remain transparent to both the frontends and the additional application servers.

The complete system is installed if there is no server LAN at all, that is, all profiles only refer to addresses of the frontend LAN. Static routing of the relocatable addresses from the client LAN to the server LAN redirect the communication traffic of the application servers and the database to the backend LAN.

Directory Structures

This section provides a deeper understanding of the directory layout. For information about LVM setup, refer to the section, “Planning the Volume Manager Setup” on page 49.

Depending upon your particular setup, the following important groups of directories need special treatment:

- Common Directories that are Kept Locally
- Directories that Reside on Shared Disks
- Directories that are Handled by the Automounter

Common Directories that are Kept Locally

Use a standard setup for the following common directories and their files that are kept locally on any host of the cluster:

- `/etc/cmcluster`—the directory in which ServiceGuard keeps its configuration files and the runtime directory
- `/home/<SID>adm`—the home directory of the SAP WAS system Administrator
- `/oracle/<SID>`—the directory in which the ORACLE client software resides

  If you install SAP WAS release 4.0A or higher with ORACLE database, you have this additional local directory on each internal application server.

The contents of this group of directories must be synchronized manually on all hosts of the cluster. Do not automatically make `/home/<SID>adm` the same on all of the hosts. For example, it is possible to install an additional application server on a host of the cluster that is not be part of any package. If it is local to its host, the SAP WAS startup scripts are only needed on its dedicated host. You do not need to distribute them to other hosts.

Change the SAP WAS startup scripts `startsap_<local_hostname>_<id>` individually in the home directory of `<SID>adm` on all cluster nodes, to configure different startup profiles. The central instance can be configured differently, depending on which

node it is actually started. Be careful with this option. You have to make sure by yourself that the central instance is capable of doing its work in any of the possible configurations.

The standard HP-UX configuration files are local. To prevent malfunction, never delete the mutual `.rhosts` entries of the root user and `<SID>adm` on any of the nodes.

The SAP WAS option to introduce a local copy of the `ORACLE_HOME` directory `/oracle/<SID>` allows you to use older versions of the ORACLE client software in combination with a more recent ORACLE server release. This provides more flexibility and is useful if the operating system releases are supported by a different client than the server component.

In switchover environments only external application servers can take advantage of this. The server component has to be able to run on all nodes. You cannot use an operating system release with a server component that is not supported. This is true for any host within the cluster. Within a cluster, the client and server components must have the same release number.

This is important, because `/oracle/<SID>` also acts as a mountpoint for the directories of the server component. The client software residing on a local disk is no longer accessible after a mount takes place. This happens, if the database switches to a host on which an application server is running. The whole process remains transparent to the running application server because the database server directories deliver the same files at the same places. Every file is where the application server expects it. Already opened files can still be accessed, even though they are no longer visible in the directory tree. Please note, that the application server will only have read access. No writing will occur.

NOTE

Because `/oracle/<SID>` is needed as a free mountpoint by the package, it must not be a mountpoint for a logical volume containing the client ORACLE files. Alternatives include using `/oracle` as a mountpoint for the local filesystem or using a link. Using crossmounting with `/oracle/<SID>` impacts performance even during normal operation. Crossmounting is not needed since only read access occurs on the client side.

Directories that Reside on Shared Disks

Changing shared disk directory files on any host of the cluster affects the whole cluster. Files in the following directories and all subdirectories are normally shared:

- /usr/sap/<SID>/DVEBMGS<instance_id>
- /export/usr/sap/trans
- /export/sapmnt/<SID>
- /export/informix or /oracle/<SID>

The files in these directories are only available on a host if the package they belong to is running on it. ServiceGuard switches them to another node with the package. Under a two package concept the three /export and /oracle directories belong to the database package. The /usr directories belong to the central instance package.

Never use /usr/sap/<SID> as mountpoint on a shared disk. You risk making the Instance Directory of an internal application server unreachable if a switchover takes place.

All filesystems mounted below /export are part of the crossmounting. Please note that /oracle/<SID> is treated differently for performance reasons.

Directories that are Handled by the Automounter

Directories handled by the automounter are mounted automatically as needed. This is true not only for the nodes of the cluster, if you use external application servers, they will also need them. Automounter directories are:

- /sapmnt/<SID>
- /usr/sap/trans
- /informix

These directories are NFS-mounted from their equivalents inside of the /export directory of the nodes which run the packages. The automounter uses the relocatable IP addresses. This ensures that the directories are quickly available again after a switchover.

There are two important issues concerning these directories:

- The directories below /export are exported without root permissions.

  This happens according to the recommendations of SAP WAS and enhances the security of the installation. The effect is, that the root user cannot modify these directories or their contents. With standard permissions set, the root user cannot even see the files. This is not an error and the system runs without problems.

  If you want to modify anything as root, use the equivalent directory below /export on the host the package runs on.

- If the database package is halted, you cannot log in as <SID>adm unless you keep the binaries local.

  The reason for this is, that /usr/sap/<SID>/SYS/exe is part of the path of <SID>adm. Without local binaries, this directory links to /sapmnt/<SID> which is handled by the automounter. The automounter cannot contact the host belonging to the relocatable address that is configured because the package is down. The system will hang. To prevent this, always keep local copies of the executables.

  On a HP-UX system, all SAP WAS executables should be kept locally on all nodes. This avoids problems during a failover. You have to setup sapcpe which allows SAP WAS to distribute new executables after an SAP WAS upgrade.

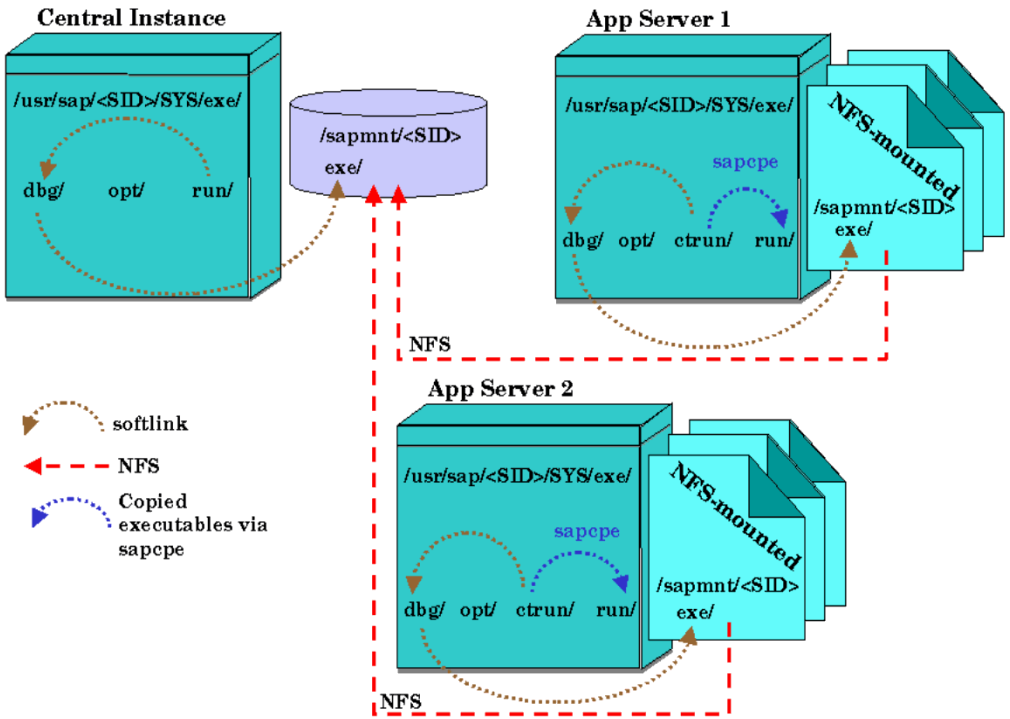
SAP WAS Local Executables

In clustered SAP WAS system environments it is required to install local executables. For example, starting SAP WAS instances from local executables is quicker than starting them from centrally stored executables.

To automatically update the executables SAP WAS includes the sapcpe mechanism. With every startup of an SAP WAS system, sapcpe matches new executables stored centrally with those stored locally.

Figure 1-4 shows a filesystem layout with sapcpe.

Figure 1-4 sapcpe Mechanism for Executables



ServiceGuard Package Concepts

Specific files are required to build ServiceGuard packages. The files required apply to both the SGeSAP one package model and two package model. Figure 1-5 illustrates the required configuration files, and the questions they answer, for an SGeSAP application package in ServiceGuard.

Figure 1-5 Configuration Files Needed to Build Cluster

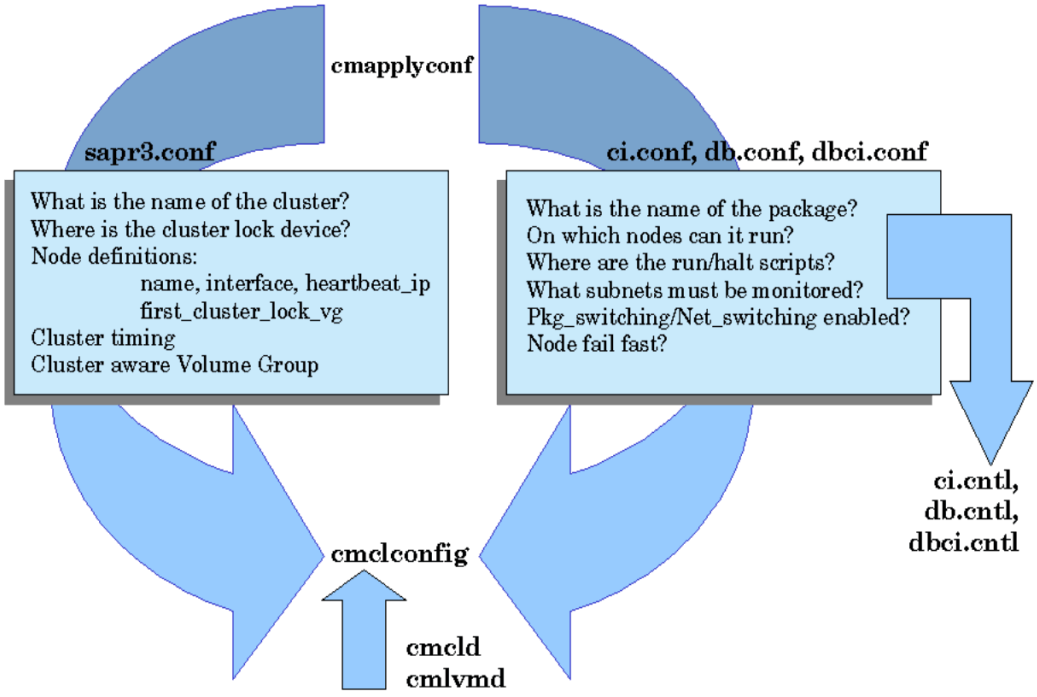
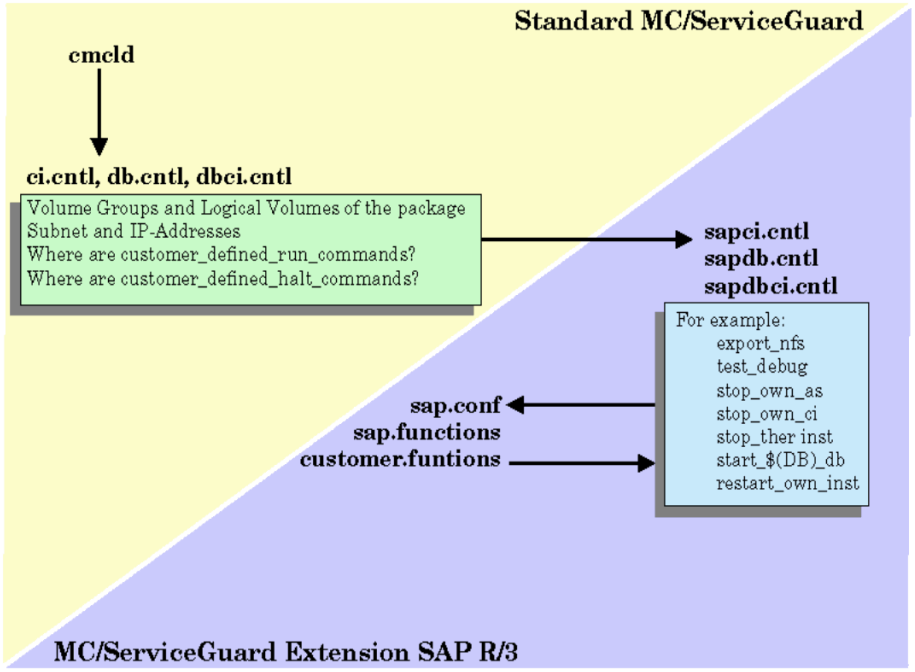


Figure 1-6 illustrates the required control files for an SGeSAP application package in ServiceGuard.

Figure 1-6 Control Files Containing Runtime Logic



Extension Files for the One Package Model

In the one package model, the SAP WAS functionality—database and central instance—along with the highly available NFS server are all been placed in one ServiceGuard package.

The SGeSAP scripts of the one package concept are organized as follows:

- `/etc/cmcluster/sapr3.conf`—cluster configuration file of ServiceGuard.
- `/etc/cmcluster/<SID>/dbci.conf`—package configuration file based on the current ServiceGuard configuration file.
- `/etc/cmcluster/<SID>/dbci.cntl`—package control file based on the current ServiceGuard file. It must call `<SID>/sapdbci.cntl` in the `customer_defined_[run|halt]_cmds` section.
- `/etc/cmcluster/<SID>/sap.conf`—contains the actual configuration and options of the SGeSAP Integration Scripts. It needs to be customized.
- `/etc/cmcluster/sap.functions`—contains all standard runtime logic used by one package and two package configurations. Do not modify this file!
- `/etc/cmcluster/customer.functions`—contains all functions which can be modified for special setups. This file may also contain your own additional functions. If you have to change functions in `/etc/cmcluster/sap.functions`, copy them over to `/etc/cmcluster/customer.functions` and modify them there. Never modify `/etc/cmcluster/sap.functions` itself. Any function provided in `customer.functions` overrides its equivalent in `sap.functions`.
- `/etc/cmcluster/<SID>/sapdbci.cntl`—contains database and SAP WAS specific control logic. This file is valid for one package configurations only. Sources include `<SID>/sap.conf`, `customer.functions`, and `sap.functions`.

Use the one pkg concept for all configurations where you can put the database and central instance on one node and have an equally sized node as a backup. During normal operation, backup nodes can be used as an idle standby, application server host, or test system.

If you are planning to distribute the database and central instance on two nodes in the near future apply the two package concept.

Extension Files For the Two-Package Model

In the two package model, the SAP WAS functionality is separated into two ServiceGuard packages. One for the database (DB) and the other for the SAP WAS central instance (CI). The database package contains the filesystems for the NFS mount points. The SGeSAP scripts of the two package concept are organized as follows:

- `/etc/cmcluster/sapr3.conf`—cluster configuration file of ServiceGuard.
- `/etc/cmcluster/<SID>/db.conf`—database package configuration file based on the current ServiceGuard configuration file. The provided file is for reference only, create your own.
- `/etc/cmcluster/<SID>/ci.conf`—central instance package configuration file based on the current ServiceGuard configuration file. The provided file is for reference only, create your own.
- `/etc/cmcluster/<SID>/db.cntl`—database package control file based on the current ServiceGuard configuration control file. It must call `<SID>/sapdb.cntl` in the `customer_defined_[run|halt]_cmds` sections.
- `/etc/cmcluster/<SID>/ci.cntl`—central instance package control file based on the current ServiceGuard control file. It must call `<SID>/sapdbci.cntl` in the `customer_defined_[run|halt]_cmds` sections.
- `/etc/cmcluster/<SID>/sap.conf`—contains the actual configuration and options of the Integration Scripts. It needs to be customized.
- `/etc/cmcluster/sap.functions`—contains all standard runtime logic used by one package and two package configurations. Do not modify this file!.
- `/etc/cmcluster/customer.functions`—contains all functions which can be modified for special setups. This file may also contain your own additional functions. If you have to change functions in `/etc/cmcluster/sap.functions` copy them over to `/etc/cmcluster/customer.functions` and modify them there.

Never modify /etc/cmcluster/sap.functions itself. Any function provided in customer.functions overrides its equivalent in sap.functions.

- /etc/cmcluster/<SID>/sapci.cntl—contains SAP WAS specific control logic. This file is valid for a two package configuration only. Sources include <SID>/sap.conf, customer.functions, and sap.functions.
- /etc/cmcluster/<SID>/sapdb.cntl—contains database control logic. This file is valid for a two package configuration only. Sources include <SID>/sap.conf, customer.functions, and sap.functions.

Extension Files for an Application Server Package

The creation of an Application server package only makes sense if a one-package or two-package model has already been implemented. The needed Application server files are organized as follows:

- /etc/cmcluster/<SID>/ap<SID><INSTNR>.conf—capplication server package configuration file based on the standard ServiceGuard configuration file.
- /etc/cmcluster/<SID>/ap<SID><INSTNR>.cntl—application server package control file based on the standard ServiceGuard control file. It must call /etc/cmcluster/<SID>/sapap.cntl in the customer_defined_[run|halt]_cmds section.
- /etc/cmcluster/<SID>/sapap.cntl— contains SAP Application server specific control logic. This file applies to the Application server package only and must not be edited.

Server Consolidation

Server consolidation in an SAP WAS environment is a strategy to put multiple SAP WAS systems on a single physical server, a single host system. Since HP introduced HP-UX 11.00 with its 64-bit technology, addressing memory is no longer a limitation. The availability of 64-bit compiled SAP WAS versions now enables a full blown 64-bit solution for SAP WAS. This results in having either multiple SAP WAS clients running on a single SAP WAS server or having multiple 2-tier SAP WAS systems running on a single host. Additional SAP WAS dialog instances can also run on the same server—in both scenarios.

Clustering a consolidated SAP WAS server is different from clustering a single SAP WAS instance. SGeSAP Version 3.1 supports dialog instance handling and shared memory resource handling on a consolidation host.

Dialog Instance Handling

With SGeSAP 3.1, SAP WAS dialog instance treatment during CI / DB / DBCI package startup and stop has changed. SGeSAP now supports dialog instances on a host where CI / DB / DBCI packages are currently started as listed in Table 1-1.

Table 1-1 Dialog Instance Treatment During Startup and Shutdown

ASTREAT	Restart	Final_ Stop	Active	Runpkg DB	Haltpkg DB	Runpkg CI	Haltpkg CI
0	0	0	0	nop	nop	nop	nop
1	0	0	1	CI-start	nop	st_if_nl	nop
2	0	1	0	nop	nop	nop	stop
3	0	1	1	CI-start	nop	st_if_nl	stop
4	1	0	0	stop	nop	stop	nop
5	1	0	1	stop CI-start	nop	stop start	nop
6	1	1	0	stop	nop	stop	stop

Table 1-1 Dialog Instance Treatment During Startup and Shutdown

ASTREAT	Restart	Final_ Stop	Active	Runpkg DB	Haltpkg DB	Runpkg CI	Haltpkg CI
7	1	1	1	stop CI-start	nop	stop st_if_nl	stop
nop = no operation st_if_nl = start if *not* on local host							

Shared Memory Resource Handling

SGeSAP 3.1 provides shared memory resource cleanup policies as listed in Table 1-2.

Table 1-2 Shared Memory Cleanup Policies and their Effects

CLEANUP_ POLICY	CI Resources	DB Resources	AppServer Resources
lazy	no action	no action	no action
normal	use `cleanipc` to remove unused own resources	unused Oracle SGA is removed	use `cleanipc` to remove unused own resources
strict	remove any resource belonging to any SAP WAS instance	remove any resource belonging to any database	user `cleanipc` to remove unused own resources

SGeSAP Product Files

All the files that are available after the installation of SGeSAP can be found in `/opt/cmcluster/sap`. The files that are installed are:

- The standard SGeSAP run-time functions and configurable functions repository container in `/opt/cmcluster/sap`:

  `sap.functions`—contains standard SAP WAS functions. Do not edit this file!

  `customer.functions`—contains functions that can be modified for special setups.

- The run-time configuration options and control logic in `/opt/cmcluster/sap/SID`:

  `sap.conf`—contains configuration and options for SGeSAP

  `sapci.cntl`—contains SAP WAS specific control logic for two package configurations

  `sapdb.cntl`—contains database specific control logic for two package configurations

  `sapdbci.cntl`—contains database and SAP WAS specific control logic for one package configurations

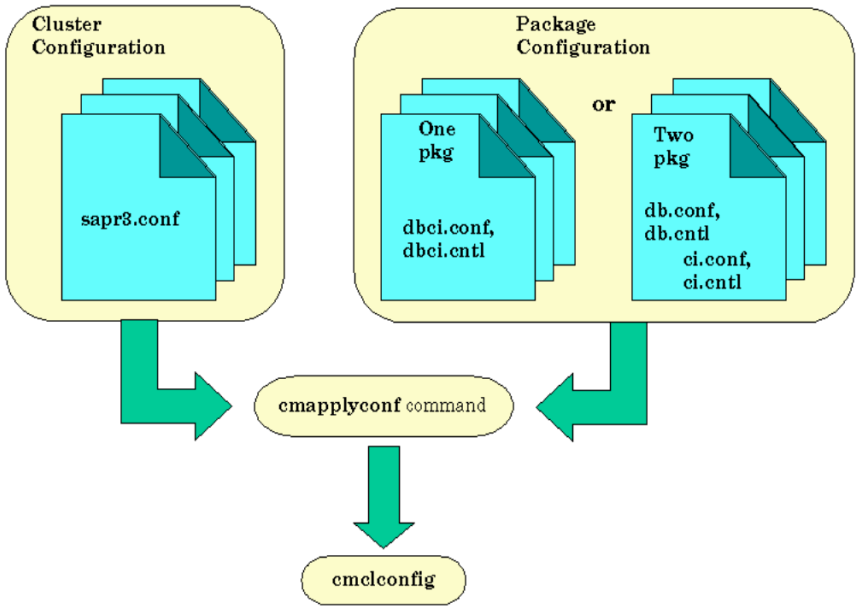
  `sapap.cntl`—contains the Application specific control logic for an application server package

  `startdb.sql`—contains SQL sample that has to be merged into the SAP "startdb" script.

  The `/opt/cmcluster/CCmon` directory contains the file needed for the setup of the Cluster Consitancy monitor for SGeSAP. Refer to Cluster Consitancy Chapter within this documentation for more details.

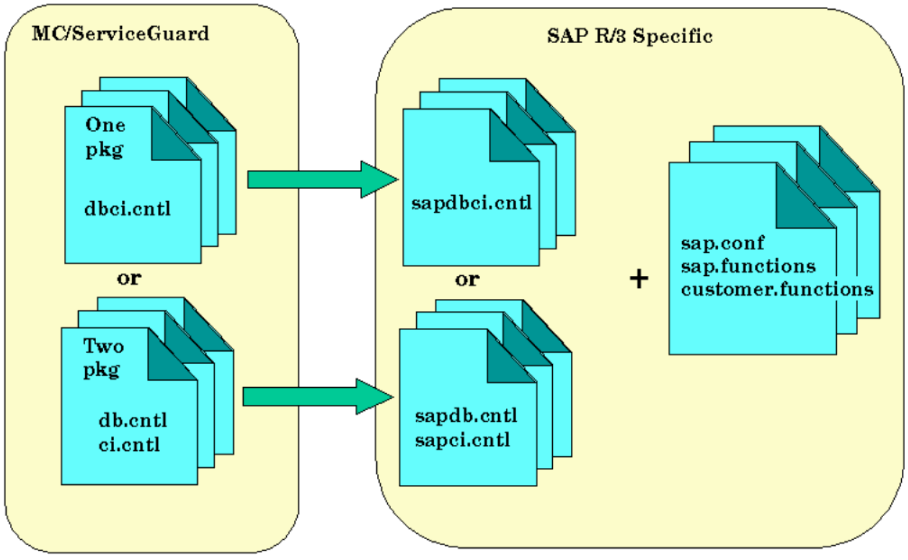
Figure 1-7 shows that the configuration files are used as input for the standard ServiceGuard `cmapplyconf` command to generate the `cmclconfig` file that is used to manage the cluster.

Figure 1-7 Configuration Files



The SGeSAP files work in tandem with standard ServiceGuard files to monitor the health of the cluster and packages, and to move packages to another node in case of failure. Figure 1-8 shows the relationship between the standard ServiceGuard files and the SGeSAP scripts and control logic.

Figure 1-8 Control Files Containing Run-time Logic



Cluster Consistency Monitor for SGeSAP Environments

The Cluster Consistency Monitor is intended to be an adaptive toolkit. This means that during the initial run, all the generic resources like the shared volume groups and the file systems of an application, which may influence the proper operation of that application in the cluster, are specified in the cluster resource DB. Then, individual configurations for a certain application or in a certain customer environment can be added whenever necessary by simply adding the resource definition to the resource DB. This flexibility enables the Monitor to work with all kinds of applications. Even in uniform applications like SAP WAS, many customers have different individual setups and need flexibility to address this.

The Cluster Consistency Monitor must be run periodically in order to maintain the cluster failover capability. The current implementation of the monitor supports a "Learning Mode" which gives detailed Information about a discovered failure as well as some hints how to fix the problem. The knowledge of configuring and maintaining a cluster has been put in the Monitor. On the other side, in case a cluster is well tuned, this feature may be switched off and the monitor checks only on changes. Of course, potential problems may be fed into a management system like HP ITO.

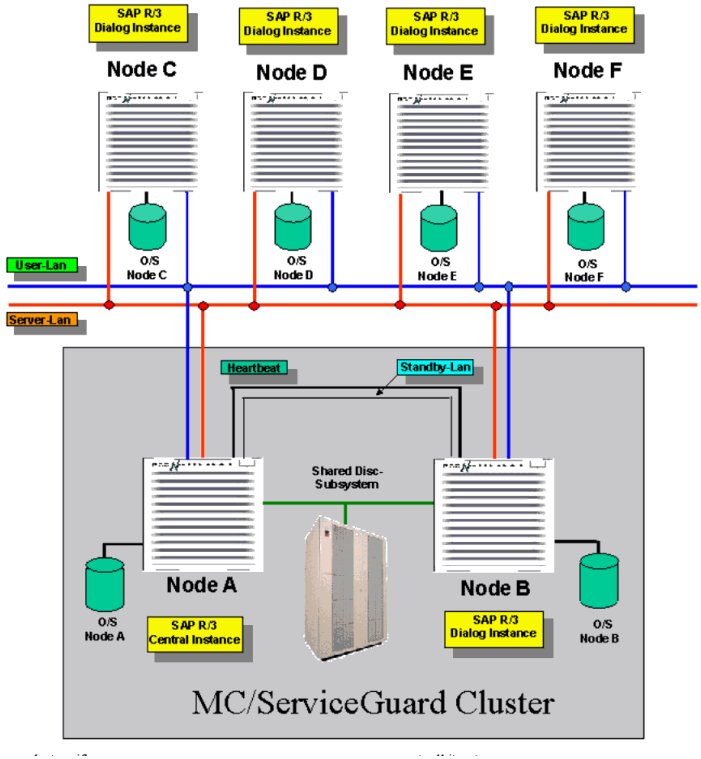
In a small installation consisting only on a few clusters, the output should be delivered in an ASCII report. It might be however useful to have a central maintenance server in a large IT infrastructure. The current implementation of the monitor supports an ASCII report for standalone operations as well as HTML reports for centralized infrastructure repositories.

The basic function of the Cluster Consistency Monitor is to compare resource configurations of nodes. Each node that is monitored runs a program which reads a configuration profile about named resources and creates a resource database. These resource databases are then compared by another program and possible error conditions are presented in an ASCII or HTML report which can be used for troubleshooting.

Why Cluster Consistency Monitor is Needed

Clusters exist to maintain the operation of an application even in case a hardware resource failure. The SAP WAS environment is very complex, depending on a number of resources on multiple servers. Some of these resources run on the nodes in the cluster, while others are on non-cluster nodes. Figure 1-9 depicts a typical view of a cluster configuration for an SAP WAS application.

Figure 1-9 Cluster Configuration for an SAP WAS Application

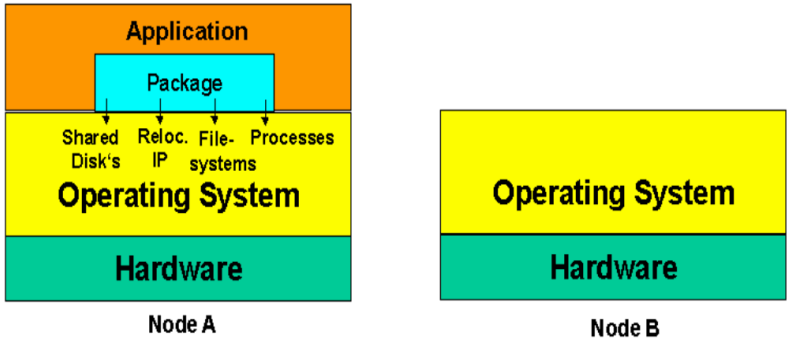


Because of this complexity, the precise definition of resources on each server in and out of the cluster must be managed carefully. The **Cluster Consistency Monitor** is a tool that assists in maintaining consistent relationships among all component servers.

Challenges in a Cluster Environment

While we typically find the Central Instance and the Database of a SAP WAS System along with some important filesystems shared by NFS, there are a number of other dialogue instances associated with the cluster. Figure 1-10 shows the applications resources, filesystems, volume groups, IP name and process control have been put in a package and outside of a cluster configuration for a SAP WAS application.The start and stop of these instances may also be controlled by the SGeSAP solution.

Figure 1-10 Dialogue Instances of the Cluster Configuration



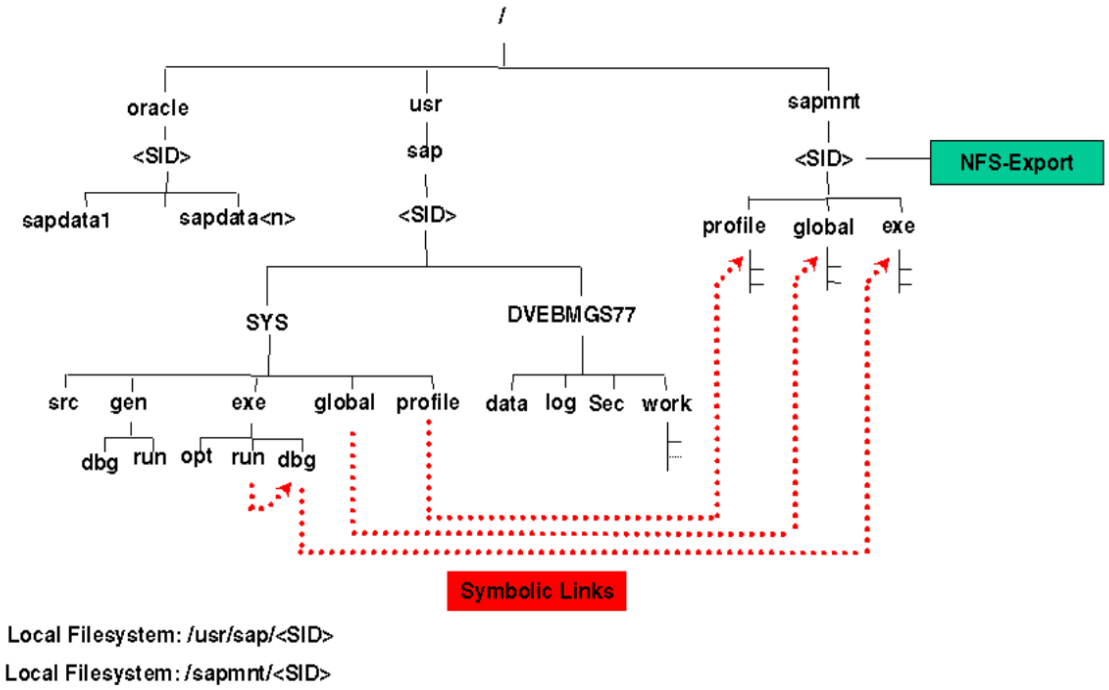
The package controls these resources and relocates it upon a node failure to another node. While important application components are in the shared devices, the definition for the shared devices, as well as the configuration of the relocatable IP address have to be maintained manually on each node capable of running this application.

Application Perspective on Resources in the OS

The following description is based on the the WAS application example shown in Figure 1-10. Each installation of an WAS application has many dependencies on the configuration of a hosting operating system.

Figure 1-11 depicts an SAP WAS installation at the level of the node's file systems. The bullet list shows items that must be consistent between nodes to allow failover.

Figure 1-11 SAP WAS Installation into a Nodes File Systems



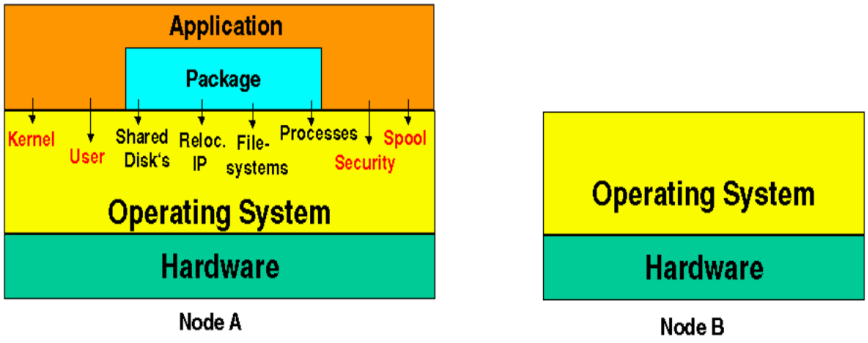
The following list of characteristics are illustrated in figure 1-13:

- Directory and File Structures
- Symbolic Links
- Kernel II Parameter
- Permissions
- NFS Configurations
- Processes
- Traces and Logfiles
- Shared Memory Segments

- Security Settings
- Resource Consumption
- User Configurations
- Spool Configurations

Figure 1-12 shows other application resources, such as “Kernel, User, Security, and Spool,” outside of the cluster configuration.

Figure 1-12 SAP WAS Application Installation with Other Resources



NOTE Any difference in resource configurations between nodes in a cluster may cause problems to an application upon a fail over. These problems may appear immediately after a switch and may disable the start of the application but may also appear only in certain situations during the operation of an application.

Features of the Cluster Consistency Monitor for SGeSAP Environments

The implemented features of the Cluster Consistency Monitor are as follows:

- The information to examine is configurable in order to support all kind of applications.

- Due to the flexibility in configuring resources, any kind of cluster may be monitored. This includes the basic ServiceGuard Cluster, MetroCluster, and installations of shadow DB and HP Sommersault units as well.
- Updates on the configurations are easy.
- The compare run may be performed automatically as a background task or individually from the command line.
- The Cluster Consistency Monitor may be used to trigger an event for a management system like ITO upon an error.
- Results are provided in a report. Output format of this report may be HTML or ASCII.
- Output reports may contain error interpretation text in order to help during interpretation of the compare.
- All resources configured in a package may be easily added to the resource DB by using the statement `ANALYSE PACKAGE` upon creation of the Resource DB.
- It is possible to suppress acknowledged error messages. Changes in a cluster setup, which may be OK, and may be acknowledged by implementing an entry in an error exclusion table in order to suppress them. No trigger to external systems would be created on suppressed messages.
- The Cluster Consistency Monitor detects and reports all kinds of setup problems of the monitor itself.
- The operation of the cluster monitor does not impact applications, which are currently running due to the resource consumption of the monitor is very low. Typically, the comparator takes less than 10 seconds in comparing two nodes, which includes the creation of the resource DB.
- The use of the Cluster Consistency Monitor avoids fail over tests. Due to the fact, that on a fail over test, only the startup of the package on a secondary node is examined, potential problems which may arise in case the application runs for a longer period on that node are not recognized. The Cluster Consistency Monitor detects even that class of problems as it simulates the view an application would have toward the configurations of operating systems.

- The storage format of the resource DB is very dense. It allows fast transfer even over wide area networks. Due to that density, memory consumption during the comparator run or storage consumption for the resource DB is very low.
- The current implementation supports HP-UX 10.20 and 11.00, 32 Bit and 64 Bit.

Installation and Setup of the Cluster Consistency Monitor

Refer to OS998 in Chapter 2 for information on installing the Cluster Consistency Monitor. More detail about installation and setup of the monitor is found in the monitor handbook located after installation in the following directory: `/opt/cctool/doc/CCmon_Handbook.doc`.

Metropolitan Clusters and SGeSAP

What is a Metropolitan Cluster?

A MetroCluster is one of Hewlett Packards Disaster Tolerant solutions. Two types of Metropolitan Cluster are provided, depending on the type of data storage and replication you are using. You will need either MetroCluster with Continuous Access CA or MetroCluster EMC Symmetrix Remote Data Facility SRDF if you want to secure an application whose data is on an HP-XP array or an EMC Symmetrix ICDA which data is replicated to a second HP-XP array or EMC Symmetrix ICDA using Continuous Access (CA) or Symmetrix Remote Data Facility (SRDF).

Beginning with SGeSAP 3.05, ServiceGuard 11.13, MetroCluster 3.0 and HA-NFS 1.1, the integration of all four products is possible without any change of the standard ServiceGuard control files. The ServiceGuard 11.13 control file version now has a call out function to MetroCluster embedded.

What is a MetroCluster doing?

In the event of node failure, the integration of MetroCluster/CA or MetroCluster/SRDF with the SAP application will allow the application package to automatically fail over to the recovery site:

- Among local host systems that are attached to the same XP / Symmetrix.
- Between one system that is attached locally to the primary XP / Symmetrix and another "remote" host that is attached locally to the secondary XP ;/ Symmetrix.

NOTE Failover and failback methods are both supported.

MetroCluster and SGeSAP: Supported Configurations

- ServiceGuard 11.13 or later

- MetroCluster with Continuous Access XP version 3.0 or later or MetroCluster with EMC SRDF version 3.0 or later
- SGeSAP 3.05 or later
- HA-NFS 1.1 or later

MetroCluster: Cluster Architecture Constraints

All host systems in a MetroCluster environment must be members of a single ServiceGuard cluster.

Either a single Data Center architecture without an arbitrator or a Three Data Center architecture with one or two arbitrator systems can be implemented. The arbitrator(s) are not physically connected to the storage units. In a three data center architecture, no cluster lock device is used since it does not work across the ESCON link.

Three hosts are allowed for each data center. In a campus environment, the same number of systems must be present in each of the two data centers whose systems are connected to the storage units.

The ESCON fibre that realizes the dual ESCON links between the ICDAs can extend to 40km.

Combining MetroCluster and SGeSAP

Beginning with SGeSAP 3.05, ServiceGuard 11.13, MetroCluster 3.0 and HA NFS 1.1, the integration of all four product is possible without any change of standard control files. The ServiceGuard 11.13 control file version has a call out function to MetroCuster embedded.

The standard ServiceGuard 11.13 package control file controls the runtime steps during a package setup. Before the SGeSAP specific logic step is started within the `customer_defined_runcommands` section, the package control file has hooks embedded for MetroCluster and HA NFS specific runtime steps. This allows a smooth integration of all products (ServiceGuard, MetroCluster, HA-NFS and SGeSAP) that are needed to run a SAP Application in a disaster tolerant environment.

The following example shows how this is technically realized:

- *verify_physical_data_replication:* This function checks the availability of a file called "`DRCheckDiskStatus`". If the file exists, the MetroCluster specific runtime steps will be executed.

- *verify_ha_nfs $1:* This function checks the availability of the $HA_NFS_SCRIPT file that is defined in the controlfile. In SGeSAP environments the convention is that these files are to be called "hanfs.db" for the two package SGeSAP concept and "`hanfs.dbci`" for the one package SGeSAP concept. If this file exists, the HA-NFS specific runtime steps will be executed.
- *customer_defined_run_cmds:* This function is used to call the SGeSAP specific runtime steps. It is required to add the script that has to be called appropriately. In SGeSAP environment this script is called "sapdbci.cntl for the one package SGeSAP concept and "`sapdb.cntl`" / "sapci.cntl" for he two package concept. More details can be found in Chapter 2, Step by Step Installation Guide, of this manual.

All other implementation steps in regards to MetroCluster, HA NFS and SGeSAP are standard and have to be performed according to the standard documentation. Refer to *Designing Disaster Tolerant HA Clusters* for complete details.

A MetroCluster environment may increase package startup time significantly. Packages with many disk devices will take longer to start up than those with fewer devices due to the time needed to get device status from the XP / Symmetrix array. Clusters with multiple packages that use devices on the XP / Symmetrix will cause package startup time to increase when more than one package is starting at the same time. You have to add the time needed to restart all attached WAS application servers to get an estimation of the total package startup time. It is strongly recommended to disable the timeout mechanism of ServiceGuard that monitors the switchover.

If you are planning to control additional SAP application servers out of the SAP SG packages, an analysis of the physical machine environment for the APP-servers is required. In disaster tolerant configurations it is required to run the complete SAP production, including all the SAP application servers in the primary DC and fail it over completely to the secondary DC in case of a disaster. This requires changes in the APP-server configuration within the SGeSAP environment. You need a sap.conf for the primary DC and a `sap.conf` file for the secondary DC. Refer to chapter 2 of this documentation to get more details on APP-server configuration within the SGeSAP environment.

It is required to follow all configuration steps that are referenced in the MetroCluster and HA-NFS documentation. These documents can be found at `http://docs.hp.com/hpux/ha/index.html`.

2 Step by Step Installation Guide

This chapter describes how to configure and install highly available SAP components based R/3 kernal or SAP WAS systems on HP-UX 11.x using ServiceGuard. The process is split into the following logical tasks:

- Planning the Volume Manager Setup
- Preparing the Integration
- HP-UX Configuration
- SGeSAP Files Configuration
- ServiceGuard Configuration
- Automounter/Auto FS Configuration
- Database Configuration
- SAP WAS System Configuration

The tasks are presented as a sequence of steps. Each mandatory installation step is accompanied by a unique number of the format *XXnnn*, where *nnn* are incrementing values and *XX* indicates the step relationship, as follows:

- ISnnn—Installation Steps - mandatory
- OSnnn—Optional Steps
- ORnnn—ORacle database only steps
- IRnnn—InfoRmix database only steps
- DBnnn—DB2 database only steps
- SDnnn—SAPDB database only steps

Whenever appropriate, HP-UX sample commands are given to guide you through the process in as detailed a manner as possible. It is assumed that hardware as well as the operating system and ServiceGuard are already installed properly on all cluster hosts. Sometimes a condition is specified with the installation step. Follow the information presented *only* if the condition is true for your situation.

NOTE

For installation steps in this chapter that require the adjustment of SAP specific parameter in order to run the SAP WAS system in a switchover environment usually example values are given. These values are for reference ONLY and it is recommended to read and follow the appropriate SAP OSS notes for SAP's latest recommendation. Whenever possible the SAP OSS note number is given.

Planning the Volume Manager Setup

Volume managers are tools that let you create units of disk storage known as storage groups. Storage groups contain logical volumes for use on single systems and in high availability clusters. In ServiceGuard clusters, package control scripts activate storage groups.

With ServiceGuard 11.13 two volume managers can now be used: the standard Logical Volume Manager (LVM) and the Veritas Volume Manager (VxVM). SGeSAP can also be used with both volume managers.

The following steps describe the standard setup for the LVM volume manager. For more details refer to the "Integrating VxVM with ServiceGuard" manual that can be found on `http://docs.hp.com/hpux/11i` under the Veritas Volume Manager section.

IS020 **Installation Step:**

Record all the minor numbers already in use on the cluster hosts. See Table 2-1.

The device minor numbers of the shared volume groups must be the same on all cluster hosts. They must differ from all other volume group device minor numbers used on the cluster hosts.

To find these minor device numbers on each cluster hosts, type:

```
ll /dev/vg*/group|cut -c 44-45
```

Table 2-1 **Hosts and Device Minor Numbers**

hostname	device minor numbers
. . .	
. . .	
. . .	
. . .	

IS030 **Installation Step:**

Create new special files for the volume groups as needed.

Refer to Table 2-1, check that all the minor numbers for one host are unique for that host. If any host uses the same number twice create new special files with a unique number.

IS040 **Installation Step:**

Specify the needed volume groups in another table.

For the standard setup there should be at least three cluster volume groups:

- one volume group for filesystems that are specific to the Central Instance
- one volume group for more general SAP WAS filesystems
- one volume group for the database specific filesystems

Two shared volume groups would also work but this is not recommended because of lack of flexibility. If you put all logical volumes in one volume group, you will be restricted to the one package concept. You should end up with a table similar to Table 2-2.

NOTE If there is more than one SAP WAS system in the cluster then `/usr/sap/trans` may be outside of this configuration.

Table 2-2 Volume Groups Needed for ServiceGuard Packages

SG package	VG Name	lvol Names	Later Mount Point	Device minor number
ci<SID>	vgsap<SID>	lvsap<SID>	/usr/sap/<SID>/DVEBMGS<INSTNR>	08
ap<SID><INSTNR>	vg<SID><INSTNR>	lvsap<SID>	/usr/sap/<SID>/D<INSTNR>	11

Depending on your RDBMS the database for your db<SID> package would look different

Table 2-3 Exaple for ORACLE Based Installations:

SG package	VG Name	lvol Names	Later Mount Point	Device minor number
db<SID>	vgmnt<SID>	lvmnt<SID> lvtrans	/export/sapmnt/<SID> /export/usr/sap/trans	09
	vgdb<SID>	lvoracle lvstage_805 lvsaparch lvsapreorg lvsapdata1 lvsapdata2 lvsapdata3 lvsapdata4 lvsapdata5 lvsapdata6 lvoriglogA lvoriglogB lvmirrlogA lvmirrlogB	/oracle/<SID> /oracle/stage/stage_805 /oracle/<SID>/saparch /oracle/<SID>/sapreorg /oracle/<SID>/sapdata1 /oracle/<SID>/sapdata2 /oracle/<SID>/sapdata3 /oracle/<SID>/sapdata4 /oracle/<SID>/sapdata5 /oracle/<SID>/sapdata6 /oracle/<SID>/origlogA /oracle/<SID>/origlogB /oracle/<SID>/mirrlogA /oracle/<SID>/mirrlogB	10
ap<SID><INSTNR>	vg<SID><INSTNR>	lvsap<SID>	/usr/sap/<SID>/D<INSTNR>	11

Table 2-4 Example for INFORMIX Based Installations:

SG package	VG Name	lvol Names	Later Mount Point	Device minor number
db<SID>	vgmnt<SID>	lvmnt<SID> lvtrans	/export/sapmnt/<SID> /export/usr/sap/trans	09
	vgdb<SID>	lvinfxx ...	/export/informix... ...	10

Table 2-5 Example for DB2 Based Installations:

SG package	VG Name	lvol Names	Later Mount Point	Device minor number
db<SID>	vgmnt<SID>	lvmnt<SID> lvtrans	/export/sapmnt/<SID> /export/usr/sap/trans	09
	vgdb<SID>	lvdb2 lvsapdata1 lvsapdata1 lvsapdata1 ...	/db2 DB container DB container DB container ...	10
		lvsaplog1 lvlogarchive1 lvlogretrieve lvsapdatatemp lvdb2admin lvdb2dump	/db2/<SID>/log_dir1 /db2/<SID>/log_arch1 /db2/<SID>/log_retrieve1 /db2/<SID>/sapdatat /db2/db2as /db2/<SID>/db2 dump	

Using the one package concept, all the volume groups will belong to one package. For reference we call this package dbci<SID> or the DBCI package.

Using the two package concept, vgsap<SID> belongs to the Central Instance package (CI). The CI package contains all logical volumes that are exclusively needed by the Central Instance. vgsapmnt<SID> and vgdb<SID> are configured to belong to the database package (DB).

If you use only two cluster volume groups, put /export/sapmnt/<SID> and /export/usr/sap/trans in the volume group which belongs to the node that runs the HA NFS server (this should be the database package). If you plan to use more than one SAP WAS system using the same transport directory /usr/sap/trans, do not configure it with an SAP WAS system package. If you stop the SAP WAS system package, other systems will be affected, too. They cannot be restarted and transports are impossible. CCMS functionality should be used with care. For additional /export/usr/sap/trans security use the HP provided NFS toolkit. When using the Application server package concept you should setup a separate volume group for this package. This volume group vg<SID><INSTNR> consists of one logical volume called lvsap<SAP>. It holds the work, log and data directories for this specific instance.

If you have more than one system, place /oracle/stage/stage_nnn and /usr/sap/put on separate volume groups created on shared drives. These directories should not be added to any package. This ensures that they are independent from any SAP WAS system and you can mount them on any host by hand if needed.

If SAP WAS is not installed yet, specify values in the device minor column that are different from all values you identified in Step IS020. Use Table 2-1 as a template for recording the device minor values.

Table 2-6 Device Minor Values

SG package	VG name	lvol names	later mount point	Device Minor Number
. . .				
. . .				
. . .				
. . .				

Preparing the Integration

If the SAP WAS system you are going to make highly available is already installed, proceed section “Steps if SAP WAS is Already Installed.” If the SAP R/3 system you are going to make highly available is not already installed, proceed to section “Steps if SAP WAS is Not Installed Yet.”

Steps if SAP WAS is Already Installed

IS050 **Installation Step:**

- Logon as root to the system where the SAP WAS Central Instance is installed. If the database is installed on a different host, also open a shell as root on the database machine. Stop SAP WAS and the database if they are not already.
- Verify that the existing volume group layout is compliant with the needs of the ServiceGuard package(s) as specified in Table 2-6 of Step IS040. Test the following:
  - — If you install using the two package concept, make sure that database specific filesystems and Central Instance specific filesystems are separated onto different volume groups.
  - — Specify one mountpoint as /usr/sap/<SID>. For ServiceGuard, change this to /usr/sap/<SID>/DVEBMGS<INSTNR>. If /usr/sap/<SID> is the mountpoint you have to move some files to a local logical volume and change the mountpoint. For example:

```
mkdir /usr/sap/<SID>.new
cd /usr/sap/<SID>
bdf . # Remember the filesystem column.
      # It will be refered to as <dev_path> later.
find . -depth -print|cpio -pd /usr/sap/<SID>.new
cd /
umount /usr/sap/<SID>
rmdir /usr/sap/<SID>
mv /usr/sap/<SID>.new /usr/sap/<SID>
chmod 751 /usr/sap/<SID>
chown <SID>adm:sapsys /usr/sap/<SID>
cd /usr/sap/<SID>/DVEBMGS<INSTNR>
rm -r * # be careful with this
```

```
cd ..
mount <dev_path> /usr/sap/<SID>/DVEBMGS<INSTNR>
cd DVEBMGS<INSTNR>
ls
# remove everything that is different from
DVEBMGS<INSTNR>,
# Example: rm -r SYS
#          rm -r D00
cd DVEBMGS<INSTNR>
find . -depth -print|cpio -pd
/usr/sap/<SID>/DVEBMGS<INSTNR>
rm -r * # be careful with this
cd ..
rmdir DVEBMGS<INSTNR>
```

— Mark all volume groups as members of the cluster. This only works if the cluster services are already available. For example:

```
cd /
# umount all logical volumes of the volume group
vgchange -a n <vg_name>
vgchange -c y <vg_name>
vgchange -a e <vg_name>
# remount the logical volumes·
```

— The device minor numbers must be different from all device minor numbers gathered on the other hosts. Verify this by comparing numbers listed in Table 2-2 in Step IS040 to the numbers listed in Table 2-1 in Step IS020.

- Create and distribute mapfiles for all shared volume groups.

NOTE

The logical volumes that are called `lvmnt<SID>` and `lvtrans` in the sample Table 2-2 of Step IS040 are not mounted below `/export`, but below the root directory. Keep it like this. It's OK.

IS060 **Installation Step:**

Comment out the references to any filesystem that is mentioned in Step IS040 from the `/etc/fstab`.

OS070 **Optional Step:**

If there are additional internal application servers on the cluster hosts:

Logon to all cluster hosts that have application servers installed. Verify the following:

- Application servers in the cluster must have an instance number which is different from the instance number of the Central Instance. Execute the command:

  ```
  ls -d /usr/sap/<SID>/*<INSTNR>
  ```

  It should reply with No Match. If it does not, reinstall the instance with another instance ID.

- Comment out `/etc/fstab` entries to filesystems mentioned in Step IS040.

- If you use SAP WAS 4.0B or higher, `/oracle/SID` must not be a mountpoint of a local filesystem. You can use `/oracle` as mountpoint.

IS080 **Installation Step:**

The usage of local executables with the SAPCPE mechanism is required for SGeSAP environments. You can do this now, because it prevents hangs when performing a `su` to `<sid>adm` later in this installation.

Check if the Central Instance host and all application servers have a directory called, type:

`/usr/sap/<SID>/SYS/exe/ctrun`.

If the directories exists, you are done. The system is already using local executables through `sapcpe`.

If the directories do not exist, setup `sapcpe`. Refer to your OSS notes and the SAP WAS online documentation for information about how to do this.

Steps if SAP WAS is Not Installed Yet

SAP WAS is installed on one host. This remains true even if you want to distribute the database and Central Instance using the three-tier concept. ServiceGuard allows you to remove one or the other later. The host you choose now will become the primary host.

IS090 **Installation Step:**

Logon to the primary host as root.

Create the shared volume groups using the minor numbers specified in Step IS040. Use the command line interface to create the whole volume group, and later, the distribution. Do not use SAM. SAM will scramble the minor numbers. For example:

```
mkdir /dev/vgsap<SID>
mknod /dev/vgsap<SID>/group c 64 0x080000
pvcreate <pv_path>
vgcreate vgsap<SID> <pv_path>
vgchange -c y vgsap<SID>
lvcreate -n lvsap<SID> vgsap<SID>
vgchange -a e vgsap<SID>
```

Refer to your SAP WAS documentation for a description on how to distribute the filesystems on physical disks. Refer to the SAP WAS installation documents to find out about the filesystem sizes you need. Use `lvextend` and `mkfs -F vxfs` commands to create the filesystems on the shared volume groups. Create and distribute the mapfiles with `vgexport(1m)`.

NOTE `R3INST/R3SETUP` creates a file called `SAPMADB.PAR`, `SAPMAIN.PAR` or `SAPAPPL.PAR` in the installation directory. It shows needed filesystems and recommends sizes for them.

For the recommended file system refer to the latest OSS notes.

OS100 **Optional Step:**

If there is a different SAP WAS system already installed on another cluster node, propagate the additional SAP WAS HP-UX users and groups to all the nodes of the cluster. This prevents conflicts with the UIDs. The SAP WAS installation process creates users, groups and services and it performs local consistency checks. It is possible, that UIDs, GIDs or service ports already exist on other cluster hosts.

Synchronize the `/etc/passwd`, `/etc/group` and `/etc/services` files of the cluster nodes so that they have the same contents. This prevents conflicts later in the installation process.

IS110 **Installation Step:**

Logon to the primary host as root.

Create the mountpoints as recommended in the SAP WAS installation documents, except, do not use /usr/sap/<SID> as a mountpoint. Mount one level below the instance directory. For example:

```
mkdir -p /usr/sap/<SID>/DVEBMGS<INSTNR>
...
```

The filesystems should be distributed on disks and volume groups as recommended in Step IS040. But instead of using the mountpoints below /export that are suggested in Step IS040, create mountpoints below / as recommended by the SAP WAS documentation. For example:

```
mkdir -p /sapmnt/<SID>
mkdir -p /usr/sap/trans
mkdir -p /informix
...
```

Mount all created filesystems manually. Do not create entries in /etc/fstab.

IS120 **Installation Step:**

Begin SAP WAS installation now. Install the complete system on the primary machine. Refer to your standard SAP WAS documentation and OSS notes on how to do this.

If you use R3SETUP to install, the space check might fail. Modify the SPACECHECK entries in the .R3S file in the installation directory to reflect the mountpoint changes. For example:

Search for [Z_CENTRDBSPACECHECK_IND_IND] in CENTRDB.R3S.

The @SAPDIR@/@SAPSYSTEMNAME@ entries check the diskspace of /usr/sap/<SID>.

Additional free space caused by the logical volume that is mounted below at the mountpoint /usr/sap/<SID>/DVEBMGS<INSTNR> is not counted.

Do not continue until SAP WAS is properly installed on the primary node.

OS130 **Optional Step:**

If internal application servers are needed:

Setup application servers with `R3INST/R3SETUP` as usual. If you install an application server on a node within the cluster, consider the following:

- Application servers in the cluster must have an instance number which is different from the instance number of the Central Instance.
- Do not create `/etc/fstab` entries. Mount manually to avoid problems with the automounter later on.

IS140 **Installation Step:**

The usage of local executables with the SAPCPE mechanism is required for SGeSAP environments. Do this now to prevents hangs when performing a `su` to `<sid>adm` later in this installation process. Refer to your OSS notes and the SAP WAS online documentation for information.

HP-UX Configuration

Correct HP-UX configuration ensures that all cluster nodes provide the environment and system configuration required to run SAP WAS. This section describes how to distribute the SAP WAS installation configuration changes made in the previous section, “Preparing the Integration,” among the cluster nodes, that is, from all primary nodes to all other nodes.

Several of the following steps must be repeated on each node. Record the steps completed for each node, as you complete them. This helps identify errors in the event of a malfunction later in the integration process. The HP-UX configuration task is split into the following sections:

- Cluster Node Synchronization

  This section consists of steps performed on the backup nodes. These ensure that the primary node and the backup nodes have a similar environment.

  Repeat the steps in this section for each node of the cluster that is different from the primary.

  The primary host is the host where the Central Instance was installed. If your database is currently running on a machine different from this, repeat all steps once again for each node with the database machine as primary host. This distributes the database configuration, too.

- Cluster Node Configuration

  This section consists of steps performed on all the cluster nodes, regardless if the node is a primary node or a backup node.

  Repeat the steps in this section for each node of the cluster.

- External Application Server Host Configuration

  This section consists of steps performed on any host outside of the cluster that runs another instance of the SAP R/3 system.

  Repeat this section for each host that has an external application server installed.

Performing some of the iterations in parallel is fine, just use caution in any complex setup situation.

Rather than using the cut and paste mechanism you can also fill out the tables provided by first analyzing the primary host(s). Afterwards you can use the tables to synchronize all nodes.

Cluster Node Synchronization

Repeat the steps in this section for each node of the cluster that is different than the primary.

- Logon as root to the primary host.
- Prepare a logon for each of its backup hosts.

IS150 **Installation Step:**

Look at the groupfile file, `/etc/group`, on the primary side.

If any of the groups listed in Table 2-7 exist on the primary node and they do not exist on the backup node, copy them from the primary node to the backup node. If any group exists, verify that it has the same GID on both the primary and backup nodes. Merge the group members lists.

Table 2-7 Groupfile File Groups

groups	GID	group members
sapsys		
dba		
oper		
informix		
super_archive		
db<SID>adm		
db<SID>ctl		
db2asgrp		

IS160 **Installation Step:**

Look at the password file, `/etc/passwd`, on the primary side.

If any of the users listed in Table 2-8 exist on the primary node, recreate them on the backup node. Assign the users on the backup nodes the same user and group ID as the primary nodes.

INFORMIX users must have the same passwords, as well, on both the backup and primary nodes. Beware of copying over into `/etc/passwd` if your HP-UX is running in `Trusted System` mode.

Table 2-8 Password File Users

username	UID	GID	home directory	shell
<sid>adm				
ora<sid>				
informix				
sapr3				
db2as				
db2<SID>				

IS170 Installation Step:

Look at the service file, `/etc/services`, on the primary side.

Replicate all services listed in Table 2-9 that exist on the primary node onto the backup node.

Table 2-9 Service File Services

service name	service port
sapdp00	
sapdp01	
. . .	
sapdp<nn>	
. . .	
sapdp99	
sapdp<nn>s	

Table 2-9 Service File Services (Continued)

service name	service port
sapgw00	
sapgw01	
. . .	
sapgw<nn>	
. . .	
sapgw99	
sapgw<nn>s	
sapms<SID>	
orasrv	
tlisrv	
sapinf<SID>	
sapdb2<SID>	
sapdb2<SID>i	

IS180 **Installation Step:**

Change the HP-UX kernel on the backup node to meet the SAP WAS requirements.

Compare the Tunable Parameters section of `/stand/system` on both hosts. All values on the backup host must reach or exceed the values of the primary host. A tool, `kinst`, configures the kernel automatically. It is bundled with SAP WAS releases prior to 3.0F.

Install all HP-UX patches that are recommended for ServiceGuard and patches recommended for SAP WAS.

If you are planning to configure multiple Application Servers that are configured to be started parallel (see IS510) make sure the kernel parameter “tcp_conn_request_max” of your HP-UX systems are configured appropriately to 1024. This will prevent closing tcp ports (for example the LISTENER port configured for Oracle DB).

IS200 **Installation Step:**

Build a new kernel with `mk_kernel(1m)` on the backup host if `/stand/system` was changed in either Step IS180 or .

IS205 **Installation Step:**

If the setup is based on SAP basis 6.10 or higher, the startsap / stopsap alias has been removed from the environment of the <SID>adm user. Instead SAP ships shell scripts called startsap/stopsap in `/sapmnt/<SID>/exe`.

To create SGeSAP 3.x cluster environments for these releases, it is necessary to create the following scripts in `/home/<SID>adm directory`:

- `startsap_<hostname>_<INSTNUMBER>` containing the following line: `/sapmnt/<SID>/exe/startsap $1 <INSTANCENAME>`
- `stopsap_<hostname>_<INSTNUMBER>` containing the following line: `/sapmnt/<SID>/exe/stopsap $1 <INSTANCENAME>`

An example `<INSTANCENAME>` would be `DVEBMGS00` for a central instance with instance number 00.

IS210 **Installation Step:**

If the primary node has the Central Instance installed:

- Copy the `<sid>adm` home directory to the backup node. This is a local directory on each node.
- Rename the start, stop and environment scripts in `/home/<sid>adm` on the secondary node. Some of the environment scripts may not exist. For example:

```
su - <sid>adm
mv startsap_<primary>_<INSTNR>
startsap_<secondary>_<INSTNR>
mv stopsap_<primary>_<INSTNR> stopsap_<secondary>_<INSTNR>
mv .sapenv_<primary>.csh .sapenv_<secondary>.csh
mv .sapenv_<primary>.sh .sapenv_<secondary>.sh
mv .dbenv_<primary>.csh .dbenv_<secondary>.csh
mv .dbenv_<primary>.sh .dbenv_<secondary>.sh
# Remove logfiles from <primary> if any
exit
```

Never use the relocatable address in these filenames. If an application server was already installed, do not overwrite any files which will start the application server. If the rc-files have been modified, correct any hardcoded references to the primary hostname.

IR220 **Informix Database Step:**

If the primary node has the INFORMIX database installed:

- Copy the home directories of `sapr3` and `informix` to the backup node.
- Rename the environment scripts in `/home/informix` on the backup node. For example:

```
su - informix
mv .dbenv_<primary>.csh .dbenv_<secondary>.csh
mv .dbenv_<primary>.sh .dbenv_<secondary>.sh
exit
```

OR230 **Oracle Database Step:**

If the primary node has the ORACLE database installed:

Create additional links in /oracle/<SID> on the primary node. For example:

```
su - ora<sid>
ln .dbenv_<primary>.csh .dbenv_<secondary>.csh
ln .dbenv_<primary>.sh .dbenv_<secondary>.sh
exit
```

NOTE If you are implementing an APP server package make sure that you install the Oracle Client libraries on all nodes you run the package on. Refer to OSS Note 180430 for more details.

OR240 **Oracle Database Step:**

If you are using ORACLE:

Create a mountpoint for the Oracle files on the backup node if it is not already there. For example:

```
su - ora<sid>
mkdir -p /oracle/<SID>
exit
```

NOTE Complete this step also, if you are using 4.0A, the two package concept, and the backup node will be configured to run the Central Instance package only.

OS250 **Optional Step:**

If you need different configurations of the Central Instance depending on the node it runs on:

Be careful if you create different instance profiles for the different machines. You have to make sure by yourself that the Central Instance is capable of doing the intended work with any of the possible configurations.

Perform the following steps to setup different configurations:

- Execute the command:

  `su - <sid>adm`

- On each host the files `/home/<sid>adm/startsap_<local>_<INSTNR>` and `/home/<sid>adm/stopsap_<local>_<INSTNR>` contain a line that specifies the start profile. After a standard installation this line is similar to:

  `START_PROFILE="START_DVEBMGS<INSTNR>_<primary>"`

  Change the line individually on each host, MC/ServiceGuard in a two-node cluster:

  — On the primary host keep:

  `START_PROFILE="START_DVEBMGS<INSTNR>_<primary>"`

  — On the secondary host change the value in both files to:

  `START_PROFILE="START_DVEBMGS<INSTNR>_<secondary>"`

- The start profile and the instance profile can be found in the shared profile directory. Be careful if you rename the instance profile. You must change all references to this file afterwards. The following is an example of the steps to create two individual configurations in a two-node cluster:

```
cp /sapmnt/<SID>/profile/START_DVEBMGS<INSTNR>_<primary>\
/sapmnt/<SID>/profile/START_DVEBMGS<INSTNR>_<secondary>
cp /sapmnt/<SID>/profile/<SID>_DVEBMGS<INSTNR>_<primary>\
/sapmnt/<SID>/profile/<SID>_DVEBMGS<INSTNR>_<secondary>
```

- Now replace any reference to `<SID>_DVEBMGS<INSTNR>_<primary>`, type:

  `<SID>_DVEBMGS<INSTNR>_<secondary> in the file /sapmnt/<SID>/profile/START_DVEBMGS<INSTNR>_<secondary>.`

IS260 **Installation Step:**

If the primary node has the Central Instance installed and the other node has no internal application server installed:

Distribute the local directory tree /usr/sap/<SID>/SYS. Do not use rcp(1), it will follow all links and copy a lot of files from the shared disks that are not needed.

For example:

On the primary node:

```
cd /usr/sap/<SID>/SYS
find . -depth -print | cpio -o >/tmp/SYS.cpio
use ftp(1) to copy the file over to the secondary node
```

On the secondary node:

```
su - <sid>adm
mkdir -p /usr/sap/<SID>/SYS
cd /usr/sap/<SID>/SYS
cpio -id </tmp/SYS.cpio
exit
```

IS270 **Installation Step:**

Import the shared volume groups using the minor numbers specified in Step IS040.

The whole volume group distribution should be done using the command line interface. Do not use SAM. SAM will scramble the minor numbers. Specify the device minor numbers explicitly by creating the groupfile manually. For example:

```
mkdir /dev/vgsap<SID>
mknod /dev/vgsap<SID>/group c 64 0x080000
```

Now you can use vgimport(1m) with the mapfile created on the primary host during Step IS050 or Step IS090.

IS275 **Installation Step:**

Create a mountpoint for the Central Instance directory so the node can run the Central Instance. For example:

```
mkdir -p /usr/sap/<SID>/DVEBMGS<INSTNR>
```

Cluster Node Configuration

Repeat the steps in this section for each node of the cluster.

Logon as root.

IS280 **Installation Step:**

Check that `/etc/logingroup` is a link to `/etc/group`.

IS290 **Installation Step:**

Create an `.rhosts` file in the home directories of the HP-UX users `root`, `<sid>adm` and (if applicable) `informix`. Allow login for `root` as `root` from all nodes including the node you are logged into. Allow login for `root` and `<sid>adm` as `<sid>adm` from all nodes including the node you are logged into. Be careful with this step, many problems result from an incorrect setup of remote access.

Check the setup with `remsh` commands. If you have to provide a password, the `.rhosts` does not work.

IS300 **Installation Step:**

Create all directories below `/export` as specified in Step IS040. For example:

```
su - <sid>adm
mkdir -p /export/sapmnt/<SID>
mkdir -p /export/usr/sap/trans
exit
su - informix
mkdir -p /export/informix (Informix only)
exit
```

IS310 **Installation Step:**

Add all relocatable IP address information to `/etc/hosts`. Do not forget heartbeat IP addresses. Use Table 2-10 to record the addresses.

Table 2-10 **Relocatable IP Address Information**

name/aliases	IP address
	. . .
	. . .
	. . .

Table 2-10 Relocatable IP Address Information (Continued)

name/aliases	IP address
	. . .
	. . .
	. . .
	. . .
	. . .
	. . .
	. . .

IS320 **Installation Step:**

If you use DNS:

Configure `/etc/nsswitch.conf` to avoid problems. For example:

```
hosts: files[NOTFOUND=continue UNAVAIL=continue \
TRYAGAIN=continue]dns
```

IS330 **Installation Step:**

If you establish frontend and server LANs to separate network traffic:

Add routing entries to the internet routing configurations of `/etc/rc.config.d/netconf`. This is the only phase of the whole installation in which you will need to specify addresses of the server LAN. Route all relocatable client LAN addresses to the local server LAN addresses. For example, a two package concept:

```
ROUTE_DESTINATION[n]=<relocdb>
ROUTE_MASK[n]=""
ROUTE_GATEWAY[n]=<relocdb_s>
ROUTE_COUNT[n]=1
ROUTE_ARGS[n]=""
ROUTE_DESTINATION[n+1]=<relocci>
ROUTE_MASK[n+1]=""
ROUTE_GATEWAY[n+1]=<relocci_s>
ROUTE_COUNT[n+1]=1
ROUTE_ARGS[n+1]=""
```

IS335 **Installation Step:**

If you have SAP WAS 4.0A or higher:

Beginning with SAP WAS 4.0A, during installation SAP WAS appends some entries to the standard `.profile` files in the user home directories instead of using a new file defined by SAP WAS. On HP-UX, by default, there is the following in the given profiles:

```
set -u
```

This confuses the `.dbenv*.sh` and `.sapenv*.sh` files of SAP WAS. They fail during execution if the environment is not setup properly. Using SGeSAP the package startup fails trying to bring up the database.

Search the `.profile` of `<sid>adm` and remove the `set -u`, if found.

External Application Server Host Configuration

Repeat the steps in this section for each host that has an external application server installed.

Logon as root.

IS340 **Installation Step:**

Create an `.rhosts` file in the home directories of the HP-UX users `root`, `<sid>adm` and (if applicable) `informix`. Allow login for users `root` and `<sid>adm` from all cluster nodes.

For reasons described in Step IS335, search `.profile` in the home directory of `<sid>adm` and remove the `set -u`, if found.

IS345 **Installation Step:**

If you have SAP WAS 4.0A or higher:

For reasons described in Step IS335, search `.profile` in the home directory of `<sid>adm` and remove the `set -u`, if found.

IS350 **Installation Step:**

Add all relocatable IP address information to `/etc/hosts`.

IS360 **Installation Step:**

If you establish frontend and server LANs to separate network traffic:

Add routing entries to the internet routing configurations of `/etc/rc.config.d/netconf`. This is the only phase of the whole installation in which you will need to specify addresses of the server LAN. Route all relocatable client LAN addresses to the local server LAN addresses. For example, a one package concept:

```
ROUTE_DESTINATION[n]=<relocdbci>
ROUTE_MASK[n]=""
ROUTE_GATEWAY[n]=<relocdbci_s>
ROUTE_COUNT[n]=1
ROUTE_ARGS[n]=""
```

SGeSAP Files Configuration

The ServiceGuard Extension for SAP R/3 (SGeSAP) integration needs information about the specific setup at the customer site. It gathers this information from a file that is called `sap.conf`. You have to modify this file manually. Refer to the example provided with the integration files. It can be used as a template.

Logon to the primary host as root.

IS370 **Installation Step:**

Install the product depot file SGeSAP B7885BA using the `swinstall` tool.

B7885BA depends on B5140BA, the NFS toolkit which will also be installed if it is not already available. ServiceGuard 11.13 introduced a new concept of integrating various toolkits within standard packages. For more information on the latest reference to the HA-NFS toolkit refer to `http://docs.hp.com`.

IS375 **Installation Step:**

The swinstall process copied relevant files to `/opt/cmcluster` for reference. For your installation to work, copy the following files to the runtime directory:

```
cp /opt/cmcluster/sap/SID/sap.conf
/etc/cmcluster/<SID>/sap.conf
cp /opt/cmcluster/sap/sap.functions
/etc/cmcluster/sap.functions
cp /opt/cmcluster/sap/customer.functions
/etc/cmcluster/customer.functions
```

IS377 **Installation Step:**

In addition, you need the following package control and configuration files. These files should always be created using the SG 11.13 commands:

 # cmmakepkg

The `hanfs.sh` control file should be copied from the HA-NFS repository in `/opt/cmcluster/nfs` to the appropriate runtime directory of the SGeSAP package `/etc/cmcluster/<SID>`.

For the one package installation you need:

```
cp /opt/cmcluster/sap/SID/sapdbci.cntl
/etc/cmcluster/<SID>/sapdbci.cntl
cmmakepkg -s /etc/cmcluster/<SID>/dbci.cntl
cmmakepkg -p /etc/cmcluster/<SID>/dbci.conf
cp /opt/cmcluster/nfs/hanfs.sh /etc/cmcluster/<SID>/hanfs.dbci
```

For the two package installation you need:

```
cp /opt/cmcluster/sap/SID/sapdb.cntl
/etc/cmcluster/<SID>/sapdb.cntl
cp /opt/cmcluster/sap/SID/sapci.cntl
/etc/cmcluster/<SID>/sapci.cntl
cmmakepkg -s /etc/cmcluster/<SID>/db.cntl
cmmakepkg -s /etc/cmcluster/<SID>/ci.cntl
cmmakepkg -p /etc/cmcluster/<SID>/db.conf
cmmakepkg -p /etc/cmcluster/<SID>/ci.conf
cp /opt/cmcluster/nfs/hanfs.sh /etc/cmcluster/<SID>/hanfs.db
```

For the a application server package installation you need:

```
cp /opt/cmcluster/sapap.cntl /etc/cmcluster/<SID>/sapap.cntl
```

```
cmmakepkg -s /etc/cmcluster/<SID>/ap<SID><ASNR>.cntl
```

```
cmmakepkg -p /etc/cmcluster/ap<SID><ASNR>.conf
```

IS380 **Installation Step:**

Customize the package control and configuration files as described in the standard ServiceGuard manual *Managing ServiceGuard* (Part Number B3936-90026) and *Managing Highly Available NFS* (Part Number B5125-90001). Refer to http://docs.hp.com for the latest edition of this document.

IS385 **Installation Step:**

Customization of `sap.conf`.

The customization of `sap.conf` is divided into subsections as follows:

- Standard Parameters in sap.conf
- Additional Parameters in sap.conf
- Using Database Reconnect and Transaction Reset
- Advanced Options in sap.conf

Standard Parameters in sap.conf

Open the file `/etc/cmcluster/<SID>/sap.conf` with a text editor.

IS390 **Installation Step:**

SGeSAP performs activities specific to the database you use. Specify the underlying database vendor using the DB parameter. The compatible options are: ORACLE DB2 or INFORMIX.

IS400 **Installation Step:**

Provide information about the Central Instance that will be protected by a ServiceGuard package. Set the parameter CINR to the *Instance ID* of your Central Instance.

IS410 **Installation Step:**

If you are using the two package concept you can use the DB_RECONNECT functionality that is provided by SAP WAS.

If you want to use DB_RECONNECT with any instance of your SAP R/3 system, you *must* disable the automatic restart of the Central Instance, type:

```
CIRESTART=0
```

Setting this prevents the Central Instance from restarting in the event of a database failover. This means that the Central Instance profiles must be configured to use DB_RECONNECT. Remember that if you want to use DB_RECONNECT with any application server you must use it with the Central Instance, too. A reconnection of the application server to the switched database is only possible if the Central Instance has not been restarted.

The CIRESTART parameter has no meaning with a Central System that is secured by the one package concept and should not be modified. A failure of the database with a one package concept always means that the Central Instance has failed. The Central Instance is always restarted and reconnection of application servers is not allowed. The same conditions apply if you plan to use a two package approach with both packages on the same node all the time. When configuring the DBRECONNECT mechanism follow the appropriate OSS notes 109036, 98051 and 24806

IS420 **Installation Step:**

The file sap.conf contains four arrays that describe the additional application servers of your system.

Additional application servers are all application servers that belong to the SAP R/3 system and are different from the Central Instance. Each array has one entry for each application server. This is true whether the application server is running on a node in the cluster or outside of the cluster.

- In the ASHOST[*] array, specify the hostnames on which the application servers reside. Never use a relocatable name, even if the host is part of the cluster!
- Specify the Instance ID for each application server using the ASNR[*] array.

  You already specified the Instance ID of the Central Instance with the parameter CINR.

  Each index value *n*, ASNR[*n*] refers to the same application server as ASHOST[*n*]. If the corresponding ASHOST entry specifies a host that is part of the cluster, provide an ID that is different from the ID used by the Central Instance.

  Make sure that there is no other SAP WAS Instance on the same host using the same ID.
- In the third array called ASTREAT[*], you can define the way the application server acts if the status of the package changes. ASTREAT[*]=0 means that the application server is not affected by any changes that happen to the package status.
  - Add 1 to ASTREAT[*] if the application server should be started automatically during startup of a package.
  - Add 2 to ASTREAT[*] if the application server should be stopped automatically after initiating cmhaltpkg for ci-pkg.
  - Add 4 to ASTREAT[*] if the application server should be restarted automatically if a DB-switchover caused by a failure takes place. If you do not use the restart option you have to configure the instance to use DB-RECONNECT.
  - In the fourth array called ASPLATFORM[*] you specify the platform on which the Application Server runs. Supported values are:
    - "HP-UX": standard SAP Application server running on an HP-UX server

— "LINUX": standard SAP Application server running on a linux server

— "NT": standard SAP Application server running on a NT server. The NT Application server handling is not standardized as there is no way to open a remote DOS shell that starts R/3 Application servers on a windows platform. SGeSAP right now contains examples of functions using the ATAMAN ™ TCP Remote Logon.

— "SG-PACKAGE": ServiceGuard packaged SAP Application server. Specify this value is you want to run the Application Server within a ServiceGuard Cluster package. Refer to IS 377 for prerequisites.

IS430 **Installation Step:**

Specify the relocatable hostnames of the database and the Central Instance in the parameters DBRELOC and CIRELOC.

They will be the same if you use the one package concept.

Specify the relocatable hostname in TRANSRELOC, the host from which the mount to the transport directory /usr/sap/trans is initiated. In a multiple SAP R/3 system environment, specify a separate relocatable hostname for TRANSRELOC. In single system environments TRANSRELOC can be set equal DBRELOC.

OS435 **Installation Step:**

Specify AS_PSTART=0 if you want the Application Server startup to run sequentially. The default value here is AS_PSTART=1 for parallel startup. Setting AS_PSTART=0 will slow down your total failover time.

OS450 **Optional Step:**

Specify SAPOSCOL_STOP=1 if saposcol should be stopped together with each instance that is stopped.

SGeSAP makes sure that the collector only stops if there is no instance of an SAP R/3 system running on the host.

OS460 **Optional Step:**

It is possible to failover both packages of the two package concept to the same node.

If both packages try to come up on a single node after a failover at the same time, it is likely that the Central Instance package wants to start up the Central Instance before the database is fully recovered. This would lead to a failure because SAP WAS cannot connect to the database.

To deal with this situation, there is a loop implemented that polls the database in increasing intervals of time. The startup of the CI package is delayed until the database is reached. After the first poll, the script waits 10 seconds before initiating another one. After the second poll, the waiting time is increased to 20 seconds, and so forth. This continues until the database responds or up to a maximum of `DELAY_INTERVALS` polling attempts. You can modify `DELAY_INTERVALS` if you expect long recovery times.

Additional Parameters in sap.conf

Starting with release 3.0.03 additional parameters were introduced. You can add them on an *as needed* basis to already existing `sap.conf` files. The `sap.functions` file and `customer.functions` of release 3.0.03 or higher works with `sap.conf` files of previous releases without changing them.

OS470 **Optional Step:**

If your setup consists of application servers that are significantly slower than the Central Instance host, it is possible that the Central Instance shuts down before application server shutdown is completed. This can lead to unsafe shutdowns and Instance crash.

To be safe, specify one of the following:

- `WAIT_OWN_AS=1`

  the shutdown of all application servers takes place in parallel, but the scripts do not continue before all of these shutdown processes have come to an end.

- `WAIT_OWN_AS=2`

  if the package should also wait for all application servers to come up successfully. You have to use this value if you want to prevent the integration from temporarily opening a new process group for each application server during startup.

- `WAIT_OWN_AS=0`

can significantly speed up the package start and stop, especially if Windows NT application servers are used. Use this value only if you have carefully tested and verified that timing issues will not occur.

OS480 **Optional Step:**

Control the handling of resources.

On 32bit HP-UX, a shared memory shortage can occur if you install more than one instance on a host. Prior to any instance startup the SGeSAP tries to free up unused or unimportant resources to make the startup more likely to succeed. A database package only frees up database related resources, a Central Instance package only removes IPCs belonging to SAP WAS administrators. Table 2-11 summarizes how the behavior of SGeSAP is affected by changing the `CLEANUP_POLICY` parameter:

Table 2-11 Resource Handling and the `CLEANUP_POLICY` Parameter

CLEANUP_ POLICY =	CI resources	DB resources	AppServer resources
lazy	no action	no action	no action
normal	use cleanipc to remove unused own resources	unused ORACLE SGA is removed	use cleanipc to remove unused own resources
strict	remove any resource belonging to any SAP WAS-Instance	remove any resource belonging to any database	use cleanipc to remove unused own resources

- Integration versions prior to release 3.0.03 use the `lazy` policy: No additional cleanup takes place apart from SAP WAS standard shutdown activities.
- `cleanipc` is an SAP WAS tool used to free up the IPC resources of specific SAP WAS Instances. It is used to free up resources of an Instance that is to be started soon on HP-UX. This prevents `shmem`-problems due to a prior crash of the Instance. Accordingly, an obsolete `ORACLE SGA` is also removed if a database crash occurred.

- The `strict` policy uses HP-UX commands to free up all semaphores and shared memory segments that belong to *any* SAP WAS Instance of *any* SAP R/3 system on the host if the Central Instance is to be started soon. It uses HP-UX to free up all semaphores and shared memory segments that belong to *any* database if the SAP WAS database is to be started soon.

  Do not use the `strict` policy unless it is critical that you do. Be aware that the `strict` option can crash running instances of different SAP WAS systems on the backup host!

  — Use this value only if you have a productive system that is much more important than any other SAP R/3 system you have. In this case a switchover of the productive system is more robust, but additional SAP WAS systems will crash.

  — You can also use `strict` policy, if your SAP R/3 system is the only one running at the site and you are low on memory. `Strict` policy frees up more of its own shared memory segments than the normal policy does.

OS485 **Optional Step:**

`SERVER_CONSOLIDATION` is used to identify the usage of a consolidated system environment that has multiple SAP WAS systems on the same node.

To enable the handling of shared memory resources and application servers in a consolidated environment, specify:

```
SERVER_CONSOLIDATION=1
```

To specify the Dialog instances handling refer to Step OS510.

Using Database Reconnect and Transaction Reset

Starting with V3.0 of the SGeSAP Integration Scripts you can configure the DB-Reconnect functionality. DB-Reconnect is a feature provided by SAP WAS. It allows SAP WAS to remain running if the database switches and the Central Instance stays up. DB-Reconnect can be used with the two package concept. Transaction Reset can be used with either the one package or two package concept. Both DB-Reconnect and Transaction Reset can be combined in the two package concept.

SAP WAS V4.0B kernel patch level 85 introduced the Transaction Reset functionality. It allows additional SAP WAS application servers to remain running if the Central Instance switches. The first implementation of Transaction Reset was unreliable and caused problems with some SAP WAS transactions. Use at least kernel patch level 269.

The DB-Reconnect can be different for each application server. If you want to use DB-Reconnect for any application server, the Central Instance must also be configured to reconnect. Make sure that the data remains consistent under all conditions. If the Central Instance fails, all application servers are always restarted.

OS490 **Optional Step:**

If you want to use the DB-Reconnect functionality:

Add entries to:

- `/sapmnt/<SID>/profile/<SID>_DVEBMGS<INSTANCENR>`
- Instance Profiles of all application servers that use DB-Reconnect

When configuring the DBRECONNECT feature follow the appropriate OSS notes 109036, 98051 and 24806.

For example:

```
rsdb/reco_trials = 15
rsdb/reco_sleep_time = 60
rsdb/reco_sosw_for_db = off (based on OSS #109036)
rsdb/reco_sync_all_server = on
```

OS500 **Optional Step:**

If you want to use the DB-Reconnect functionality:

Make sure that you configured `CIRESTART=0` as described in Step IS410 the default setting of `CIRESTART` is `0` starting with SGeSAP Release 3.01.

OS510 **Optional Step:**

If you want to use the DB-Reconnect functionality:

Use `ASTREAT[]` values that are <4 for any application server that uses DB-Reconnect.

The `ASTREAT[]` array defines how the application servers are treated during switchover.

ASHOST[] and ASTREAT[] values with the same index belong to the same application server:

ASTREAT [*X*] = *Y*

where —

X specifies the corresponding application server.

Y is an integer value that defines the way the application server is treated.

Y = 0 means that the application server is not affected by any changes that happen to the package status. At the moment there are three triggers you can use to customize the way the scripts affect the application server. They are:

- ACTIVE—Add 1 to Y if the application server should be started automatically during the startup of a package.

NOTE Not during failover – only during package startup.

Under normal circumstances all application servers should be configured to be active.

Under special conditions, for example, ServiceGuard maintenance, you can stop an application server manually. In situations such as these, also configure the application server to be inactive. For now you need to change the scripts manually.

To deactivate an application server, you have to decrease his ASTREAT[] value by one in the sap.conf files on all nodes. This ensures the number is even. Otherwise a switchover would accidentally cause an unwanted startup of the previously stopped application server.

To reactivate the application server Instance, increase the ASTREAT[] value on all nodes.

If you deactivate an application server by changing the sap.conf files, the application server Instance does *not* stop automatically or immediately. When you deactivate this way, the Instance is not (re-)started automatically by ServiceGuard. You can still have manual startups and ServiceGuard-triggered shutdowns of the Instance.

- FINAL STOP—Add 2 to Y if the application server should automatically be stopped after initiating `cmhaltpkg`.
- RESTART—Add 4 to Y if the Application Instance processes should be stopped and restarted automatically if a DB-switchover caused by a failure takes place.

  Use this for extremely critical environments, in which safety is the most important concern. The restart does not take place if the application server is configured to be inactive.

Table 2-12 provides a summary of values for Y:

Table 2-12 Y Values for ASTREAT[]

Value	Action
0	This application server configured is never touched by the Extension Scripts. *Note*: This means the application server has to be configured for DB-RECONNECT.
1	This application server is an active server. This means the scripts try to start the application server if the package is coming up. *Note*: Configure the application server to use DB-RECONNECT.
2	This application server stops if the ci-package halts. All startups have to be done manually. *Note*: This means the application server has to be configured for DB-RECONNECT.
3	This application server always starts if the package is coming up. It stops if the package is halted manually. In case of a failover the application server reconnects by itself. Use this value if you want to use RECONNECT and additionally want to control the SAP R/3 system as a whole by using the package commands. *Note*: This means the application server has to be configured to use DB-RECONNECT.
4,5 or 6	Do not configure.
7	This application server stops and starts in case of a switchover (no DB-RECONNECT). It stops if the ci-package halts. It always starts if the package is coming up.

The Central Instance is treated the same as any of the additional application servers. Use CIHOST and CIRESTART instead of ASHOST[] and ASTREAT[].

Table 2-13 provides an overview of the possible configurations and the corresponding actions.

Table 2-13 ASTREAT Configuration Options

ASTREAT	Restart	Final_ Stop	Active	Runpkg DB	Haltpkg DB	Runpkg CI	Haltpkg CI
0	0	0	0	nop	nop	nop	nop
1	0	0	1	CI-start	nopstart	start	nop
2	0	1	0	nop	nop	nop	stop
3	0	1	1	CI-start	nop	start	stop
4	1	0	0	stop	nop	stop	nop
5	1	0	1	stop CI-start	nop	stop start	nop
6	1	1	0	stop	nop	stop	stop
7	1	1	1	stop CI-start	nop	stop start	stop
nop = no operation st_if_nl = start if *not* on local host							

The Central Instance should always be ACTIVE and configured for FINAL_STOP.

The RECONNECT behavior can be specified.

Advanced Options in sap.conf

In /etc/cmcluster there is a file called customer.functions. Do not change the sap.functions. The customer.functions templates that are delivered with SGeSAP work with additional parameters in the second part of sap.conf. Use the customer.functions as needed.

OS520 Optional Step:

SAPROUTER is a sample additional program that always starts on the Central Instance host.

To start an saprouter on the CI host automatically, specify:

```
SAPROUTER_START=1
```

You can also provide an option string that is passed to the `saprouter`, for example to set the path to the `saprouttab` file to reside on a shared volume.

Typically, set a maximum of one `saprouter` inside of the cluster. `saprouters` cannot share a service port. So make sure that you use different ones, if a failover to the same node is possible for the saprouters.

OS530 **Optional Step:**

Sometimes, when a failover occurs, you want to stop other SAP WAS Instances running on backup nodes.

This is useful if you have to free up limited resources before you are able to start the failed Instance again. You have to configure two groups of arrays to use this functionality:

- The first group consists of the arrays `RMNR[*]`, `RMADM[*]` and `RMDEP[*]`.
  - — Specify the Instance-IDs and the name of the System Administrators of each instance that shall be stopped in the arrays `RMNR[*]` and `RMADM[*]`.
  - — For Dialog-Instances and other additional application servers also specify:

    ```
    RMDEP[*]=-1
    ```

    This means, that there are no instances which depend on the services offered by them.
  - — For Central Instances, specify the index into the second group of arrays in `RMDEP[*]`.

    If you allow stopping of Central Instances, you have to specify the list of related additional application servers, too. The application servers are stopped first. It does not matter on which host they are running.
- The second group of arrays consists of `RMDEPNR[*]` and `RMDEPHOST[*]`.

  Specify the application servers depending on the Central Instances which are allowed to be stopped. Fill in the Instance-IDs and the local hostnames on which the application servers run. The list of

application servers which belong to the same SAP R/3 system start at the index which is specified in the RMDEP[*] entry of their Central Instance and the list continues until an RMDEPNR[*]=-1 is found. Never forget this entry.

The RMNR[*] specified instances are halted only if they are running on the host you currently fail over to. This functionality never halts any ServiceGuard package. It is possible to stop a Central Instance that is using ServiceGuard packages, but the associated package remains in the running state. Application servers belonging to the same SAP R/3 system as the failing package do not need to be specified here. They stop automatically.

OS540 **Optional Step:**

If there is a special demand to use values different from the default, it is possible to redefine some of the following values:

- name of SAP WAS System Administration User
- home directory of SIDADM
- name of the database package
- name of the Central Instance package
- name of Oracle Administrator
- SID of Oracle Database
- database home directory
- path of the startdb logfile
- path of the stopdb logfile
- path of the SAP WAS default profile
- path of the SAP WAS global transport directory
- name of the database listener

OS550 **Optional Step:**

If you want to use a database different from ORACLE or INFORMIX:

Currently only ORACLE and INFORMIX databases are supported, but there is a mechanism to easily integrate another database, provided its handling is not significantly different.

Specify a database vendor different from ORACLE and INFORMIX in the DB parameter of sap.conf: DB=<VENDOR>

SGeSAP Integration calls the script functions:

`start_<VENDOR>_db, stop_<VENDOR>_db, and stop_<VENDOR>_as`

These functions start and stop the database as well as stop an application server gracefully. The third function is needed, in case there are specific tasks required to stop an application server after the underlying database has failed. These functions must be specified in the database dependent part of the `customer.functions` file.

Configuring an Application Server Package with SGeSAP

The configuration of an application server package introduces some advantages for SAP application servers. As described in Chapter 1, an application server package has advantages within a 2 node cluster configuration with symmetric nodes or when protecting dedicated batch servers within a cluster. The following intallation step describes the generic setup of an application server package.

OS551 **Optional Step:**

The APP-package should be called ap<SID><INSTNR> according to the conventions shown in the following table.

Table 2-14 Application Server Package Setup

SG package	VG Name	lvol Names	Later Mount Point	Device minor number
ap<SID><INSTNR>	vg<SID><INSTNR>	lvsap<SID>	/usr/sap/<SID>/D<INSTNR	11

- Configure a volume group for each application server package according to step IS040:
- The use of local executables for the application server package is required. Configure local executables according to Installation Step IS080.
- Synchronize all cluster nodes according to step IS150 to IS210.
- For each application server package, configure .conf and .cntl files according to Installation Step IS377:

cmmakepkg -p /etc/cmcluster/<SID>/ap<SID><INSTNR>.conf

cmmakepkg -s /etc/cmcluster/<SID>/ap<SID><INSTNR>.cntl

- The application server package must be named according to the following convention:

ap<SID><INSTNR>

- Copy the runtime control logic file from /opt/cmcluster/SID/sapap.cntl to /etc/cmcluster/<SID>:

cp /opt/cmcluster/SID/sapap.cntl /etc/cmcluster/<SID>

See IS377.

- Configure DBRECONNECT for each application server that will be configured within a package according to Installation Step IS410.
- Configure each application server within the ASPLATFORM[@] array to "SG-PACKAGE" according to IS420.
- For each application server package, configure ASTREAT[@] as follows:

ASTREAT[@]=2: halt the application server package after switching of ci- or dbci-package (in ap<SID><INSTNR>.conf set AUTO_RUN to NO)

ASTREAT[@]=1: start the application server package when starting the dbci- or ci-package (in ap<SID><INSTNR>.conf set AUTO_RUN to NO)

ASTREAT[@]=0: start/halting/switching is controlled by the cluster management, the dbci- or ci-package related to the application server package is not controlling the application server package (in ap<SID><INSTNR>.conf set AUTO_RUN to YES).

- Adjust all settings according to steps IS610, IS630, IS650, IS659, IS660 and IS710 in sequence.
- Create a “debug” file in /etc/cmcluster and start the application server package on all nodes where it should run.
- Complete the SAP specific settings according to step IS920 to IS940.

Your application server package has now been configured completely. Remove the “debug” file and restart the package to see whether the SAP application server comes up correctly on all nodes configured.

ServiceGuard Configuration

Logon as root on the primary host.

OS560 **Optional Step:**

Recommendation

Set `AUTO_VG_ACTIVATE=0` in `/etc/lvmrc`. Edit the `custom_vg_activation()` function if needed. Distribute the file to all cluster nodes.

OS570 **Optional Step:**

Recommendation

Set `AUTOSTART_CMCLD=1` in `/etc/rc.config.d/cmcluster`. Distribute the file to all cluster nodes.

IS580 **Installation Step:**

Create cluster configuration file in `/etc/cmcluster`:

```
cmquerycl -v -C  /etc/cmcluster/sapr3.conf [-n <nodename> ...].
```

IS590 **Installation Step:**

Customize the cluster configuration file. Specify the volume groups listed in Step IS040. You might need to increase the timing parameters.

IS600 **Installation Step:**

Obsolete.

IS610 **Installation Step:**

Specify package name, node names and subnets in the package configuration files `db.conf` and `ci.conf` or in `dbci.conf`.

IS620 **Installation Step:**

If you implement the one package concept:

The standard package name is dbci<SID>. If you want to use different variables follow OS530.

If you implement the two package concept:

The standard package names are db<SID> and ci<SID>. If you want to use different names here, make sure that you also specified the appropriate variables in Step OS530.

IS630 **Installation Step:**

Specify the package control script names in the package configuration files:

For one package concept specify in /etc/cmcluster/<SID>/dbci.conf:

```
RUN_SCRIPT /etc/cmcluster/<SID>/dbci.cntl
HALT_SCRIPT /etc/cmcluster/<SID>/dbci.cntl \
```

For two package concept specify in /etc/cmcluster/<SID>/db.conf:

```
RUN_SCRIPT /etc/cmcluster/<SID>/db.cntl
HALT_SCRIP /etc/cmcluster/<SID>/db.cntl
```

For two package concept specify in /etc/cmcluster/<SID>/ci.conf:

```
RUN_SCRIPT /etc/cmcluster/<SID>/ci.cntl
HALT_SCRIPT /etc/cmcluster/<SID>/ci.cntl
```

For each Application Server package specify in /etc/cmcluster/<SID>/ap<SID><ASNR>.conf:

```
RUN_SCRIPT /etc/cmcluster/PRC/ap<SID><ASNR>.cntl
HALT_SCRIPT /etc/cmcluster/PRC/ap<SID><ASNR>.cntl
```

IS650 **Installation Step:**

Follow the directions in Steps IS040 and IS310 to define volume groups, logical volumes, IP addresses and subnets in the package control scripts dbci.cntl, db.cntl, ci.cntl and ap<SID><ASNR>.cntl as appropriate.

Fill the IP[*] and SUBNET[*] array with the IP addresses and subnet addresses that the package is attached to. For example a 1-pkg control script dbci.cntl:

```
IP[0]="15.27.218.238"
SUBNET[0]="15.27.216.0"
```

The filesystems you specify in the LV[*], FS[*] and FS_MOUNT_OPT[*] array are not identical to the filesystems that are exported. For example:

```
LV[0]="/dev/vgDBPRD/lvoracle"; FS[0]="/oracle/PRD";
FS_MOUNT_OPT[0]="-o rw"
LV[1]="/dev/vgDBPRD/lvora805"; FS[1]="/oracle/805_64";
FS_MOUNT_OPT[1]="-o rw"
LV[2]="/dev/vgDBPRD/lvsapmnt"; FS[2]="/export/sapmnt/PRD";
FS_MOUNT_OPT[2]="-o rw"
LV[3]="/dev/vgPRC/lvusrsap"; FS[3]="/usr/sap/PRD/DVEBMGS20";
FS_MOUNT_OPT[3]="-o rw"
LV[4]="/dev/vgTRANS/lvtrans"; FS[4]="/export/usr/sap/trans";
FS_MOUNT_OPT[4]="-o rw"
```

IS655 **Installation Step:**

The follwing steps will customize the hanfs.<xxx> scripts. It will customize all required directories for the usage of the HA-NFS. All directories that are handled by the automounter must be exported by the scripts if they are part of the packages, for example:

`/export/sapmnt/<SID>`

`/export/usr/sap/trans`

Exported directories can usually be found beneath the special export directory `/export`. The directories to be exported are specified including their export options, using the `XFS[*]` array the hanfs.<xxx> script out of the HA-NFS toolkit. This script is called within the runtime by the standard ServiceGuard contol script `dbci.cntl`, `db.cntl`.

The transport directory is also part of the package in standard installations. In those cases it also has to be mentioned here. On INFORMIX installations without local database executables, access must be granted to the database filesystem too.

Only allow access from machines inside of the cluster that are configured as backup hosts and additional application servers of this specific SAP R/3 system. Make sure to allow access for all addresses that the servers use so they can reach the directories. Allowing the right addresses is essential if your hosts use more than one LAN card.

Example of an `hanfs.db` file:

```
# XFS[0]="-o root=hpcc006:hpcc008:hpcc071:trans:db:sap
/export/usr/sap/trans"

# XFS[1]="-o root=hpcc006:hpcc008:hpcc071:trans:db:sap
/export/sapmnt/<SID>"
```

```
# XFS[2]="-o root=hpcc006:hpcc008:hpcc071:trans:db:sap
/export/informix/<SID>"
```

IS659 **Installation Step:**

To enable parameter export of variables that are set within the package control script add the following line "`set -a`" to the top of each ServiceGuard control script, specifically at the top line within each package control file:

`/etc/cmcluster/<SID>/dbci.cntl`
/etc/cmcluster/<SID>/db.cntl
`/etc/cmcluster/<SID>/ci.cntl`
`/etc/cmcluster/<SID>/ap<SID><NR>.cntl`

Example:

```
# "A.11.13 $Date: 02/12/2001 $".
# HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)
# Note: This file MUST be edited before it can be used.
# set -a
# UNCOMMENT the variables as you set them.
```

IS660 **Installation Step:**

To enable the SAP WAS specific scripts change the `customer_defined_commands` sections of the package control script(s):

For the one package concept:

```
function customer_defined_run_cmds
{
/etc/cmcluster/<SID>/sapdbci.cntl startDBCI <SID>
test_return 51
}
function customer_defined_halt_cmds
{
/etc/cmcluster/<SID>/sapdbci.cntl stopDBCI <SID>
test_return 52
}
```

For a two package concept:

Use the commands `stopDB`, `startDB` in file `db.cntl` or `stopCI`, `startCI` in `ci.cntl` respectively.

For an Application Server package in ap<SID><ASNR>.cntl:

```
{
function customer_defined_run_cmds
# ADD customer defined run commands.
/etc/cmcluster/<SID>/sapap.cntl startAP <ASNR> <SID> <SIDADM>
test_return 51
}
function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
/etc/cmcluster/<SID>/sapap.cntl stopAP <ASNR> <SID> <SIDADM>
test_return 52
}
```

OR680 **Oracle Database Step:**

If you use ORACLE database:

Remove INFORMIX specific files from the cluster directory:

```
rm /etc/cmcluster/customer.sh /etc/cmcluster/customer.csh
```

IR690 **Informix Database Step:**

If you use INFORMIX database:

Move INFORMIX specific files into the INFORMIX home directory. Read the command example carefully. Make sure you move the 'dot files'. For example:

```
mv /etc/cmcluster/customer.sh /home/informix/.customer.sh
mv /etc/cmcluster/customer.csh /home/informix/.customer.csh
chown informix:informix /home/informix/.customer.sh
chown informix:informix /home/informix/.customer.csh
```

IS700 **Installation Step:**

Copy all integration files below `/etc/cmcluster` to the other cluster hosts using `rcp(1)`.

IS710 **Installation Step:**

Create the binary cluster configuration file and distribute it using `cmapplyconf(1m)`.

IS720 **Installation Step:**

If you plan to use an additional package for the transport directory:

Specify this package in the last position of the `cmapplyconf` command. Later, if you force a shutdown of the whole cluster with `cmhaltcl -f`, the package with the transport directory is the last one stopped. This prevents the transport directory from disappearing before all SAP WAS systems in the cluster have completed their shutdown.

Automounter/Auto FS Configuration

Enabling AutoFS on the HA NFS server needs to be done with caution. However, this kind of configuration is a fully supported configuration if one of these two actions are taken before restarting the AutoFS daemon or before performing any stopping or restarting NFS client (via "`nfs.client stop/start`") or NFS server (via "`nfs.server stop/start`"):

- Halt the NFS package(s) running on the server node first, perform the action needed (see above), and then run the NFS package(s) again.
- Manually move the NFS package(s) from primary server nodes to adoptive server node.

WARNING **The NFS-client or NFS-server must NOT be stopped/started a SG package with NFS exports is started!**

Repeat the steps in this section for each node of the cluster and for each external application server host.

Logon as root.

IS730 **Installation Step:**

Check that the Automounter is active. In `/etc/rc.config.d/nfsconf`, the section for the `autofs` configuration must look like:

```
AUTOMOUNT=1
AUTO_MASTER="/etc/auto_master"
AUTOMOUNT_OPTIONS="-f $AUTO_MASTER"
AUTOMOUNTD_OPTIONS=
AUTOFS=1
```

Older installations on HP-UX 10.x and installations without `autofs` require a slightly different syntax for the "old" automounter:

```
AUTOMOUNT=1
AUTO_MASTER="/etc/auto_master"
AUTO_OPTIONS="-f $AUTO_MASTER"
```

IS740 **Installation Step:**

Make sure that at least one NFS client daemon and one NFS server daemon is configured to run. This is required for the automounter to work. Check the listed variables in `/etc/rc.config.d/nfsconf`. They should be specified as greater or equal to one. For example:

```
NFS_CLIENT=1
NFS_SERVER=1
NUM_NFSD=4
NUM_NFSIOD=4
```

IS750 **Installation Step:**

Add the following line to your `/etc/auto_master` file:

```
/- /etc/auto.direct
```

IS760 **Installation Step:**

Create a file called `/etc/auto.direct`.

For each directory configured to be mounted below `/export` in Step IS040, add a line to this file. For example:

```
/usr/sap/trans <relocdbci_s>:/export/usr/sap/trans
/sapmnt/<SID> <relocdbci_s>:/export/sapmnt/<SID>
```

For INFORMIX databases, add another entry. For example:

```
/informix/<SID> <relocdbci_s>:/export/informix/<SID>
```

For the two package concept, typically the relocatable IP address of the database package `<relocdb_s>` is used.

When configuring AUTOFS the automounter map file /etc/auto.direct must NOT be executable. Make sure to set the appropriate permissions of /etc/auto.direct to 644.

IS770 **Installation Step:**

Restart the automounter with:

```
/sbin/init.d/nfs.client stop
```

and

```
/sbin/init.d/nfs.client start
```

NOTE

Never kill the automount process. Always use `nfs.client` to stop or start it.

Never stop the NFS client while the automounter directories are still in use by some processes. If `nfs.client stop` reports that some filesystems could not be unmounted, the automounter may refuse to handle them after `nfs.client start`.

After configuring all automounters by performing the above mentioned steps on all hosts, verify that the setup works correctly to this point. Do not continue with the following application dependent sections without doing this! Perform the tests using the debug-switch

OS780 **Optional Step:**

Create a file called /etc/cmcluster/debug on all nodes in the cluster. This enables you to start the ServiceGuard packages without running the SAP WAS specific steps.

Reboot all machines.

Start the cluster and run all packages. This might be done automatically. If everything is working correctly, the log-file(s) in /etc/cmcluster/<SID> will indicate successful package startup.

On the primary host(s) it should be possible to run the database and the Central Instance. Be sure to start the listener as ora<sid> before trying to bring up an ORACLE database. All application servers should work after starting them manually.

Now, that everything works stop all application servers. Stop the Central Instance and the database. Stop the listener if needed. Do not stop the packages yet.

Database Configuration

This section describes the following:

- Additional Steps for Informix
- Additional Steps for Oracle
- Additional Steps for DB2

Additional Steps for Informix

Logon as root to the primary host of the database where the package is running in debug mode.

IR790 **Informix Database Step:**

Perform the following steps as an INFORMIX user:

```
su - informix
```

Comment out the remsh sections in the files called .dbenv.csh and .dbenv.sh in the home directory. If they are missing, check for alternative files with hostnames in them:

```
.dbenv_<local>.csh and .dbenv_<local>.sh.
```

The remsh section looks similar to:

```
# remsh <local> date >& /dev/null
# if ( $status <= 0 ) then
#   echo Remote login check to dbserver <local> failed.
#   echo Check if <local> is alive and file ...
#   echo on <local> contains an entry for ...
# endif
```

Copy the two .dbenv files over to the INFORMIX home directory on all cluster nodes, all external application server hosts, and MC/ServiceGuard by using ftp.

If you do not do this, switching time increases dramatically; each remsh command to the original database host causes delay if the node is not available.

IR800 **Informix Database Step:**

In Step IR690 you copied two files to the INFORMIX home directory of the primary node. At this time, still as an INFORMIX user, customize these files by replacing the string `relocdb` with your individual `<relocdbci>` (or `<relocdb>` in case of two package concept). For example:

.customer.sh:

.customer.csh:

```
######################## .customer.sh ##########################

DBRELOC=relocdb; export DBRELOC
ONCONFIG=onconfig.relocdb.${INFORMIX_DBID}; export ONCONFIG

TEMP=`netstat -i | awk ' {print $4 }' | sort -u | \
  awk -F. '$1~/'$DBRELOC'/ {if (length($1) == length("'$DBRELOC'")) "\
 {print 1;exit}}'`


setenv INFORMIXSERVER ${DBRELOC}${INFORMIX_DBID}shm

##################.customer.csh################################
##

setenv DBRELOC relocdb

setenv ONCONFIG onconfig.${DBRELOC}.${INFORMIX_DBID}

setenv TEMP `netstat -i | awk ' {print $4 }' | \

 sort -u | awk -F. '$1~/'$DBRELOC'/ \

 {if (length ($1) == length("'$DBRELOC'")) {print 1;exit}}'`

if ( $TEMP == 1 ) then
```

Copy the `.customer.sh` and `.customer.csh` to all INFORMIX home directories on all nodes including the application servers outside of the cluster.

IR810 **Informix Database Step:**

Perform the following steps as `<sid>adm` user:

```
su - <sid>adm
```

Copy the files that were manipulated in Steps IS760 and IS770 to the home directory:

```
cp /home/informix/.dbenv* ~
cp /home/informix/.customer.* ~
```

Copy the files over to the home directories of <sid>adm on all cluster nodes and all external application server hosts.

IR820 **Informix Database Step:**

Perform the following steps as an INFORMIX user.

Modify the parameters DBSERVERNAME and DBSERVERALIAS in the ONCONFIG file:

/informix/<SID>/etc/onconfig.<primary>.<DBID>.

The parameter default to:

<local><INFORMIX_DBID><shm|tcp>

Set them to:

<relocdb><INFORMIX_DBID><shm|tcp>

For example:

```
DBSERVERNAME relocdb<sid>shm
DBSERVERALIASES relocdb<sid>tcp
```

IR830 **Informix Database Step:**

Rename the ONCONFIG file to:

/informix/<SID>/etc/onconfig.<relocdb>.<DBID>

IR840 **Informix Database Step:**

Add a line with rel-IP-name of the database package to the file:

/informix/<SID>/etc/sqlhosts.soc.

After SAP WAS installation this file will be similar to:

```
demo_on onipcshm on_hostname on_servername
demo_se seipcpip se_hostname sqlexec
<local><sid>shm onipcshm <local> sapinf<SID>
<local><sid>tcp onsoctcp <local> sapinf<SID>
```

Change the <local> entries to <relocdb> of the database package. For example:

```
<relocdb><sid>shm onipcshm <relocdb> sapinf<SID>
<relocdb><sid>tcp onsoctcp <relocdb> sapinf<SID>
```

IR845 **Informix Database Step:**

If you are using SAP WAS version 4.0x:

Create additional entries in the INSTVERS table of SAP WAS. Print out the content of the table. Note the ID column. This is the primary key of the table. All new entries need a unique key.

Review the STATUS column. There should be one row containing a 0 as status. This indicates the current, successful installation.

Copy this row using a unique key as well as secondary hostnames for HOSTNAME and DBHOSTNAME.

For the two package concept or more than two cluster nodes include entries for all possible combinations. For example, you need two additional entries for a two-package concept with two cluster nodes to reflect that the two packages can run on different nodes. Refer to Step OR915 for more information.

Additional Steps for Oracle

With the introduction of Oracle 8.i Enterprise a new 2-phase instance and crash recovery mechanism is available that enables a faster and predictable recovery time after a crash.

The instance and crash recovery is initiated automatically and consists of two phases:

Roll-forward phase: Oracle applies all committed and uncommitted changes in the redo log files to the affected datablocks. Following parameters can be used to tune the roll forward phase:

- The parameter RECOVERY_PARALLELISM controls the number of concurrent recovery processes.
- The parameter FAST_START_IO_TARGET controls the time a crash / instance recovery may take. Use this parameter to make crash / instance recovery predictable.

Roll-back phase: Oracle applies information in the rollback segments to undo changes made by uncommitted transactions to the data blocks. Following parameters can be used to tune the roll-back phase:

- Fast-Start On-Demand rollback: with this feature Oracle automatically allows new transactions to begin immediately after the roll forward phase of recovery completes. This means that the database will be available again right after the completion of phase one roll-forward. This means that there will be no long waits until long running transactions are rolled back.
- Fast-Start Parallel Rollback: configure the FAST_START_PARALLEL_ROLLBACK parameter to roll-back set of transaction in parallel. This parameter is similar to the RECOVERY_PARALLELISM parameter for the roll-forward phase.

All these parameters can be used to tune the duration of Instance / Crash recovery.

More details on these useful new High-Availability features can be found in the Oracle 8.1.7 documentation "Oracle8i Designing and Tuning for Performance" (Chapter 24: Tuning Instance recovery Performance).

The following steps have to be performed in order to adjust the Oracle DB setting to the HA configuration.

Logon as root to the primary host of the database where the package is running in debug mode.

OR850 **Oracle Database Step:**

Perform the following step as `<sid>adm`.

To ensure that a database that crashed during an online backup starts correctly after the crash, all datafiles that were in 'begin backup' state need to be altered with an "end backup' statement. Adjust the required steps in `/sapmnt/<SID>/exe/startdb`.

Therefore, 'startdb' needs to be adjusted accordingly. Insert / Change the following code within the '`/sapmnt/<SID>/exe/startdb`' file. The sample code can be found in the file:
`/opt/cmcluster/sap/SID/startdb.sql`

```
#
# Startup the database without changing the ARCHIVELOG state
#
echo "connect internal;" > $SRVMGRDBA_CMD_FILE
echo "startup;" >> $SRVMGRDBA_CMD_FILE
echo "exit;" >> $SRVMGRDBA_CMD_FILE
eval $SRVMGRDBA command=@$SRVMGRDBA_CMD_FILE >> $LOG 2>&1
```

```
#
# Startup the database without changing the ARCHIVELOG state
# alter datafile 'end backup' when instance crashed during
# backup
echo "connect internal;" > $SRVMGRDBA_CMD_FILE
echo "startup mount;" >> $SRVMGRDBA_CMD_FILE
echo "spool endbackup.log" >> $SRVMGRDBA_CMD_FILE
echo "select 'alter database datafile '''||f.name||''' end
backup;'" >> $SRVMGRDBA_CMD_FILE
echo "from v\$datafile f, v\$backup b" >> $SRVMGRDBA_CMD_FILE
echo "where b.file# = f.file# and b.status = 'ACTIVE'" >>
$SRVMGRDBA_CMD_FILE
echo "/" >> $SRVMGRDBA_CMD_FILE
echo "spool off" >> $SRVMGRDBA_CMD_FILE
echo "!grep '^alter' endbackup.log >endbackup.sql" >>
$SRVMGRDBA_CMD_FILE
echo "@endbackup.sql" >> $SRVMGRDBA_CMD_FILE
echo "!rm endbackup.*" >> $SRVMGRDBA_CMD_FILE
echo "alter database open;" >> $SRVMGRDBA_CMD_FILE
echo "exit;" >> $SRVMGRDBA_CMD_FILE eval $SRVMGRDBA
command=@$SRVMGRDBA_CMD_FILE >> $LOG 2>&1
```

OR860 **Oracle Database Step:**

If you are using SAP WAS version 3.x:

Perform the following steps as ora<sid>.

Configure the listener to listen on the relocatable name of the database package. To do this, change all references from <local> to the relocatable name <relocdb> in the files. For example:

```
/usr/sap/trans/listener.ora
/usr/sap/trans/tnsnames.ora
```

OR870 **Oracle Database Step:**

If you are using SAP WAS version 4.0A or higher:

Perform the following steps as ora<sid>.

Configure the listener to listen on the relocatable name of the database package. To do this, change all references from <local> to the relocatable name <relocdb> in the files on the shared volume group. Be careful if these files were customized after SAP WAS installation. For example:

```
$ORACLE_HOME/network/admin/listener.ora
$ORACLE_HOME/network/admin/tnsnames.ora
```

OR880 **Oracle Database Step:**

If you are using SAP WAS version 4.0A or higher:

Copy `$ORACLE_HOME/network/admin/tnsnames.ora` to all additional application server hosts. Be careful if these files were customized after SAP WAS installation.

OR885 **Oracle Database Step:**

If you are using SAP WAS version 4.0A or higher and you are using the two package concept:

As with all application servers, the Central Instance needs a subset of the files in $ORACLE_HOME to work properly. Verify if your Central Instance can run on any host without an application server installed. These hosts need a local copy of the filetree.

If you have access to the SAP WAS install disks you can get the files from the SAP WAS archive `ORACLI.CAR` or `OCL<oclrel>.CAR` on the kernel CD. The extractor program `CAR` functions the same as the `tar(1m)` command. It is shipped with SAP WAS.

If the Central Instance was installed on a node different from the database node, you can get the files there. Or you can use any additional application server host as source. Copy the filetree, for example, by using the procedure explained in Step IS260. The filetree should contain the following subdirectories: `lib`, `network ocommon`, and `rdbms`.

Do not copy `$ORACLE_HOME` from the database host or you will copy much more than you want. Refer to SAP WAS OSS Note 180430 for more details.

OR890 **Oracle Database Step:**

If you are using SAP WAS version 3.x:

If existing locally, copy `/etc/listener.ora` to all adoptive nodes in the cluster.

OS900 **Optional Step:**

If you are using SAP WAS version 3.x and plan to use more than one SAP R/3 system inside of your cluster:

It is possible that more than one database is running on the same node. Even though one listener is capable of serving many database instances problems can occur in switchover environments because needed filesystems may not be available at the startup time of the listener. Use a dedicated listener process for each database.

You can use the standard listener.ora file that is created during the installation of *one* SAP R/3 system <SID1> as a template. Double its contents. For example:

```
cat listener.ora listener.ora >listener.ora.new
mv listener.ora.new listener.ora
```

Now the file consists of two identical parts.

- In the first part of the file:

  Replace each occurrence of the word LISTENER by a new listener name. You can choose what suits your needs, but it is recommended to use the syntax LISTENER<SID1>:

  ```
  ( host = <relocdb_1> )
  ```

  Change nothing.

- In the second part of the file: Replace each occurrence of the word LISTENER by a new listener name different from the one chosen above. For example, use LISTENER<SID2> if <SID2> is the SID of the second SAP R/3 system. Replace any other occurrence of <SID1> by <SID2>. The line should be modified to contain the appropriate relocatable address belonging to the database package (db or dbci) of the second system. For example:

  ```
  ( host = <relocdb_2> )
  ```

  In the line:

  ```
  ( port = 1527 )
  ```

  a new previously unused port should be placed. For example:

  ```
  ( port = 1528 )
  ```

  Adapt the (host=...) and (port=...) lines corresponding to the values you have chosen in the listener.ora file.

Test your setup by starting the listeners as ora<sid1/2>:

```
lsnrctl start LISTENER<SID1/2>
```

Create an `/etc/services` entry for the new port you specified above. Use `tlisrv<SID2>` as service name. The name is not needed anyhow. This entry has to be made on *all* hosts that run an instance that belongs to the system. This includes all external application server hosts outside of the cluster.

OS910 **Optional Step:**

If you use the two package concept:

Set the optional parameter `SQLNET.EXPIRE_TIME` in `sqlnet.ora` to a reasonable value in order to take advantage of the Dead Connection Detection feature of ORACLE. The parameter file `sqlnet.ora` resides either in `/usr/sap/trans` or in `$ORACLE_HOME/network/admin`.

The value of `SQLNET.EXPIRE_TIME` determines how often (in seconds) SQL*Net sends a probe to verify that a client-server connection is still active. If the Central Instance switches, the application servers may crash, thereby leaving shadow processes running on the database host. While the CI package cleans up the application server hosts, it does not touch the ORACLE shadow processes running on the database host. Remove them, because their number increases with every CI package switch. After an application server crash, a connection to the database shadow process may be left open indefinitely. If the `SQLNET.EXPIRE_TIME` parameter is specified, SQL*Net sends a probe periodically to determine whether there is an invalid connection that should be terminated. It finds the dead connections and returns an error, causing the server process to exit.

OR915 **Oracle Step:**

If you are using SAP WAS version 4.0A or higher:

Perform the following step as `<ora>sid`. Start the listener if it is not already running:

```
lsnrctl start <listenername>
```

Start `svrmgrl` and type:

```
> connect internal;
> startup;
> select * from sapr3.instvers;
```

The output is similar to:

```
D R3RELEASE  SAPSID SY OPSYS HOSTNAME DBHOSTNA DBTYPE DATUM TIME A STATUS TOOL
INFO

---------- ---------- ---------- -- ---------- -------- -------- -----------
----------- - ---------- ----------
-------------------------------------------------

1 40B        <SID>        <INSTNR> HP-UX      <pri_host>  <pri_host>  ORACLE
19980527 094221 N          2 R3SETUP    [RFCRADDBDIF_IND_IND]CRFCJOB-RADDBDIF

0 40B        <SID>        <INSTNR> HP-UX      <pri_host>  <pri_host>  ORACLE
```

- Review the ID column. This is the primary key of the table. All new entries need a unique key.
- Review the STATUS column. There should be one row containing a 0 as status. This indicates the current, successful installation. Copy this row using a unique key as well as secondary hostnames for HOSTNAME and DBHOSTNAME.

If you use one package concept with a two node cluster, enter the following line in the svrmgrl session and you are done:

```
> insert into sapr3.instvers values
(<key>,'40B',<SID>,<INSTNR>,'HP-UX',<sec_host>,
<sec_host>,'ORACLE','19980527','094252','N','0','MCSG',
'[RFCRADDBDIF_IND_IND]CRFCJOB-RADDBDIF');
```

For two package concept or more cluster nodes add entries for all possible combinations. For example, you need two additional entries for a two-package concept with two cluster nodes to reflect that the packages can run on different nodes:

```
> insert into sapr3.instvers values
  (<key>+1,'40B',<SID>,<INSTNR>,'HP-UX',<pri_host>,
  <sec_host>,'ORACLE','19980527','094252','N','0','MCSG',
  '[RFCRADDBDIF_IND_IND]CRFCJOB-RADDBDIF');
> insert into sapr3.instvers values
  (<key>+2,'40B',<SID>,<INSTNR>,'HP-UX',<sec_host>,
  <pri_host>,'ORACLE','19980527','094252','N','0','MCSG',
  '[RFCRADDBDIF_IND_IND]CRFCJOB-RADDBDIF');
> Refer to SAP OSS note number 144978 for more information
about the INSTVERS table.
```

Stop the database before continuing.

OR916 **Oracle Step:**

Additional steps for Oracle 8.1.x DB:

For the SGeSAP <=V3.02 the sapbackup directory needs to be in $ORACLE_HOME/sapbackup. If you are using an Oracle 8.1.x DB the $ORACLE_HOME directory is set to /oracle/<SID>/8.1.x whereas the sapbackup directory is still placed /oracle/<SID>/sapbackup. It's best practice to create the symbolic link $ORACLE_HOME/sapbackup -> /oracle/<SID>/sapbackup.

OR917 **Oracle Step:**

Additional steps for Application server and Oracle 8.1.x DB:

If you run an Oracle DB >= 8.1.x and install additional R/3 Application Server follow the procedure described in OSS303238 to install the ORACLE client libraries.

If you configure multiple Application servers to be started with the "parallel" startup option in sap.conf, make sure the "tcp_conn_request_max" ndd parameter on the DB-nodes (primary and backup) are configured with an appropriate value:

Example: tcp_conn_request_max = 1024

If this parameter is set too low, incoming tcp connections from starting SAP Application servers that want a connect to the DB via the Oracle Listener may halt. This will hang the starting process of the SAP Application server.

Additional Steps for DB2

OR918 **DB2 Step:**

DB2 installs the DB2 executables on local disks below /opt.

It is necessary to install a copy of the DB2 executables on all node where the DB / DBCI package may run. It is important to synchronize the patch level of the DB2 executables whenever installing fixes. The executables should be installed using the DB2 installer because DB2 checks the version when starting the DB server.

This version checking is based on OS-swinstall routines in HP-UX. The executable should be installed with the following command: db2setup for the standard executables. Use the installpatch command for the

installation of the required fix pack. Details about installing and diskspace requirements can be found in R/3 Installation on UNIX - IBM DB2 UDB Installation for UNIX.

OR919 **DB2 Step:**

In a one packages concept DBCI where Central Instance and Database are always running on the same server, there is no configuration for the client connection needed. Various service ports that are synchronized already in IS170 initiate the client connection for the database.

This configuration is done within the DB2 administration client which will always be installed in addition to the SAP database. In the one package model the administration client resides under the shared directory `/db2/db2as`.

In two packages configurations the following has to be considered: The shared directory in `/db2` is mounted on the DB-server, the `/db2` client directory must be present on the Central Instance server. However in case the DB-server is failing and the DB-package wants to move over to the adoptive node where the CI -package is running, the `/db2` mountpoint would be busy. Therefore it is necessary to install the administration client to `/db2_client` and then create a symbolic link on the adoptive node as follows:

```
ln -s /db2_client/db2as /db2/db2as
```

The installation of an administration client should be performed following the appropriate steps in the R/3 Installation on UNIX - IBM DB2 UDB Installation for UNIX documentation. Make sure that you only select DB2 UDB Administration Client on the Install DB2 screen and select Do not create any Database Instances.

SAP WAS System Configuration

Logon as <sid>adm on the primary node on which the Central Instance has been installed. The appropriate ServiceGuard package should still run on this host in debug mode.

IS920 **Installation Step:**

Change into the profile directory by typing the alias:

```
cdpro
```

In the DEFAULT.PFL change the following entries and replace the hostname with the relocatable name. For example:

```
SAPDBHOST = <relocdb>
rdisp/mshost = <relocci>
rdisp/sna_gateway = <relocci>
rdisp/vbname = <relocci>_<SID>_<instnr>
rdisp/enqname = <relocci>_<SID>_<instnr>
rdisp/btcname = <relocci>_<SID>_<instnr>
rslg/collect_daemon/host = <relocci>
```

The following parameters are only necessary if an application server is installed on the adoptive node. For example:

```
rslg/send_daemon/listen port
rslg/collect_daemon/listen port
rslg/collect_daemon/talk port
```

NOTE If you are using the one package concept use <relocdbci> instead of <relocdb> or <relocci>. Never use server LAN addresses for any of these parameters.

IS930 **Installation Step:**

In the <SID>_DVEBMGS<INSTNR> profile add or modify the following entries:

```
SAPLOCALHOST = <relocci>
SAPLOCALHOSTFULL = <relocci>.<domain>
```

Because we are using relocatable IP addresses for SAP WAS services, the SAP WAS profile parameters SAPLOCALHOST and SAPLOCALHOSTFULL are very important for SAP WAS operations.

NOTE The profile name is sometimes extended by the hostname. You do not need to change this filename to include the relocatable hostname.

SAPLOCALHOST is set to the hostname per default at startup time and is used to build the SAP WAS application server name:

```
<SAPLOCALHOST>_<SID>_<INSTNR>
```

This parameter represents the communication path inside an SAP R/3 system and between different SAP WAS systems. SAPLOCALHOSTFULL is used for rfc-connections. Set it to the fully qualified hostname.

The application server name appears in the server list held by the message server, which contains all instances, hosts and services of the instances. The application server name or the hostname is also stored in some system tables on the database.

When using the default addressing scheme for the application server name, this name changes during a failover to another node because the backup node has a different hostname than the primary node. This creates a need to cleanup ABAP code similar to the processes required in version 1.0 of the SGeSAP Integration scripts.

When setting the SAPLOCALHOST parameter to the name associated with the relocatable IP address - which moves with the package to the other node - the application server name stays the same after a switchover. This enables all services which make use of the application server name to continue working after a switchover without any change.

The name of a relocatable IP address is also used to specify the SAPDBHOST. This is different from the standard setup with previous SGeSAP Integration versions.

IS940 **Installation Step:**

The parameter SAPLOCALHOSTFULL must be set even if you do not use DNS. In this case you should set it to the name without the domain name:

```
SAPLOCALHOSTFULL=<relocci>
```

IS950 **Installation Step:**

If you use a SAP WAS release prior to 3.1H:

Add the following entry to DEFAULT.PFL:

```
gw/netstat =/usr/bin/netstat -in | cut -c 29-
```

IS960 **Installation Step:**

Starting with SAP R/3 releases 4.5 TPPARAM has been replaced by 2 new transport configuration files: DOMAIN.CFG and TP_DOMAIN_<SID>.PFL

Modify the dbhost entries appropriately.

In the file /usr/sap/trans/bin/TPPARAM, modify the dbhost entry as follows:

```
<SID>/dbhost = <relocdb>
```

IS970 **Optional Step:**

If you have already received your licenses from SAP WAS install them on all the nodes where the Central Instance can start. Refer to the *ServiceGuard Extension for SAP R/3 Release Notes* for further information on how to do this. The package comes up without a license, too. But certain restrictions apply to the SAP WAS application. A warning is printed into the package log-file.

SAP WAS will now be ready to run on all cluster nodes. Test the manual startup on all adoptive nodes.

IS980 **Installation Step:**

Switch the packages to the adoptive nodes with debug mode still enabled.

Start SAP WAS as <sid>adm on the adoptive nodes manually. You should end up with SAP WAS running on any cluster node.

IS982 **Installation Step:**

On all cluster nodes, remove the file /etc/cmcluster/debug.

IS984 **Installation Step:**

Connect with a SAPGUI. Import the changed SAP WAS profiles within SAP WAS. The transaction is called RZ10.

After importing the profiles, check with `rsparam` in SE38 if the parameters SAPLOCALHOST and SAPLOCALHOSTFULL are correct. If you do not import the profiles, the profiles within SAP WAS can be edited by the SAP WAS Administrator. The values listed from SAP WAS will be wrong and, when saved, will overwrite the values which edited on the HP-UX level.

IS986 **Installation Step:**

The destination for print formatting, which is done by a Spool Work process, uses the application server name.

If the application server name stays the same because SAPLOCALHOST has been set to the relocatable name, after a switch no changes need to be done.

Printing works consistently. A print job in process at the time of the failure is canceled and needs to be reissued manually after the failover. To make a spooler highly available on the Central Instance server, set the destination of the printer to `<relocci>_<SID>_<INSTNR>` using transaction SPAD.

IS988 **Installation Step:**

Batch jobs can be scheduled to run on a particular instance.

You select a particular instance by its hostname at the time of job scheduling. The application server name and the hostname retrieved from the Message Server are stored in the Batch control tables TBTCO, TBTCS...

When the batch job is ready to run, the application server name is used to start it on the appropriate instance. When using `<relocci>` to build the application server name for the SAP WAS instance, you do not need to change batch jobs, which are tied to the locality, after a switchover, even if the hostname which is also stored in the above tables differs.

IS990 **Installation Step:**

Within the SAP WAS Computing Center Management System (CCMS) you can define operation modes for SAP WAS instances.

An operation mode defines a resource configuration for the instances in your SAP R/3 system. It can be used to determine which instances are started and stopped, and how the individual services are allocated for each instance in the configuration.

An instance definition for a particular operation mode consist of the number and types of work processes as well as start and instance profiles (starting with version 3.0 the CCMS allows profile maintenance from within SAP WAS).

When defining an instance for an operation mode, enter the hostname and the system number of the application server. By using `<relocci>` to fill in the hostname field, the instance is working under control of the CCMS after a failover without any change.

If an instance is running on the standby node in normal operation (and is stopped when switching over), the control console shows this instance to be down (for example, you will get a red node on a graphical display) after the switchover.

IS991 **Installation Step: SGeSAP Configuration for SAP ICM**

A SAP Internet Communication Manager (ICM) may run as part of any Application Server. It is started as a separate multi-threaded process and can be restarted independently from the Application Server. E.g. usage of BW web reporting, Business Server Pages (BSP) rely on ICM to work properly.

In scenarios using external web communication, one or more SAP Web Dispatcher often accompany the ICMs. The SAP Web Dispatcher can be layed out redundantly with hardware load balancing. SAP also allows SGeSAP packaging for a single Web Dispatcher.

To find out if ICM is configured as part of an Application Server, check for the following SAP profile parameter:

IS995 **Installation Step:**

Configure the frontend PCs to attach to `<relocci>`. Most of the time, this can be achieved by distributing a new `saplogon.ini` file to the windows directory.

Installation is complete.

The next step is to do some comprehensive switchover testing covering all possible failure scenarios. It is important that all relevant SAP WAS application functionalities are tested on the switchover nodes. There exist several documents provided by HP or SAP WAS that can guide you through this process.

```
rdisp/start_icman=TRUE
```

ICM gets switched over with the Instance package. In order to make it work, ICM has to be registered with the message server using a virtual IP address.

This mechanism is different from SAPLOCALHOST and SAPLOCALHOSTFULL since ICM allows HTTP Server Aliasing via icm/server_port_<nn> settings. During startup, icman reads its configuration and propagates it to the SAP Message Server or the SAP Web Dispatcher Server.

These servers then act as the physical point of access for HTTP requests. They classify requests and send HTTP redirects to the client in order to connect them to the required ICM Instance. This only works properly if the bound ICM instances propagate virtual IP addresses. Therefore the following parameter needs to be set within the Instance Profile of the Application Server:

```
icm/host_name_full=<relocatable_ip>
```

Doublecheck the settings with report RSBB_URL_PREFIX_GET or review the parameters within the SMMS transaction.

SGeSAP Installation verification: CCmon Installation and Configuration

OS998

Installation Step: CCmon installation

After SGeSAP has been successfully installed and configured, it is possible to install the Cluster Consistency Monitor as well. There is a setup tool available with CCmon. It has to be called as follows:

```
# cd /opt/cmcluster/sap/ccmon
```

```
# ./Setup_Ccmon -v
```

```
# Setup_CCmon: Installation Tool for the Cluster Consistency
Monitor

# Node           : hpcc006
# Time           : Fri Dec  7 10:06:11 2001
# Version       : 1.00 (32 Bit), compiled at  15. November 2001
for HP-UX 11.00
# Build          : 20
# Author         : SAP/HP Competence Center, Walldorf, Germany

10:06 Setup_CCmon(): Execute: "/usr/bin/tar -xf
/opt/cmcluster/sap/ccmon/ccmon.tar"
10:06 Setup_CCmon(): Found O/S Version "B.11.11"
10:06 Setup_CCmon(): Execute: "/usr/bin/tar -xf
/opt/cmcluster/sap/ccmon/ccmon.tar"
10:06 Setup_CCmon(): Found 64 Bit enabled in the O/S

You may now configure the following Profiles:
-> /opt/cctool/etc/profile/SCAN_RSDB.pfl
-> /opt/cctool/etc/profile/COMP_MCSG.pfl
-> /opt/cctool/bin/CheckMCSG
-> Configure your Alarm Handlers
-> Copy the Configuration to other Clusternodes
-> Insert an Entry into the crontab of each Node

*** More Information on this Configuration Steps is in the
Installation
*** Guide /opt/cctool/doc/CCmon_Handbook.doc (Microsoft Word
Format)
*** Normal End of Programm, Bye ...
```

After this step has been finished, you will find more information about how to configure and execute the CCMon in the manual located at: `/opt/cctool/doc/CCmon_Handbook.doc`.

For a temporary (10 Days valid) or a final Installation, it is required to install a license. Please contact your HP representitive to generate this license for your installation. For the license generation a hardware key of the system you are running CCmon is required. Generate the key on each node you want to run CCmon.

The following command generates the hardware key: `/opt/cctool/bin/licgen -get`.

>Node-Key: 1705415577

Send this key to your HP representative and you will obtain a license key to install.

3 SAP Supply Chain Management

Within SAP Supply Chain Managment (SCM) there are two main technical components: the APO System and the LiveCache. An APO System is based on SAP Web Application Server (WAS) technology. Thus, ci, db, dbci, sapnfs and appSIDno ServiceGuard packages may be implemented for APO. These APO packages are set up similar to R/3 Enterprise packages with one exception. APO needs access to LiveCache client libraries. These files are found via the path `/sapdb/programs`. How access to the client libs is ensured depends on the setup of the LiveCache, which are described in combination with the LiveCache volumegroup layout.

The second technical SCM component, called LiveCache, is an in-memory database. The technical realization of LiveCache is based on SAPDB technology. Even though LiveCache is somewhat similar to the SAP owned database technologies, and has special requirements. This is because a LiveCache Instance usually is closely coupled with the accompanying APO. In order to allow support for LiveCache High Availabilit, SAP has defined a list of policies and requirements of failover solutions.

This chapter describes how to configure and setup SAP Supply Chain Management using a failover LiveCache. Topics are as follows:

- Planning the Volume Manager Setup
- HP-UX Setup
  - — Cluster Node Synchronization
  - — Cluster Node Configuration
- SGeSAP Package Configuration
  - — Package Creation
  - — Service Monitoring
- APO Setup Changes
- General ServiceGuard Setup Changes

The tasks are presented as a sequence of steps. Each mandatory installation step is accompanied by a unique number of the format *XXnnn*, where *nnn* are incrementing values and *XX* indicates the step relationship, as follows:

- LCnnn—LC Package Installation Steps
- GSnnn—General Installation Steps that manipulate already existing SGeSAP integrations

Whenever appropriate, HP-UX sample commands are given to guide you through the process in as detailed a manner as possible. It is assumed that hardware as well as the operating system and MC/ServiceGuard are already installed properly on all cluster hosts. Sometimes a condition is specified with the installation step. Follow the information presented *only* if the condition is true for your situation.

NOTE For installation steps in this chapter that require the adjustment of SAP specific parameter in order to run the SAP R/3 system in a switchover environment usually example values are given. These values are for reference ONLY and it is recommended to read and follow the appropriate SAP OSS notes for SAP's latest recommendation. Whenever possible the SAP OSS note number is given.

Planning the Volume Manager Setup

The following describes the lc package of SGeSAP and was developed according to the SAP recommendations, which fulfills all SAP requirements for LiveCache failover solutions.

LiveCache distinguishes an instance dependant path /sapdb/<LCSID> and two instance independent paths; IndepData and IndepPrograms. By default all three point to the directory below `/sapdb`.

NOTE <LCSID> denotes the three letter database name of the LiveCache instance in uppercase and is the same name in lowercase `<lcsid>`.

An overview of the current settings can be viewed at the file `/var/spool/sql/ini/SAP_DBTech.ini`.

The following is a sample SAP_DBTech.ini for a host with a LiveCache 7.4 (LC1) and an APO 3.1 using a SAPDB 7.3 database instance (AP1):

```
[Globals]
IndepData=/sapdb/data
IndepPrograms=/sapdb/programs
[Installations]
/sapdb/LC1/db=7.4.2.3,/sapdb/LC1/db
/sapdb/AP1/db=7.3.0.15,/sapdb/AP1/db
[Databases]
.SAPDBLC=/sapdb/LC1/db
LC1=/sapdb/LC1/db
_SAPDBAP=/sapdb/AP1/db
AP1=/sapdb/AP1/db
[Runtime]
/sapdb/programs/runtime/7240=7.2.4.0,
/sapdb/programs/runtime/7250=7.2.5.0,
/sapdb/programs/runtime/7300=7.3.0.0,
```

```
/sapdb/programs/runtime/7301=7.3.1.0,
/sapdb/programs/runtime/7401=7.4.1.0,
/sapdb/programs/runtime/7402=7.4.2.0,
```

Changing this file in an editor is not allowed.

NOTE [Globals] section is commonly shared between LC1 and AP1. This prevents setups that keep the directories of LC1 and AP1 completely separated.

There are different configuration options for the filesystems caused by a tradeoff between simplicity and flexibility. The options described below are in order of increasing complexity. The following bullet list describe cluster layout constraints that need to be fulfilled to allow the simplifications of a given option.

LVM Option1: Simple Clusters with Separate Packages

Cluster Layout Constraints:

- The LiveCache package does not share a failover node with the APO Central Instance package.
- There is no SAPDB or additional LiveCache running on cluster nodes.
- There is no intention to install additional APO Application Servers within the cluster.

Table 3-1 File System Layout for LiveCache Package running separate from APO (Option 1)

storage Type	package	mount Point	volume group	logical volume	device number
shared	lc<LCSID>	/sapdb			
shared	lc<LCSID>	/sapdb/<LCSID> /data*n*			
shared	lc<LCSID>	/sapdb/<LCSID> /log*n*			
shared	lc<LCSID>	/var/spool/sql			
shared	dbci<SID> or db<SID>[1]	/sapdb/programs			

[1]This denotes the package for the database of the APO Central Instance. The logical volume should be added to the logical volumes that already belong to this package.

In the above layout all relevant files get shared via standard procedures. The setup causes no administrative overhead for synchronizing local files. SAP default paths are used.

NOTE The data and log disk spaces of SAPDB are called. For security and performance reasons SAP recommends placing the *devspaces* on raw devices. All logical volumes mounted to data*n* and saplog*n* are *devspaces*.

LVM Option 2: Non-SAPDB Environments

Cluster Layout Constraints:

- There is no SAPDB or additional LiveCache running on cluster nodes. Especially the APO System RDBMS is either based on Informix, ORACLE or DB2, but not on SAPDB.

Often APO does not rely on SAPDB as an underlying database technology. But independent from that, all Instances of the APO System still need access to the LiveCache client libraries. Therefore, it is recommended to make the client libraries available throughout the cluster via AUTOFS cross-mounts from a dedicated NFS package.

Table 3-2 File System Layout for LiveCache in a non-SAPDB Environment (Option 2)

storage type	package	mount point	volume group	logical volume	device number
shared	lc<LCSID>	/sapdb/data			
shared	lc<LCSID>	/sapdb/<LCSID>			
shared	lc<LCSID>	/sapdb/<LCSID> /data*n*			
shared	lc<LCSID>	/sapdb/<LCSID> /log*n*			
shared	lc<LCSID>	/var/spool/sql			
autofs shared	sapnfs[2]	/sapdb/programs			

[2]This can be any standard, standalone NFS package. The SAP global transport directory should already be configured in a similar package. This is why this package is often referred to as “trans package”, which can optionally be extended to also serve the global LiveCache fileshares.

LVM Option 3: Full Flexibility - No Contraints

If multiple SAPDB based components are either planned or already installed, the setup looks different. All directories that are shared between SAPDB instances must not be part of the LiveCache package. Otherwise, a halted LiveCache package would prevent other SAPDB instances from being started.

The following directories are affected:

`/sapdb/programs`: This is a central directory with all Livecache/SAPDB executables, which is shared between all Livecache/SAPDB Instances that reside on the same host. It is also possible to share the directory

across hosts. However, it is not possible to use different executable directories for two Livecache/SAPDB Instances on the same host. Furthermore, it is not unusual to install different SAPDB versions on the same host. This can be seen in the LC1/AP1 example of the SAP_DBTech.ini file printed above. It is a standard version combination for SCM3.1.

During normal operation, the LiveCache most likely won't share the host with the APO database. But the failover might allow mutual backup between the LiveCache host and the APO host. This implies that the files in `/sapdb/programs` have to be of the newest version that any SAPDB in the cluster has. Files in `/sapdb/programs` are downwards compatible. For the LiveCache 7.4 and the APO 3.1 using SAPDB 7.3 this means that within /sapdb/programs there have to be the 7.4 version executables installed.

`/sapdb/data/config`: This directory is also shared between instances, though you can find many files that are Instance specific in `/sapdb/data/config/<LCSID>.*` According to SAP, this path setting is static.

`/sapdb/data/wrk`: The working directory of the main Livecache/SAPDB processes are also a subdirectory of the IndepData path for non-HA setups. If a LiveCache restarts after a crash, it copies important files from this directory to a backup location. This information is then used to determine the reason of the crash. In HA scenarios, this directory will move with the package. Therefore, SAP provided a way to redefine this path for each Livecache/SAPDB individually. SGeSAP expects the work directory to be part of the lc package. The mount point moves from `/sapdb/data/wrk` to `/sapdb/data/<LCSID>/wrk`. This directory should not be mixed up with the directory `/sapdb/data/<LCSID>/db/wrk` that might also exist.

Core files of the kernel processes are written into the working directory. These core files have file sizes of several Gigabytes. Sufficient free space needs to be configured for the shared logical volume to allow core dumps. Files of that size should not be transferred via NFS.

`/var/spool/sql`: This directory hosts local runtime data of all locally running Livecache/SAPDB Instances. Most of the data in this directory becomes meaningless in the context of a different host after failover. The

only critical portion that still has to be accessible after failover is the initialization data in /var/spool/sql/ini. This directory is almost always very small (< 1Megabyte).

Table 3-3 General File System Layout for LiveCache (Option 3)

storage type	package	mount point	volume group	logical volume	device number
shared	lc<LCSID>	/sapdb/<LCSID>			
shared	**lc<LCSID>**	**/sapdb/<LCSID>/wrk**			
shared	lc<LCSID>	/sapdb/<LCSID>/data*n*			
shared	lc<LCSID>	/sapdb/<LCSID>/saplog*n*			
autofs shared	sapnfs	/export/sapdb/programs			
autofs shared	sapnfs	/export/sapdb/data			
autofs shared	sapnfs	/export/var/spool/sql/ini			

LC010 LiveCache Installation Step:

If you decided to use option three:

Log on as <lcsid>adm onto the machine that LiveCache is installed. As described above, ensure that you have mounted a sharable logical volume on `/sapdb/<LCSID>/wrk` .

Change the path of the runtime directory of the LiveCache and move the files to the new logical volume as follows:

```
cd /sapdb/data/wrk/<LCSID>
find . -depth -print | cpio -pd /sapdb/<LCSID>/wrk
cd ..
rmdir /sapdb/data/wrk/<LCSID>
dbmcli -d <LCSID> -u control,control
dbmcli on <LCSID>>param_directget RUNDIRECTORY
OK
```

```
RUNDIRECTORY    /sapdb/data/wrk/<LCSID>

---

dbmcli on <LCSID>>param_directput RUNDIRECTORY
/sapdb/<LCSID>/wrk

OK

---

dbmcli on <LCSID>>
```

LC020 **LiveCache Installation Step:**

Mark all LiveCache volume groups as members of the cluster, which only works if the cluster services are already available. For example:

```
cd /

# umount all logical volumes of the volume group

vgchange -a n <vg_name>

vgchange -c y <vg_name>

vgchange -a e <vg_name>

# remount the logical volumes·
```

The device minor numbers must be different from all device minor numbers gathered on the other hosts. Distribute the shared volume groups to all potential failover nodes.

HP-UX Setup

Cluster Node Synchronization

Repeat the steps in this section for each node of the cluster, which are different from the primary. Logon as root to the primary host and prepare a logon for each of its backup hosts.

LC030 **LiveCache Installation Step:**

Synchronize the files in `/etc/group and /etc/passwd`. The LiveCache installation has created a <lcsid>adm user belonging to the sapsys group. Make sure the user and group exist on all nodes and that UID and GID are consistent across the nodes.

LC040 **LiveCache Installation Step:**

Synchronize the sql<nn> entries from the `/etc/services` file between the nodes to match the entries on the primary node. Usually you will find service entries for sql6 and sql30.

LC050 **LiveCache Installation Step:**

Change the HP-UX kernel on the backup nodes to meet the SAP LiveCache requirements as specified in the SAP LiveCache installation documents.

LC060 **LiveCache Installation Step:**

Copy the content of the <lcsid>adm home directory to the backup node. This is a local directory on each node.

Rename the environment scripts on the secondary nodes. Some of the environment scripts may not exist. For example:

```
su - <lcsid>adm
mv .dbenv_<primary>.csh .dbenv_<secondary>.csh
mv .dbenv_<primary>.sh .dbenv_<secondary>.sh
```

NOTE Never use the relocatable address in these file names.

LC070 **LiveCache Installation Step:**

Make sure `/var/spool/sql` exists as a directory on the backup node. `/usr/spool` must be a symbolic link to `/var/spool`.

LC080 **LiveCache Installation Step:**

On the backup node, create a directory as future mountpoint for all relevent directories from the table of section *Planning the Volume Manager Setup* that refers to the layout option you chose.

```
Option 1:

mkdir /sapdb

Option 2:

mkdir -p /sapdb/data

mkdir /sapdb/<LCSID>

Option 3:

mkdir -p /sapdb/<LCSID>
```

Cluster Node Configuration

Repeat the steps in this section for each node of the cluster.

LC100 **LiveCache Installation Step:**

Check that `/etc/logingroup` is a link to `the file /etc/group`.

LC110 **LiveCache Installation Step:**

Add all relocatable IP address information to `/etc/hosts`. Remember to the heartbeat IP addresses.

LC120 **LiveCache Installation Step:**

If you use DNS:

Configure /etc/nsswitch.conf to avoid problems. For example:

```
hosts: files[NOTFOUND=continue UNAVAIL=continue
```

TRYAGAIN=continue]dns

LC130 **LiveCache Installation Step:**

If you establish frontend and server LANs to separate network traffic:

Add routing entries to the internet routing configurations of `/etc/rc.config.d/netconf`. This is the only phase of the whole installation in which you will need to specify addresses of the server LAN. Route all relocatable client LAN addresses to the local server LAN addresses. For example:

```
ROUTE_DESTINATION[n]=<reloclc>
ROUTE_MASK[n]=""
ROUTE_GATEWAY[n]=<reloclc_s>
ROUTE_COUNT[n]=1
ROUTE_ARGS[n]=""
```

SGeSAP Package Configuration

Package Creation

Log on as root on any cluster host.

LC150 **LiveCache Installation Step:**

Install the product depot file for SGeSAP (B7885BA) using swinstall (1m) if this has not been done already. The installation staging directory is `/opt/cmcluster/sap`. All original product files are copied there for reference purposes.

LC160 **LiveCache Installation Step:**

Copy the relevant SGeSAP files for LiveCache to the cluster directory:

```
mkdir /etc/cmcluster/<LCSID>
cp /opt/cmcluster/sap/LC/saplc.mon /etc/cmcluster/<LCSID>
cp /opt/cmcluster/sap/LC/saplc.cntl /etc/cmcluster/<LCSID>
cp /opt/cmcluster/sap/LC/sap.conf /etc/cmcluster/<LCSID>
```

If this has not already been done, also copy the system independent integration files:

```
cp /opt/cmcluster/sap/*.functions /etc/cmcluster
```

LC170 **LiveCache Installation Step:**

Create standard package control and configuration files:

```
cd /etc/cmcluster/<LCSID>
cmmakepkg -s lc<LCSID>.cntl
cmmakepkg -p lc<LCSID>.conf
```

Both files need to be adapted accordingly to reflect the filesystem layout option chosen and the individual network setup. For general information on how to customize the package control and configuration files refer to the *Managing MC/ServiceGuard* manual, which is available on *http://docs.hp.com*.

For lc<LCSID>.conf the following settings are strongly recommended:

```
PACKAGE_NAMElc<LCSID>
```

```
PACKAGE_TYPEFAILOVER
RUN_SCRIPT         /etc/cmcluster/<LCSID>/lc<LCSID>.cntl
HALT_SCRIPT       /etc/cmcluster/<LCSID>/lc<LCSID>.cntl
SERVICE_NAME          SGeSAPlc<LCSID>
```

For lc<LCSID>.cntl the following settings are strongly recommended:

```
SERVICE_NAME[0]="SGeSAPlc<LCSID>"
SERVICE_CMD[0]="/sapdb/<LCSID>/db/sap/lccluster monitor"
SERVICE_RESTART[0]="-r 3"
```

All other parameters should be chosen as appropriate for the individual setup.

LC180 **LiveCache Installation Step:**

In lc<LCSID>.cntl enable the SAP specific steps within the customer_defined_commands sections:

```
function customer_defined_run_cmds
{
/etc/cmcluster/<SID>/saplc.cntl startDB <LCSID>
test_return 51
}
function customer_defined_halt_cmds
{
/etc/cmcluster/<SID>/saplc.cntl stopDB <LCSID>
test_return 52
}
```

LC190 **LiveCache Installation Step:**

The sap.conf configuration file of a LiveCache package is similar to the configuration file of mySAP Instance packages.

The following standard parameters in sap.conf have to be set for a LiveCache package:

```
DB=LIVECACHE
DBRELOC=<reloclc_s>
TRANSRELOC=<relocsapnfs_s>
```

```
CLEANUP_POLICY= [ strict | normal | lazy ]
```

NOTE The strict option can be set in order to guarantee the LiveCache package gets all resources required on the adoptive node. *This option can crash a system running on the adoptive node.*

The following LiveCache specific parameters in `sap.conf` have to be set:

```
# The mode of the LiveCache database up to which it should be
# started automatically. Possible values are
# OFFLINE: only vserver will be started
# COLD:    LiveCache started in cold mode
# SLOW:    LiveCache started in cold -slow mode
# WARM:    LiveCache started in warm mode
# After failover, LiveCache will only be recovered to the
# state referenced in LCSTARTMODE.
#
LCSTARTMODE=WARM
# The behaviour of the LiveCache service monitoring script is
# independent from the setting of LCSTARTMODE!
# The monitoring script will always monitor the vserver
process.
# It will start to monitor if LiveCache is in WARM state,
# as soon as it reaches WARM state for the first time.
# The LCMONITORINTERVAL variable specifies how often the
# monitoring polling occurs (in sec.)
#
LCMONITORINTERVAL=5
```

LC200 **LiveCache Installation Step:**

The following parameters in sap.conf can optionally be used for a LiveCache package: RMNR[], RMDEP[], RMADM[], RMDEPNR[], RMDEPHOST[], RMDEPPLATFORM[0].

Sometimes, when a failover occurs, you want to stop other SAP R/3 Instances running on backup nodes.

This is useful if you have to free up limited resources before you are able to start the failed Instance again. To use this functionality configure two groups of arrays:

The first group consists of the arrays RMNR[*], RMADM[*] and RMDEP[*].

Specify the Instance-IDs and the name of the System Administrators of each instance that shall be stopped in the arrays RMNR[*] and RMADM[*].

For Dialog-Instances and other additional application servers also specify:

```
RMDEP[*]=-1
```

This means, that there are no instances which depend on the services offered by them.

For Central Instances, specify the index into the second group of arrays in RMDEP[*].

If you allow stopping of Central Instances, you also have to specify the list of related additional application servers. The application servers are stopped first. It does not matter on which host they are running.

The second group of arrays consists of RMDEPNR[*] and RMDEPHOST[*].

Specify the application servers depending on the Central Instances which are allowed to be stopped. Fill in the Instance-IDs and the local hostnames on which the application servers run. The list of application servers which belong to the same SAP R/3 system start at the index which is specified in the RMDEP[*] entry of their Central Instance and the list continues until an RMDEPNR[*]=-1 is found.

The RMNR[*] specified instances are halted only if they are running on the host you currently fail over to. This functionality never halts any package. It is possible to stop a Central Instance that is using a ServiceGuard package, but the associated package remains in the running state. Application servers belonging to the same SAP R/3 system as the failing package do not need to be specified here. They stop automatically.

LC205 **LiveCache Installation Step:**

Distribute the content of /etc/cmcluster/<LCSID> to all cluster nodes on which the LiveCache should be able to run.

Service Monitoring

SAP recommends the use of service monitoring in order to test the runtime availability of LiveCache processes. The monitor, provided with SGeSAP, periodically checks the availability and responsiveness of the LiveCache system. The sanity of the monitor will be ensured by standard ServiceGuard functionality.

The LiveCache monitoring program is shipped with SGeSAP in the saplc.mon file. The monitor runs as a service attached to the lc<LCSID> ServiceGuard package.

If the monitor recognizes the LiveCache to be unavailable, it will try to restart LiveCache on the node it is currently running. If this does not succeed, the runtime operating system resources of LiveCache are cleaned up and another local restart attempt is made. If this still does not bring LiveCache back to work, ServiceGuard will switch the package and try to restart on a different node.

Monitoring begins with package startup. At this point, the monitor will make sure that LiveCache is working only up to the point that is specified in LCSTARTMODE.

For example, if LCSTARTMODE=OFFLINE is set in sap.conf, only the vserver processes will be part of the monitoring. The monitor detects any manual state change of LiveCache. By subsequent manual operations, LiveCache will enter cold state and finally warm state, which the monitor detects automatically. Once the LiveCache reaches the warm state once, the monitoring increases its internal monitoring level to the same state.

As any SGeSAP package, the lc package will skip SAP specific startup steps when during startup a file called debug is found in either /etc/cmcluster/<LCSID> or /etc/cmcluster. This is useful for debugging purposes to allow access to the logfiles placed on shared logical volumes if a package does not start up to its full extend. The lc monitor is started regardless of the availability of a debug file. The monitor will detect the existence of the file and will enter a pause mode until it is removed. This is valid not only during package startup, but also at runtime. As a consequence, the monitoring of a fully started package will be paused by the global debug file.

NOTE Upon activation of pause mode, the state changes of LiveCache and LiveCache restart attempts get permanently logged into the standard package logfile `/etc/cmcluster/<LCSID>/lc<LCSID>.cntl.log`.

The monitor can also be paused by standard administrative tasks that use the administrative tools delivered by SAP. Stopping the LiveCache using the SAP lcinit shell command or the APO LC10 transaction will send the monitoring into pause mode. This prevents unwanted package failovers during LiveCache administration. Restarting the LiveCache in the same way will also trigger reactivation of the monitoring. The SAPDB Database Manager GUI is not yet cluster-aware and must not be used to stop LiveCache in combination with ServiceGuard.

LC210 **LiveCache Installation Step:**

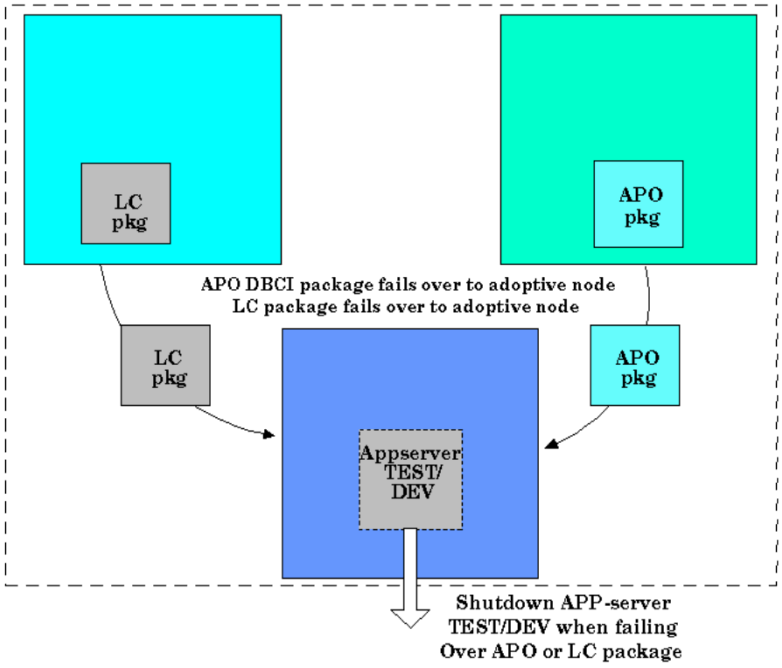
Log in as <LCSID>adm on the primary node that has the shared logical volumes mounted. Create a symbolic link that acts as a hook that informs SAP software where to find the LiveCache monitoring software to allow the prescribed interaction with it.

```
ln -s /etc/cmcluster/<LCSID>/saplc.mon
/sapdb/<LCSID>/db/sap/lccluster
```

APO Setup Changes

Running LiveCache within a ServiceGuard cluster package means that the LiveCache instance is now configured for the relocatable IP of the package. This configuration needs to be adopted in the APO system that connects to this LiveCache. Figure 2-1 shows an example for configuring LCA to reloc8.

Figure 3-1 Example HA SCM Layout



GS220 LiveCache Installation Step:

Run SAP transaction LC10 and configure the logical LiveCache names LCA and LCD to listen to the relocatable IP of the LiveCache package.

GS230 LiveCache Installation Step:

Configure the XUSER file in the APO user home directory.

The XUSER file in the home directory of the APO Administrator keeps the connection information and grant information for a client connecting to LiveCache. The XUSER content needs to be adopted to the relocatable IP the LiveCache in running on.

To list all mappings in the XUSER file, run the following command as <SID>adm of the APO user:

dbmcli -ux SAPR3,SAP -ul

This command produces a list of SAPDB user keys that may be mapped to the LiveCache database schema via a local hostname. SAP created keys commonly include c, w and DEFAULT. If a mapping was created without specifying a key, entries of the form <num><LCSID><hostname>, e.g. 1LC1rhpcc072 exist. These will only work on one of the cluster hosts.

To find out if a given user key mapping <user_key> works throughout the cluster, the relocatable address should be added to the primary host using cmmodnet -a.

Run the following command as APO Administrator:

 # dbmcli -U <user_key>

quit exits the upcoming prompt:

dbmcli on <hostname> : <LCSID>> quit

<hostname> should be relocatable. If it is not, the XUSER mapping has to be recreated. Example:

**# dbmcli -d <LCSID> -n <reloclc_s> -us <user>,<passwd> **

-ux SAPR3,SAP -uk <user_key>

Entries of the form <num><LCSID><reloclc_s> could be created by omitting the -uk option.

Refer to the SAP documentation to learn more about the dbmcli syntax. After recreation always check the connection:

dbmcli -U <user_key>

dbmcli on <reloclc_s> : <LCSID>> quit

GS240 **LiveCache Installation Step:**

Distribute the XUSER mappings by distributing the file /home/<APOSID>adm/.XUSER.<version> to all APO package nodes.

General ServiceGuard Setup Changes

Dependent from the LVM option chosen, the globally available directories need to be added to different existing ServiceGuard packages. The following installation steps require that the system has already been configured to use the automounter feature. If this is not the case, refer to installation steps IS730 to IS770 found under in Chapter 2 , “Automounter/Auto FS Configuration,” on page 96.

GS250 **LiveCache Installation Step:**

For LVM Option 1:

A logical volume for `/sapdb/programs` should already exist within the NFS package part of the dbci<APOSID> package, which contains the SAPDB client libraries. In case of a two-package installation for APO, the logical volume is part of the NFS package part of the db<APOSID> package.

Copy the content of `/sapdb/programs` from the LiveCache primary node to this logical volume, which should be a superset of the files that already exist.

Make sure `/sapdb/programs` exists as empty mountpoint on all hosts of the LiveCache package.

Make sure /sapdb/programs exists as empty mountpoint on all hosts of the APO package.

Add the following entry to `/etc/auto.direct.` on all LiveCache package nodes: `/sapdb/programs <relocdbci_s>:/sapdb/programs`.

For LVM Option 2:

Add a shared logical volume for `/export/sapdb/programs` to the global NFS package (sapnfs). Copy the content of `/sapdb/programs` from the LiveCache primary node to this logical volume.

Make sure `/sapdb/programs` exists as empty mountpoint on all hosts of the LiveCache package. Also make sure `/export/sapdb/programs` exists as empty mountpoint on all hosts of the sapnfs package.

Add the following entry to /etc/auto.direct. on all hosts of the LiveCache package:

`/sapdb/programs<relocsapnfs_s>:/export/sapdb/programs`

For LVM Option 3:

Add two shared logical volume for /export/sapdb/programs and /export/sapdb/data to the global NFS package (sapnfs). Copy the content of /sapdb/programs and the remaining content of /sapdb/data from the LiveCache primary node to these logical volumes.

Make sure /sapdb/programs and /sapdb/data exist as empty mountpoints on all hosts of the LiveCache package. Also make sure /export/sapdb/programs and /export/sapdb/data exist as empty mountpoints on all hosts of the sapnfs package.

Add the following entries to /etc/auto.direct. on all hosts of the LiveCache package:

```
/sapdb/programs <relocsapnfs_s>:/export/sapdb/programs

/sapdb/data <relocsapnfs_s>:/export/sapdb/data
```

GS260 **LiveCache Installation Step:**

Distribute all configuration files from the sapnfs (or dbci<APOSID>, db<APOSID>) package directory to the cluster nodes, which the package is able to run.

LC270 **LiveCache Installation Step:**

The last step is to reconfigure the cluster with cmapplyconf(1m). In particular, the new LiveCache package lc<LCSID> can now be specified.

4 SGeSAP Administration

This chapter describes administration of the special features of the ServiceGuard Extension for SAP (SGeSAP). Topics presented in this chapter include:

- Switching SGeSAP Off and On
- SGeSAP Administration Issues

Refer to your SAP Web Application Server (WAS) documentation for more information about managing SAP WAS environments. For HP-UX system administration, refer to the HP-UX manual *Managing Systems and Workgroups* (Part Number B2355-90157), and for ServiceGuard issues, refer to Managing MC/ServiceGuard (Part Number B3936-90065 for HP-UX 11i).

Switching SGeSAP Off and On

This section provides a brief description of how to switch off SGeSAP. Your individual setup may require additional steps that are not included in this document. Contact your HP consultant for additional information.

Switching off the SGeSAP Integration means that the SAP WAS system will not run on the relocatable IP address. If you are performing an SAP WAS upgrade, make sure that your SAP WAS does not require any relocatable IP addresses. Switching off the SGeSAP Integration is the first step in SAP WAS upgrade related work. This ensures full support from SAP WAS during the upgrade procedure.

Switch Off SGeSAP

Refer to the following sections for descriptions of how to switch of SGeSAP.

Switching off SGeSAP starts with the following processes. Perform the steps described no matter which database you are using:

- Halt Packages
- Turn on Debug Mode
- Start Packages

Perform the steps in one of these two sections, selecting the additional steps for ORACLE or INFORMIX as appropriate to your database.

- Additional steps for ORACLE
- Additional steps for INFORMIX

Follow the steps in these sections to set up SAP WAS internally. Follow these steps no matter which database your system has. They are always the same, independent from the underlying database:

- Check SAP WAS
- Backup Files Again

Halt Packages

If the SGeSAPpackages are still running, halt them using:

```
cmhaltpkg <package>
```

If you use the two package concept, stopping the database package also brings down the Central Instance package. If you use an additional package that contains the transport directory, you do not need to stop that package. This allows any other SAP R/3 system that shares the transport directory to not be affected and continue running.

Turn on Debug Mode

Turn on the debug mode of the SGeSAP Integration. Then ServiceGuard can provide all the filesystems and relocatable IP addresses, but the scripts do not start the database or SAP WAS. If a file named debug exists in `/etc/cmcluster` SGeSAP Integration packages start and no SAP WAS specific script starts. For example:

You must create a debug-file on all nodes of the cluster:

```
touch /etc/cmcluster/debug
```

Using this command prevents the SAP R/3 system from starting up on the other host as the result of a failure. This ensures that SAP WAS does not run with incorrect profiles.

If you have more than one SAP R/3 system within the cluster you can also turn on debug mode for specific systems. If the file `/etc/cmcluster/<SID>/debug` exists, the additional scripts (start and stop SAP WAS) for this system are not started. Other systems are controlled as usual by ServiceGuard.

Start Packages

Start the packages with:

```
cmrunpkg <package>
```

Remember that with the debug mode turned on, the Central Instance package of the two package concept is not triggered by the database package. You have to start it manually.

Additional steps for ORACLE

Additional steps for switching off SAP WAS with an ORACLE database include:

- Backup Files

Save the configuration files so you can switch on the SGeSAP Integration again. Save the files as user <sid>adm. This list of files you have to save include:

```
listener.ora
/usr/sap/trans/bin/TPPARAM
tnsnames.ora
/usr/sap/<SID>/SYS/profile/DEFAULT.PFL
/usr/sap/<SID>/SYS/profile/<SID>_DVEBMGS<INSTNR>
```

- Stop Listener

  If you have an oracle database check to see if the listener is running. If it is, stop it as user ora<SID> with:

  ```
  lsnrctl stop
  ```

  If you have configured names for the listeners, which allows you to have more than one listener running, make sure to specify the correct name:

  ```
  lsnrctl start <LISTENERNAME>
  ```

- Change Profiles

  The references of the relocatable name must be changed to the real hostname in the following files:

  ```
  listener.ora
  tnsnames.ora
  /usr/sap/<SID>/SYS/profile/DEFAULT.PFL
  /usr/sap/<SID>/SYS/profile/<SID>_DVEBMGS<INSTNR>
  /usr/sap/trans/bin/TPPARAM
  ```

NOTE /usr/sap/trans/listener.ora is a link to /etc/listener.ora.

- Start Listener

  Start oracle listener as oracle user with:

  ```
  lsnrctl start <LISTENERNAME>
  ```

Additional steps for INFORMIX

Additional steps for switching off SAP WAS with an INFORMIX database include:

- Rename Customer Shell Scripts

  In the home directories of the users `Informix` and `<sid>adm` you can find two files called `.customer.sh` and `.customer.csh`. Rename them to prevent them from being executed the next time the database starts. For example:

  ```
  mv .customer.csh .customer.sh.off
  mv .customer.csh .customer.csh.off
  ```

  Execute these lines in both directories. Make sure not to change owner or permissions of the files. Even though the home directory of `<sid>adm` is usually local, it is enough to do this on only one machine of the cluster. Choose the machine where you want to run the database later.

- Backup Files

  Save the configuration files so you can switch the SGeSAP Integration on again. The list of files you have to save includes:

  ```
  /informix/<SID>/etc/onconfig.<rel-IP_name_of_db>.<DBID>
  /informix/<SID>/etc/sqlhosts.soc
  /usr/sap/trans/bin/TPPARAM
  /usr/sap/<SID>/SYS/profile/DEFAULT.PFL
  /usr/sap/<SID>/SYS/profile/<SID>_DVEBMGS<INSTNR>
  ```

- Change Profiles

  Replace any reference to the relocatable IP address(es) in the files you previously saved. Insert the local hostname of your machine. This is the name you get when you use the hostname command.

  As `informix` user go to `/informix/<SID>/etc`. In the `onconfig` file replace the:

  `<rel-IP_name_of_DB><INFORMIX_DBID><shm|tcp>`

  entries with:

  `<local_hostname><INFORMIX_DBID><shm|tcp>`

  entries, for example:

```
DBSERVERNAME <local_hostname><sid>shm
     # Name of default database server
DBSERVERALIASES <local_hostname><sid>tcp
     # List of alternate dbservernames
```

Afterwards rename the file:

```
mv onconfig.<rel-IP_name_db>.<DBID> \
onconfig.<local_hostname>.<DBID>
```

In the same way replace the appropriate entries:

```
<rel-IP_name_of_db><DBID><shm|tcp> on.. <rel-IP_name_db>
<socket>
```

in `sqlhosts.soc` with:

```
<local_hostname><DBID><shm|tcp> on.. <local_hostname>
<socket>
```

As user `<sid>adm` replace any occurrence of:

```
<rel-IP_name_of_db> or <rel-IP_name_of_ci>
```

with:

```
<local_hostname>
```

in:

```
/usr/sap/trans/bin/TPPARAM
/usr/sap/<SID>/SYS/profile/DEFAULT.PFL
/usr/sap/<SID>/SYS/profile/<SID>_DVEBMGS<INSTNR>
```

Check SAP WAS

To check SAP WAS:

Start all SAP WAS Instances as `<SID>adm`.

NOTE The automounter is still using the relocatable IP-addresses. You do not need to turn off the automounter. ServiceGuard and the packages remain running and provide all necessary resources.

- Import the profiles within SAP R/3. Transaction code: RZ10.
- Check SAP WAS setup. Ping(1M) to all application servers from the Central Instance and the other way round. Transaction code: SM59
- Check that batch jobs in SAP WAS are not scheduled to run on the relocatable IP address. Transaction code: SM37
- Relocate printers to the real hostname. Transaction code: SPAD
- Check operational modes within SAP WAS. You must setup new operation modes for the new hostname. Transaction code: RZ04
- Do all testing described in the document *SAP BC High Availability*.

Backup Files Again

After successfully finishing all tests, take another copy of the files you saved in the Sections “Additional steps for ORACLE” or “Additional steps for INFORMIX”. Refer to this step for a list of the files.

You previously renamed the onconfig file if you have an INFORMIX database. You might need this second backup for reference. If you plan to switch on the SGeSAP Integration again at a later point in time, make sure that no additional changes in the files occurred in the between times. Any changes would be lost if you replaced the altered files with the backup of the originals.

Use the backup created in this step, the backup of the altered files, later on for comparison with current active versions.

Switch On SGeSAP

Refer to the following sections for descriptions of how to switch on SGeSAP.

- Stop SAP WAS
- For ORACLE Database
- For INFORMIX Database
- Start SAP WAS
- Turn Off Debugging Mode
- Reconfigure SAP WAS

Stop SAP WAS

Stop the database and SAP WAS manually on all hosts in and outside of the cluster with:

```
stopsap all
```

For ORACLE Database

Restore Profiles:

- Check to see if the listener is running. Stop the listener process as ora<SID>:

  ```
  lsnrctl stop <LISTENERNAME>
  ```

- Restore all profiles with the versions that use relocatable addresses. You saved these versions when you switched off the SGeSAP. If appropriate, incorporate any changes that occurred to these files during the switch off process. You can use `diff` and the second backup you took to easily find out if anything changed.

- Restore the following files together with their destination paths:

  ```
  listener.ora
  /usr/sap/trans/bin/TPPARAM
  tnsnames.ora
  /usr/sap/<SID>/SYS/profile/DEFAULT.PFL
  /usr/sap/<SID>/SYS/profile/<SID>_DVEBMGS<INSTNR>
  ```

- Restart the listener as oracle user:

  ```
  lsnrctl start <LISTENERNAME>
  ```

For INFORMIX Database

Restore Profiles

— Restore all profiles with the versions that use relocatable addresses. You saved these versions when you switched off the SGeSAP. If appropriate, incorporate any changes that occurred to these files during the switch off process. You can use `diff` and the second backup you took to easily find out if anything changed.

— Restore the following files together with their destination paths:

```
/informix/<SID>/etc/onconfig.<rel-IP_name_of_db>.<DBID>
/informix/<SID>/etc/sqlhosts.soc/usr/sap/trans/bin/TPPA
RAM
/usr/sap/<SID>/SYS/profile/DEFAULT.PFL
/usr/sap/<SID>/SYS/profile/<SID>_DVEBMGS<INSTNR>
```

- Rename Customized Files

  As `informix` user perform the following from your home directory:

  ```
  rm  /informix/<SID>/etc/onconfig.<local_hostname>.<DBID>
  mv .customer.csh.off .customer.sh
  mv .customer.csh.off .customer.csh
  ```

  As `<sid>adm` in home directory:

  ```
  mv .customer.csh.off .customer.sh
  mv .customer.csh.off .customer.csh
  ```

Start SAP WAS

Start SAP R/3 as `<SID>adm` using:

```
startsap all
```

Turn Off Debugging Mode

Delete `/etc/cmcluster/debug` and/or `/etc/cmcluster/<SID>/debug` if they exist.

Reconfigure SAP WAS

To reconfigure SAP WAS:

- Import the profiles within SAP WAS. Transaction code: `RZ10`.
- Check SAP WAS setup. Try to `ping(1M)` to all application servers from the Central Instance and the other way round. Transaction code: `SM59`.
- Check that the batch jobs in SAP WAS are scheduled to run on the relocatable IP address not on the local hostname. Transaction code: SM36
- Relocate printers to the relocatable name. Transaction code: `SPAD`

- Check operational modes within SAP WAS. You must setup operation modes for the relocatable name.
- Do all the testing as described in the document *SAP High Availability*.

SGeSAP Administration Issues

This section describes the new aspects that a SGeSAP System Administrator of any cluster should be aware of.

Installation of the SGeSAP Integration Scripts significantly changes the hardware and software setup of your system. This affects the way you administer your SAP WAS. To get more detailed information on your specific setup, please refer to the documentation you receive from the person that installs the SGeSAP Integration on your system.

You no longer need to treat the Central Instance of your SAP R/3 system and its accompanying database Instance as though it runs on a dedicated host. With SAP WAS they are wrapped up inside one or more ServiceGuard packages and packages can be sent to any of the hosts that are inside of your ServiceGuard cluster.

This provides not only a mechanism to cope with hardware failures but also a new amount of flexibility and opportunities. If you have to maintain the host machine on which your SAP WAS is running, you can “send the running SAP WAS over” to another host. This causes a few minutes of downtime only, in contrast to the significantly higher amount downtime if you have to wait until the maintenance is completed before you can restart your SAP WAS Instances.

On the other hand you have to be more careful in changing the setup. This applies to hardware changes as well as software issues.

The balance of this section describes the following SGeSAP administration aspects:

- Hardware Aspects
- HP-UX Software Aspects
- SAP WAS Administration Aspects
- ServiceGuard Administration Aspects
- SGeSAP Administration Aspects

Hardware Aspects

If you add new hardware and SAP WAS needs access to it to work properly, make sure to allow this access from any host of the cluster by appropriately planning the connectivity. For example:

It is possible to increase database disk space by adding a new RAID-Array to the primary host on which your Database Instance normally runs.

Setup with shared disks, so the new RAID-Array is visible on the new host and does not create an error.

You can add the RAID-Array using the conventional method you are used to. But remember: The fact that your SAP R/3 system runs correctly after the changes does *not* imply that it will work after a switchover to a different host as well.

For the same reason a high available SAP WAS Spooling Service needs a connection to the printers from any host in the cluster.

If you do not feel comfortable in changing your hardware setup, please contact your HP-consultant.

HP-UX Software Aspects

Depending on your particular setup, you will have to deal with three important groups of directories that need special treatment:

- *common directories that are kept local on any node*

  Using a standard setup the following directories and their files are kept locally on any host of the cluster:

  `/etc/cmcluster`—the directory in which ServiceGuard keeps its configuration files

  `/home/<SID>adm`—the home directory of the SAP R/3 system Administrator.

  It is your responsibility to synchronize the contents of these directories on all hosts of the cluster. `/home/<SID>adm` does not need to be the same on all of the hosts. For example:

  It is possible to install an additional application server on a host of the cluster. The application server will not be part of any package. It is local to this host. The SAP startup scripts are only needed on this dedicated host. You do not need to distribute them to other hosts.

The standard HP-UX configuration files are local, too. Never delete the mutual `.rhosts` entries of the root user and `<SID>adm` on any of the nodes. Never change entries in `/etc/hosts`, `/etc/services`, `/etc/passwd` or `/etc/group` on only some of the nodes. Keep them unified.

If you use an ORACLE database, be aware that the listener configuration file of SQL*Net V2 is kept locally as `/etc/listener.ora` by default, too.

- *directories that reside on shared disks*

  Changing these files on any host of the cluster applies the change to the whole cluster.

  Files in the following directories and all subdirectories are typically shared:

  ```
  /usr/sap/<SID>/DVEBMGS<instance_id>
  /export/usr/sap/trans
  /export/sapmt/<SID>
  /export/informix or /oracle/<SID>
  ```

  They are only available on a host if the package they belong to is running on it. ServiceGuard switches them to another node with the package. If you use a two package concept all directories belong to the database package (db) apart from the first which belongs to the Central Instance package (ci). Please refer to the description of your particular setup to obtain a list of your shared directories.

- *directories that are treated by the automounter*

  These directories are mounted automatically as needed. This is true not only for the nodes of the cluster. If you use external application servers, they also use them.

  Automounter directories are:

  ```
  /sapmnt/<SID>
  /usr/sap/trans
  /informix
  ```

  They are NFS-mounted from their equivalents in the `/export` directory of the node(s) which run(s) the package(s). The automounter setup uses the relocatable IP addresses. The directories are soon available again after a switchover has taken place.

There are two important issues concerning these directories:

— The directories below /export are exported without root permissions.

This happens according to the recommendations of SAP WAS and enhances the security of the installation. The effect is, that the root user cannot modify these directories or their contents. With standard permissions set, the root user cannot even see the files. This is not an error and the system runs without problems. If you want to modify anything as root, please use the equivalent directory below /export on the host the package runs on.

— If the database package is halted, you cannot log in as <SID>adm unless you keep the binaries local.

The reason for this is, that /usr/sap/<SID>/SYS/exe is part of the path of <SID>adm. Without local binaries this directory links to /sapmnt/<SID> which in fact is handled by the automounter. The automounter cannot contact the host belonging to the relocatable address that is configured because the package is down. The system hangs. To avoid this, always keep local copies of the executables.

SAP WAS Administration Aspects

In regards to SAP WAS, most everything remains the same. The only difference is the use of the relocatable IP-addresses. During installation of the SGeSAP Integration, the profiles are changed to contain only relocatable IP-addresses for the database as well as the Central Instance. You can check this using transaction RZ10. In the DEFAULT.PFL the entries are altered:

```
SAPDBHOST= <relocatible_db_name>
rdisp/mshost= <relocatible_ci_name>
rdisp/vbname= <relocatible_ci_name>_<SID>_<nr>
rdisp/enqname= <relocatible_ci_name>_<SID>_<nr>
rdisp/btcname= <relocatible_ci_name>_<SID>_<nr>
rslg/collect_daemon/host = <relocatible_ci_name>
```

There are also two additional profile parameters SAPLOCALHOST and SAPLOCALHOSTFULL included as part of the Instance Profile of the Central Instance. Anywhere SAP WAS uses the local hostname

internally, this name is replaced by the relocatable values <*relocatible_ci_name*> or <*relocatible_ci_name*>.domain.organization of these parameters. Make sure that they are always defined and set to the correct value. This is vital for the system to function correctly.

Normally use the relocatable name of your Central Instance package if SAP WAS asks for the hostname of the machine the Central Instance runs on. Unfortunately, SAP WAS sometimes uses local addresses by itself. If the system is set up correctly as described in Chapter 2, “Step by Step Installation Guide,” you do not have to worry about this. The values change if you switch over to another host.

The destination for print formatting, which is done by a Spool Work process, uses the application server name. Use the relocatable name if you plan to use Spool Work processes with your Central Instance. In these cases no changes need to be done in case of a failover - printing will work persistently. Note that a print job in process at the time of the failure will be canceled and needs to be reissued manually after the failover. To make a spooler highly available on the Central Instance, set the destination of the printer to <*relocatible_ci_name*>_<SID>_<nr> using the transaction SPAD. Print all *time critical* documents via the high available spool server of the Central Instance. Print requests to other spool servers stay in the system after failure until the host is available again and the spool server has been restarted. These requests can be moved manually to other spool servers if the failed server is unavailable for a longer period of time.

Batch jobs can be scheduled to run on a particular instance. Generally, it is better not to specify a destination host at all. Sticking to this rule, the batch scheduler chooses a batch server which is available at the start time of the batch job. However, if you wish to specify a destination host, specify the batch server running on the highly available Central Instance. The application server name and the hostname (which is retrieved from the Message Server) are stored in the batch control tables TBTCO, TBTCS,.... In case the batch job is ready to run, the application server name is used to start it. Therefore, when using the relocatable name to build the application server name for the SAP WAS Instance, you do not need to change batch jobs which are tied to it after a switchover. This is true even if the hostname which is also stored in the above tables, differs.

Plan to use saplogon to application server groups instead of saptemu/sapgui to individual application servers. When logging on to an application server group with two or more application servers, the SAP

WAS user does not need a different login procedure if one of the application servers of the group fails. Also, using login groups, provides workload balancing between application servers, too.

Within the CCMS you can define operation modes for SAP WAS Instances. An operation mode defines a resource configuration. It can be used to determine which instances are started and stopped and how the individual services are allocated for each instance in the configuration. An instance definition for a particular operation mode consists of the number and types of Work processes as well as Start and Instance Profiles. When defining an instance for an operation mode you need to enter the hostname and the system number of the application server. By using relocatable names to fill in the hostname field, the instance will be working under control of the CCMS after a failover without a change. Note however, that if an instance is running on the standby node in normal operation and is stopped during the switchover, the control console will show the instance to be down afterwards.

Only configure the update service on a node for Application Services running on the same node. As a result, the remaining SAP WAS servers, running on different nodes, are not affected by the outage of the update server. However, if the update server is configured to be responsible for application servers running on different nodes, any failure of the update server leads to subsequent outages at these nodes. Configure the update server on the highly available Central Instance. Using local update servers should only be considered, if performance issues require it.

ServiceGuard Administration Aspects

ServiceGuard keeps information about the cluster configuration, it especially needs to know the relocatable IP addresses and its subnets, your Volume Groups, the Logical Volumes and their mountpoints. Check with your HP consultant for information about the way ServiceGuard is configured to suite your SAP R/3 system. If you touch this configuration, you have to reconfigure your cluster.

SGeSAP Administration Aspects

The SGeSAP needs some additional information about the configuration of your SAP R/3 system. It gathers this information from the file `/etc/cmcluster/sap.conf`. It is very important that the information in this file is consistent with the way your system is configured. This file must be available on all nodes of the cluster. Under normal operation

there is no need to touch this file. But it is a good idea for you as an administrator to have a look at it to prevent you from doing things that can cause the setup to be no longer able to switch. Comments are provided within the file and most things are self-explanatory if you are trained to understand shell scripts. For example, the following administration activities are possible but they must be accompanied by an adaptation of the `sap.conf` file on all cluster nodes:

- Changing the SAP System ID
- Changing the name of the SAP System Administrator
- Migrating to another database vendor
- Adding/deleting additional application servers
- Changing Instance Numbers
- Changing the name belonging to a relocatable address
- Changing the name of a SGeSAP-Integration package
- Changing hostnames of hosts inside the cluster
- Changing hostnames of hosts that run additional application servers
- Changing the location of any SAP-specific directory in the filesystem
- Changing the name of the ORACLE listener

After performing any of the above mentioned activities the ability to failover correctly has to be tested again. Be aware that changing the configuration of your SAP WAS cluster in any way can lead to the loss of warranty. Make sure to plan those steps with your HP consultant.

Index