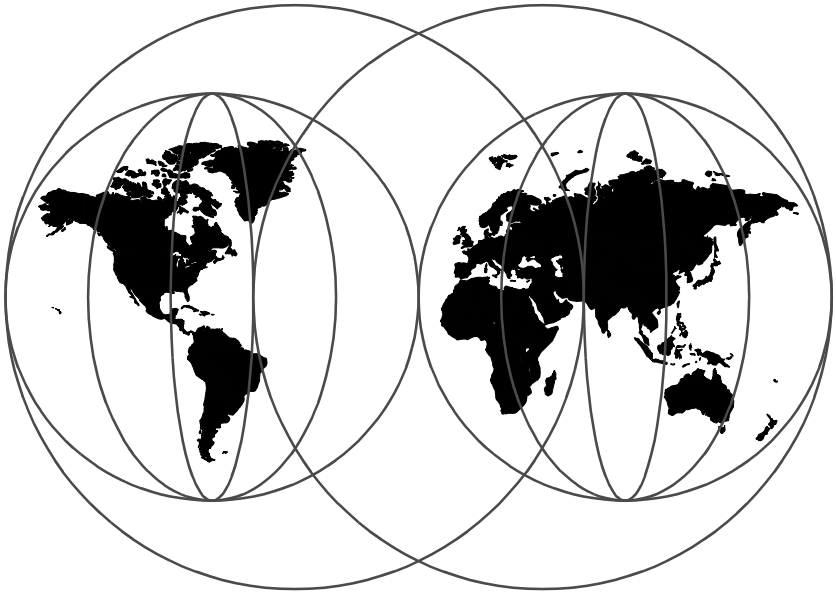
HACMP Enhanced Scalability Handbook

Yoshimichi Kosuge, Bernhard Buehler, Claudio Marcantoni, Hiroyuki Onodera, Alex Wood



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5328-00

SG24-5328-00

International Technical Support Organization

HACMP Enhanced Scalability Handbook

January 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special Notices” on page 229.

First Edition (January 1999)

This edition of HACMP Enhanced Scalability Handbook applies to the High Availability Cluster Multi-Processing for AIX Version 4.3 for use with the IBM Parallel System Support Programs for AIX Version 3.1 and the AIX Version 4.3.2. This book is based on a pre-release version of a product and may not apply when the product becomes generally available.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

Contents

Figures

Tables

Preface

The RS/6000 SP High Availability Infrastructure (HAI) was introduced with IBM Parallel System Support Programs for AIX (PSSP) Version 2 Release 2. IBM High Availability Cluster Multi-Processing for AIX Enhanced Scalability (HACMP/ES) Version 4 Release 2 was introduced with PSSP Version 2 Release 3 to exploit HAI. Now HAI has a new name, RS/6000 Cluster Technology (RSCT), and is available for the whole RS/6000 family, not only the RS/6000 SP systems. This means HACMP/ES is also available for the whole RS/6000 family.

This redbook describes how HACMP/ES components have changed to support the RS/6000 family. It introduces the useful log files and commands you need when you have trouble with HACMP/ES. Once you understand the log files and commands, this book shows you practical usage of them. Finally, this book discusses considerations of resource management.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Yoshimichi Kosuge is an IBM RS/6000 SP project leader at the International Technical Support Organization, Poughkeepsie Center. He joined IBM Japan in 1982 and has worked in wide areas: LSI design, S/390 CP microcode, VM, MVS, OS/2, and AIX. Since 1998, he writes redbooks and teaches IBM classes worldwide on all areas of RS/6000 SP.

Bernhard Buehler is an instructor at the IBM Education and Training Center (E&T) in Herrenberg, Germany. Before joining E&T he worked as an HACMP specialist at the IBM RS/6000 and AIX Center of Competence, IBM Germany. He has worked at IBM for 17 years, and has eight years of experience in the AIX field. His areas of expertise include AIX, HACMP, RS/6000 SP, and HAGEO. He is a co-author of the redbooks *DataJoiner Implementation and Usage Guide*, *Enterprise-Wide Security Architecture and Solutions Presentation Guide*, and *HACMP Enhanced Scalability*.

Claudio Marcantoni is an instructor at the IBM Education Center of Novedrate, Italy, where he teaches different courses on HACMP for AIX and HACMP/ES. He has eight years of experience in the AIX field. He has worked at IBM for nine years. His areas of expertise include AIX, Networking and HACMP.

Hiroyuki Onodera is an I/T Specialist at the IBM Japan Systems Engineering Co., Ltd. in Makuhari, Japan. He has worked at IBM for nine years, and has six years of experience in the AIX high availability field.

Alex Wood is a systems support specialist at the AIX Support Centre in Basingstoke, United Kingdom. He has been working in IBM for four years with AIX and RS/6000 and is currently the technical advisor for both the HACMP and RS/6000 SP teams. He is a co-author of the redbook *RS/6000 SP: PSSP 2.2 Survival Guide*.

Thanks to the following people for their invaluable contributions to this project:

Michael Coffey
Peter Badovinatz
IBM Laboratory Poughkeepsie

David Thiessen
IBM ITSO Austin

John Easton
Simon Marchese
IBM High Availability Cluster Competency Centre (UK)

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 245 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com`
  For IBM Intranet users `http://w3.itso.ibm.com`

- Send us a note at the following address:

  `redbook@us.ibm.com`

Chapter 1. Introduction

IBM High Availability Cluster Multi-Processing for AIX Enhanced Scalability (HACMP/ES) was developed to exploit RS/6000 SP High Availability Infrastructure (HAI) of IBM Parallel System Support Programs for AIX (PSSP), when HACMP/ES was introduced. At this point, HACMP/ES could run only on the IBM RS/6000 SP systems. One of the enhancements for HACMP/ES V4.3 removes this limitation, so that HACMP/ES now runs on any member of the RS/6000 family.

This chapter introduces new functions in HACMP/ES V4.3.

1.1 Terminologies

IBM High Availability Cluster Multi-Processing for AIX (HACMP) has a long history. Throughout this history, there have come to be many variations of HACMP. To avoid confusion, we first define some terminologies. These definitions are intended only for the discussion in this book.

HACMP Used for all HACMP LPPs regardless of version.

HACMP/ES Used for HACMP LPPs that use IBM RS/6000 Cluster Technology (RSCT) or High Availability Infrastructure (HAI) technology. It is used for all versions.

HACMP for AIX Used for HACMP LPPs that do not use RSCT or HAI technology. It is used for all versions, which are sometimes also referred to as HACMP Classic.

HACMP Vx.x Used for both HACMP/ES Vx.x and HACMP for AIX Vx.x. For example, HACMP V4.3 stands for both HACMP/ES V4.3 and HACMP for AIX V4.3.

Hardware definitions are required, too. In discussions of HACMP, it is necessary to distinguish on which hardware HACMP runs. This book uses the following terminologies:

RS/6000 Family This includes all the hardware using a name brand IBM RS/6000.

RS/6000 SP node This stands for an RS/6000 SP node. For example, 332MHz SMP wide node, 332MHZ SMP thin node, and so on.

RS/6000 This stands for any machine of the RS/6000 Family except RS/6000 SP. For example, model 140, F50, H50, and so on.

RS/6000 S70 can mean RS/6000 SP node or RS/6000. It depends on how it is configured. If it is configured as an independent node of the SP system, it is RS/6000 SP node. If not, it is RS/6000.

1.2 HACMP/ES V4.3

This section discusses the new or enhanced functions provided with Version 4.3 of the HACMP/ES product, as follows:

- Support for the RS/6000 Family
- 32-node support
- Topology Dynamic Automatic Reconfiguration Event (DARE)
- Packaging
- Concurrent access
- Improved snapshot facility
- ATM support
- SDR independency
- Heartbeat rate tunable on a network basis
- Multiple pre/post events for each cluster event
- Task Guide for creating a shared volume groups
- Global networks

Each topic is covered in detail in the following sections.

Support for the RS/6000 Family
One major limitation of HACMP/ES V4.2.2 is that it is only supported on the RS/6000 SP node. HACMP/ES V4.3 can now be installed on both RS/6000 SP nodes and RS/6000s. This means it is now possible to have an HACMP/ES V4.3 cluster made up of:

- RS/6000s only
- RS/6000 SP nodes only
- A mixture of RS/6000s and RS/6000 SP nodes
- RS/6000 SP nodes belonging to different RS/6000 SP systems

In case of an HACMP/ES V4.3 cluster composed of RS/6000 SP nodes only, these nodes can belong to different PSSP partitions.

32-Node Support

HACMP/ES V4.2.2 supports a maximum cluster size of 16 RS/6000 SP nodes without PTF, while HACMP/ES V4.3 supports clusters of up to 32 RS/6000 SP nodes.

Like all previous versions, HACMP for AIX V4.3 is still limited 8-node clusters.

Topology DARE

HACMP/ES V4.3 now supports DARE of the Cluster Topology. DARE allows the system administrator to perform changes to the active cluster configuration without having to stop and restart HACMP/ES, hence increasing the availability of customer applications and reducing downtime.

HACMP/ES V4.2.2 supports DARE only of the Cluster Resources, not of the Cluster Topology.

Packaging

In order to install the HACMP/ES V4.3 software, it is now required to install the IBM RS/6000 Cluster Technology (RSCT) filesets first. These filesets include the support on the Topology Services, Group Services and Event Management components. Before V4.3, these components were part of PSSP. Now, having extended the support of HACMP/ES to the RS/6000 Family, these components are part of a new package called RSCT, which no longer depends on PSSP. The fileset names are rsct.basic and rsct.clients.

For more details on packaging, refer to 2.1.3.4, “RS/6000 Cluster Technology Filesets” on page 13.

Concurrent Access

HACMP/ES V4.3 now supports concurrent access clusters, hence offering the opportunity to use the IBM-provided lock manager, the cllockd daemon. This means all the cluster nodes are able to read/write the data on the external shared disks concurrently.

Improved Snapshot Facility

HACMP/ES V4.3 implements an improved snapshot facility that preserves user-defined events. When applying a cluster snapshot, user-defined events are restored in the /usr/sbin/cluster/events/rules.hacmprd file.

With HACMP/ES V4.2.2 it is a customer’s responsibility to save user-defined events when taking a snapshot and recreate them when applying the snapshot.

Also HACMP/ES V4.3 allows you to define your own commands to those that are issued by the standard snapshot. This allows you to add extra information, such as application related data, to your snapshot.

ATM Support
HACMP/ES V4.3 now supports Asynchronous Transfer Mode (ATM) networks.

SDR Independency
HACMP/ES V4.3 no longer has a dependency on the System Data Repository (SDR). With HACMP/ES V4.2.2, the daemons used to query the SDR to collect information about the cluster configuration, hence creating a dependency on the Control Workstation (CWS) where the SDR resides. Now the HACMP/ES V4.3 daemons gather the necessary information from the Global ODM (GODM). This change became necessary since HACMP/ES V4.3 can now also run on a cluster composed of only RS/6000s, while the SDR is a peculiarity of the RS/6000 SP system.

For more details on GODM, refer to 3.1, “Global Object Data Manager” on page 19.

Heartbeat Rate Tunable on a Network Basis
HACMP/ES V4.3 now allows the system administrator to configure a different heartbeat rate for every network type, hence providing the same granularity the HACMP for AIX product has. HACMP/ES V4.2.2 only allows one heartbeat rate that is valid for all networks.

For more details, refer to “HACMPnim” on page 22.

1.3 IBM RS/6000 Cluster Technology

The RS/6000 SP High Availability Infrastructure (HAI) was introduced as part of IBM Parallel System Support Programs for AIX (PSSP) Version 2.2. With PSSP Version 3.1, the term RS/6000 SP High Availability Infrastructure has been replaced by the newly-minted official term IBM RS/6000 Cluster Technology (RSCT) which refers to the following three key distributed subsystem components:

- Topology Services
- Group Services
- Event Management

If you install HACMP/ES on an RS/6000 SP node, you need AIX, PSSP, and HACMP/ES. If you install HACMP/ES on an RS/6000, you need AIX and

HACMP/ES. RSCT is included in both PSSP V3.1 and HACMP/ES V4.3. For more details on packaging, refer to Chapter 2., “Installation and Migration” on page 7.

Cluster

The term *cluster* is used to refer to a collection of RS/6000 Family nodes on which the RSCT components are executing. These machines may exclusively be RS/6000 SP nodes, RS/6000s, or a combination of both.

This is a significant change of the cluster concept. A major goal of RSCT for HACMP/ES V4.3 is that it is able to support HACMP/ES on the same sets of arbitrary RS/6000 Family nodes as does HACMP for AIX today. This requires that the RSCT components likewise support these arbitrary clusters.

Note that a cluster may not be exclusive. An RS/6000 Family may be contained in multiple clusters. An example would be an RS/6000 SP node that has HACMP/ES installed. Within the PSSP, the node is part of the *PSSP cluster*, but it is also part of the *HACMP cluster*. Each cluster is independent of the other, and the subsystems within each are independent.

Domain

The term *domain* is subtly more specific than cluster. A domain describes the boundary of a set of RS/6000 Family within which the executing RSCT components provide their services. In general, the boundaries of a cluster match the boundaries of a domain. The differentiation is that a cluster does not become a domain until Group Services has established its domain via one of the Group Services daemons on a node within the cluster. At this point, the clients of Group Services are allowed to form their groups and begin offering their services.

As with a cluster, a domain may overlap with another domain, in other words, it is possible for an RS/6000 Family nodes to participate in multiple domains, each such overlapping domain remaining independent and ignorant of the other. The expected domains are:

- PSSP domain

  The boundaries of such a domain are the Control Workstation (CWS) and the nodes within an SP partition. Thus each SP partition is a separate domain.

- HACMP domain

  The boundaries of such a domain differ based on the release:

  - Before HACMP/ES V4.3, the domain was a proper subset of a PSSP domain.

- With HACMP/ES V4.3, the domain is any allowable cluster.

Realm

The term *realm* is used to refer to the different domains previously discussed, but in general terms rather than when discussing a specific domain. Thus, the term *SP realm* refers to any domain established on any generic RS/6000 SP nodes to support the PSSP. The term *HA realm* refers to any domain established on any generic cluster to support HACMP/ES.

Dual Daemons

The term *dual daemons* represents the fact that there are separate sets of RSCT components supporting the separate domains described above. Thus, on hardware that is part of both the SP realm and the HA realm, each node will execute:

- A Topology Services daemon for each realm.
- A Group Services daemon for each realm.
- An Event Management daemon for each realm.

Chapter 2. Installation and Migration

This section offers a brief description of the installation steps for HACMP/ES. For detailed information, refer to *HACMP for AIX, Version 4.3: Enhanced Scalability Installation and Administration Guide*, SC23-4284, as well as the release notes.

2.1 Prerequisites

We first provide a brief overview of the hardware and software prerequisites related to HACMP/ES V4.3. Requirements may change for future versions.

2.1.1 Hardware Prerequisites

The hardware requirements are basically the same as those for HACMP for AIX, such as a non-IP network, more than one adapter per network, and so on.

There is no difference between the hardware requirements for HACMP for AIX and HACMP/ES. Both versions can run on the same RS/6000 Family (see “Support for the RS/6000 Family” on page 2). Because of this, HACMP/ES is no longer SP partition-bounded. It is now possible to build the same kind of HACMP clusters. It may, however, happen that some hardware combinations are not yet supported by HACMP/ES V4.3. For the latest information, ask your IBM representative.

2.1.2 Software Prerequisites

HACMP/ES V4.3 has the following prerequisites:

- Each cluster node must have AIX 4.3.2 installed.
- Each cluster node must have IBM RS/6000 Cluster Technology (RSCT) V1.1 installed. The filesets are:
  - rsct.basic.rte
  - rsct.clients.rte

  For a more detailed list about RSCT, see 2.1.3.4, “RS/6000 Cluster Technology Filesets” on page 13.
- If using an RS/6000 SP node, Version 3.1 must be installed on the CWS and the RS/6000 SP nodes.
- The following optional AIX bos components are mandatory for HACMP/ES V4.3:

- bos.rte
- bos.net.tcp.server
- bos.net.tcp.client
- bos.adt.lib
- bos.adt.libm
- bos.adt.syscalls
- bos.rte.SRC
- bos.rte.libc
- bos.rte.libcfg
- bos.rte.libcur
- bos.rte.libpthreads
- bos.rte.odm
- bos.rte.lvm
- perfagent.tools

- Each cluster node requires its own HACMP/ES software license.
- The /usr file system must have a minimum of 15MB of free space (or the volume group must have enough space to extend it by 15MB).
- The / (root) file system must have a minimum of 1MB of free space (or the volume group must have enough space to extend it).
- The installation process must be performed by the root user.

2.1.3 Software Installation and Configuration

Before you start the installation and configuration of your HACMP/ES cluster, the following conditions should be met:

- The planning should be finished.
- The necessary documentation (planning worksheets) should be available.
- The hardware should be connected.

2.1.3.1 Installation

The steps you need to install the HACMP/ES code are the same as for the installation of HACMP for AIX V4.2.2. The only difference is that you have to select the HACMP/ES LPP Package filesets and, based on these filesets, different prerequisites. The HACMP/ES V4.3 packages are listed in 2.1.3.3, “HACMP/ES V4.3 Packages” on page 12.

2.1.3.2 Configuration

The configuration is also the same as for HACMP for AIX V4.2.2. There are some new options or SMIT screens, which we describe here. To understand these descriptions, it is necessary to have a good understanding of HACMP. knowledge. The differences we discuss here are related to the HACMP.

Cluster Configuration

There is a new SMIT choice, Cluster Security, added to the Cluster Configuration screen. This function has been available since HACMP Version 4.2.1. It was in the 4.2.1 and 4.2.2 versions as Cluster Security Mode in the Change/Show Run Time Parameter screen. Figure 1 on page 9 shows the new screen.

```
 Cluster Configuration

Move cursor to desired item and press Enter.

  Cluster Topology
  Cluster Security
  Cluster Resources
  Cluster Snapshots
  Cluster Verification
  Cluster Custom Modification
  Restore System Default Configuration from Active Configuration




F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 1. HACMP/ES V4.3 Cluster Configuration SMIT Screen

Cluster Topology

The parts we discuss here are shown in Figure 2 on page 10 in bold font.

The Cluster Topology SMIT screen has two new lines: Configure Global Networks and Configure Network Modules, and one modified line: Configure Topology Services and Group Services.

- Configure Global Networks

  This is new. It is used for heartbeat across separate logical subnetworks. For more information, see 4.2.2.5, “claddnetwork” on page 63, and for an example see 6.1, “Global Network” on page 131

- Configure Network Modules

  This is not part of HACMP/ES V4.2.x, but you may know it from HACMP for AIX.

- Configure Topology Services and Group Services

  This was already part of HACMP/ES V4.2.x. When it was HACMP/ES V4.2.x, it was named Configure Network Modules. Some parts in this

subscreen have changed and some functions are moved to Configure Network Modules.

Note: It is no longer required that the nodename has a matching IP-label or alias.

```
 Cluster Topology

Move cursor to desired item and press Enter.

  Configure Cluster
  Configure Nodes
  Configure Adapters
  Configure Global Networks
  Configure Network Modules
  Configure Topology Services and Group Services
  Show Cluster Topology
  Synchronize Cluster Topology



F1=Help          F2=Refresh       F3=Cancel        F8=Image
F9=Shell         F10=Exit         Enter=Do
```

Figure 2. HACMP/ES V4.3 Cluster Topology SMIT Screen

Cluster Resources

There are no major changes to the Cluster Resources screen in SMIT. *Change/Show Cluster Events* is changed to *Cluster Events, as* shown in Figure 3 on page 11. Two subscreens have changed:

```
 Cluster Resources

Move cursor to desired item and press Enter.

  Define Resource Groups
  Define Application Servers
  Change/Show Resources for a Resource Group
  Change/Show Run Time Parameters
  Cluster Events
  Change/Show Cluster Lock Manager Resource Allocation
  Show Cluster Resources
  Synchronize Cluster Resources



F1=Help          F2=Refresh          F3=Cancel          F8=Image
F9=Shell         F10=Exit            Enter=Do
```

Figure 3. HACMP/ES V4.3 Cluster Resources SMIT Screen

- Change/Show Run Time Parameters

  In this screen, the selection of the Cluster Security Mode is moved as described in, “Cluster Configuration” on page 9.

- Cluster Events

  A larger change was made to the *Cluster Events* screen. In addition to the name change from *Change/Show Cluster Events,* there are now two subscreens. Figure 4 on page 11 shows this new screen.

```
 Cluster Events

Move cursor to desired item and press Enter.

  Change/Show Cluster Events
  Define Custom Cluster Events



F1=Help          F2=Refresh          F3=Cancel          F8=Image
F9=Shell         F10=Exit            Enter=Do
```

Figure 4. HACMP/ES V4.3 Cluster Events SMIT Screen

- Change/Show Cluster Events

  In this screen you find basically the same functionality as in HACMP V4.2.2. It is merely one SMIT screen level lower. The more important

change is that you now can define more than one Pre or Post Event. For these events you have to select (use) the names you defined in the *Define Custom Cluster Events* screen.

- Define Custom Cluster Events

  This function is used to define Pre or Post Events. You have to specify a name for the event, the full path information, and a description. The event name you choose here is used in the *Change/Show Cluster Events* part. Figure 5 on page 12 shows what you will see if you select *Add a Custom Cluster Event* in the Define Custom Cluster Events screen.

  The command for this is *claddcustom*. For more details about this command, see 4.2.2.9, “claddcustom” on page 65.

```
 Add a Custom Cluster Event

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Cluster Event Name                                 []
* Cluster Event Description                          []
* Cluster Event Script Filename                      []




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5. HACMP/ES V4.3 Add a Custom Cluster Event SMIT Screen

2.1.3.3 HACMP/ES V4.3 Packages

The HACMP/ES V4.3 product has twelve packages. The following four packages belong to HACMP/ES only:

- cluster.es
- cluster.adt.es
- cluster.vsm.es
- cluster.msg.en_US.es

The following eight packages are the same for HACMP/ES V4.3 and HACMP for AIX V4.3:

- cluster.cspoc
- cluster.haview
- cluster.clvm

- cluster.taskguides
- cluster.msg.en_US.cspoc
- cluster.msg.en_US.haview
- cluster.man.en_US
- cluster.man.en_US.haview

2.1.3.4 RS/6000 Cluster Technology Filesets

The RSCT product consists of the following two packages. If one of the filesets is selected, both packages and all filesets are installed.

- The rsct.basic package

  The rsct.basic package includes the availability infrastructure provided by PSSP. The infrastructure includes Topology Services, Group Services, and Event Management.

  This package has the following filesets:

  - rsct.basic.rte
  - rsct.basic.hacmp
  - rsct.basic.sp

- The rsct.clients package

  The rsct.client package includes client services for the availability infrastructure provided by PSSP. (The infrastructure includes Topology Services, Group Services, and Event Management).

  This package has the following filesets:

  - rsct.clients.rte
  - rsct.clients.hacmp
  - rsct.clients.sp

2.2 Migration

In *HACMP for AIX, Version 4.3: Enhanced Scalability Installation and Administration Guide*, SC23-4284, you find two different migration paths. The first path is from HACMP for AIX to HACMP/ES V4.3. The second path is from an earlier HACMP/ES version to HACMP/ES V4.3.

The first two sections provide overviews of these two migration paths. The third and fourth sections contain additional information you need in case of configuration changes and fallback situations. The last section describes how to do a node-by-node migration from HACMP for AIX to HACMP/ES V4.3.

2.2.1 Migration from HACMP for AIX

HACMP for AIX and HACMP/ES currently do not support coexistence within the same cluster and system at the same time. Therefore, to be able to migrate from HACMP for AIX to HACMP/ES V4.3, you have to do the following steps:

1. Take a snapshot of the existing HACMP configuration.
2. Bring HACMP down on all the cluster nodes.
3. Uninstall HACMP for AIX on all the cluster nodes.
4. Install HACMP/ES V4.3 on all the cluster nodes.
5. Restore the cluster configuration from the snapshot taken in step 1.
6. Synchronize the cluster configuration.
7. Start IBM HACMP/ES.
8. Test the installation.

The snapshot is also useful if you ever wish to return to HACMP for AIX. Therefore, we recommend that you keep the snapshot taken in step 1 on page 14 for migration until the option of falling back to HACMP for AIX is no longer necessary.

2.2.2 Migration from HACMP/ES V4.2.x

HACMP/ES V4.3 can be installed within clusters of HACMP/ES V4.2.x, one node at a time. Therefore, current HACMP/ES sites can perform a rolling upgrade of their cluster to the newest one.

Note: None of the new functions are available until all nodes in this cluster are migrated.

To migrate the existing HACMP/ES version to HACMP/ES V4.3, do the following steps:

1. Take a snapshot of the cluster configuration.
2. Stop HACMP on the migrating node.
3. Install HACMP/ES V4.3.
4. Start HACMP/ES V4.3 on this node.
5. Test the application.
6. Repeat steps 2 through 5 for every node in the cluster.

2.2.3 Configuration Changes during Migration

This section describes what you have to care about if you have to make configuration changes during a migration of an existing HACMP/ES installation.

Note: This can only be done when migrating an existing HACMP/ES installation.

2.2.3.1 Dynamic Configuration Changes during Migration

Dynamic configuration changes are not supported in a mixed HACMP/ES cluster. The cldare program checks whether you are in a mixed environment.

2.2.3.2 Static Configuration Changes during Migration

To perform a static configuration change in a mixed HACMP/ES cluster, do the following steps:

1. Stop HACMP on all nodes in this cluster.
2. From a node with HACMP/ES V4.2.x make the necessary configuration changes.
3. Synchronize the cluster configuration.
4. Run the clconvert utility on all nodes with HACMP/ES V4.3.
5. Start HACMP on all nodes in this cluster.

2.2.4 Fallback to Previous Version

This section describes what you have to do in case of a fallback situation.

2.2.4.1 Fallback to HACMP for AIX

If you wish to, or have to, return to HACMP for AIX after the migration to HACMP/ES, the following steps should be used:

1. Stop HACMP on all nodes in this cluster.
2. Uninstall HACMP/ES on all nodes in this cluster.
3. Use the snapshot of HACMP for AIX you used for migration. Remember that any changes made after migration get lost.
4. Synchronize the cluster configuration.
5. Start the HACMP for AIX cluster.
6. Test the installation.

2.2.4.2 Fallback to an Earlier HACMP/ES Version

If you wish to, or have to, return to the HACMP/ES version from before you started the migration, the required steps depend on the number of remaining nodes that have the old HACMP/ES version installed. There are two possibilities: all nodes are already migrated, or there is at least one node with the old HACMP/ES code available.

One Node Still Using the Previous HACMP/ES Version

These steps are valid for all situations where you have at least one or more nodes:

1. Stop HACMP on all nodes where HACMP/ES V4.3 is installed.
2. Uninstall HACMP/ES V4.3 on all these nodes.
3. Install the previous version of HACMP/ES.
4. Stop HACMP on the remaining nodes.
5. Do a cluster synchronization from the remaining nodes.
6. Start HACMP on all nodes.

All Nodes using HACMP/ES V4.3

These steps are for the case when you have already migrated all nodes to HACMP/ES V4.3:

1. Stop HACMP on all nodes.
2. Uninstall HACMP/ES V4.3 on all nodes.
3. Install the previous version of HACMP/ES.
4. Apply the snapshot taken before starting the migration (see step 1 in 2.2.2, “Migration from HACMP/ES V4.2.x” on page 14).
5. Synchronize the cluster configuration.
6. Start HACMP on all nodes.

2.2.5 Node-by-Node Migration

Here we show you how to do a node-by-node migration from HACMP for AIX to HACMP/ES V4.3. The procedure is dependent on some new functions of HACMP/ES V4.3. For more information about the new features in HACMP/ES V4.3, see 1.2, “HACMP/ES V4.3” on page 2. This procedure may become obsolete in a future release of HACMP/ES.

This description is based on the fact that with HACMP/ES V4.3 it is now possible to do DARE for the HACMP Topology information (also see “Topology DARE” on page 3). The basic idea for this kind of migration is to

have two HACMP clusters, one with the old HACMP for AIX code and one with the new HACMP/ES V4.3 code installed.

2.2.5.1 General Considerations

As already mentioned, the examples in this chapter are based on DARE and create two independent HACMP clusters. The main problem you run into is that a network address can only exist once on the network. Therefore, you have to plan carefully for this kind of migration. The goal of the examples in the following two sections is to keep the downtimes very low. Depending on your actual installation, another sequence may give you shorter downtimes.

2.2.5.2 Having a Standby System

The steps described here are related to a two-node cluster, by having a standby node and using cascading or rotating. For rotating you can skip step1 because you have no real primary node. In this case, primary node means the first node you are going to migrate.

1. Start a takeover from the primary system to the standby system.
2. Uninstall HACMP for AIX on the primary system.
3. Install HACMP/ES V4.3.
4. Configure Topology on HACMP/ES V4.3 for one node cluster on the primary system (without synchronization).

   Note: Do not forget to choose a new HACMP cluster ID.
5. Configure Resources for this one-node cluster (without synchronization).
6. Stop HACMP on the standby system.
7. Synchronize Topology.
8. Synchronize Resources.
9. Start HACMP on the primary node.
10. Uninstall HACMP for AIX on the standby system.
11. Install HACMP/ES V4.3 on the standby system.
12. Configure the standby system to your HACMP/ES V4.3 cluster (Topology and Resources).
13. Synchronize Topology and Resources.

2.2.5.3 Using Mutual Takeover

The steps described here are related to a two-node cluster, by having a mutual takeover relationship. Before you start you have to decide which of these two nodes is more important for your environment. This is due to the

fact that the downtime for the second system is longer than the downtime for the first system.

1. Start a takeover from the more important primary system to its standby system.
2. Uninstall HACMP for AIX on this system.
3. Install HACMP/ES V4.3.
4. Configure Topology on HACMP/ES V4.3 for one node cluster on the primary system (without synchronization).

   Note: Do not forget to choose a new HACMP cluster ID.
5. Configure Resources for this one-node cluster (without synchronization).
6. Go to the backup system.
7. Remove the resource group in the old HACMP cluster for the node you stopped.
8. Synchronize Resources on the old HACMP cluster.
9. Go back to the HACMP/ES V4.3 node.
10. Synchronize Topology on this node.
11. Synchronize Resources on this node.
12. Start HACMP on this node.
13. Add the Topology information for the second node to the HACMP/ES V4.3 cluster (without synchronization).
14. Add the Resource information for the second node to the HACMP/ES V4.3 cluster (without synchronization).
15. Go to the remaining HACMP for AIX node.
16. Stop HACMP on this node (without takeover).
17. Uninstall HACMP for AIX on this system.
18. Install HACMP/ES V4.3.
19. Go back to the HACMP/ES V4.3 node.
20. Synchronize Topology.
21. Synchronize Resources.
22. Start HACMP on the second node.

Chapter 3. Component Design

This chapter discusses the components of IBM HACMP Enhanced Scalability (HACMP/ES) and IBM RS/6000 Cluster Technology (RSCT).

3.1 Global Object Data Manager

This section explains the HACMP/ES V4.3 Global Object Data Manager (GODM) attributes and their changes.

The GODM consists of the HACMP classes in the Object Data Manager (ODM), and is maintained by HACMP across the nodes in a cluster.

3.1.1 System Data Repository and GODM

HACMP/ES V4.2.x, which includes private Topology Services and Group Services, uses two types of System Data Repository (SDR) information: the SP Switch (SPS) information, and the node ID information.

In order to provide RS/6000 support (refer to “Support for the RS/6000 Family” on page 2), HACMP/ES V4.3 and the RSCT components running on the HACMP domain have been changed and do not use SDR information. That is, the RSCT components running on HACMP domain use only GODM.

Basically, GODM has enough information about topology, like nodes and IP address relationships. While the previous version of HACMP/ES GODM used SDR node ID information, HACMP/ES V4.3 GODM has been changed not to depend on the SDR node ID information.

The RSCT components running on the PSSP domain use SDR as before.

3.1.2 Structure of the GODM Class

HACMP/ES stores information into GODM class files in the /etc/es/objrepos directory, and symbolic links to these ODM class files are stored in the /etc/objrepos directory. These GODM class files are ODM class files that are maintained across the nodes in a cluster. HACMP usually has copies of these GODM class files in the /usr/sbin/cluster/etc/objrepos/active directory (for the active HACMP configuration), and in the /usr/sbin/cluster/etc/objrepos/stage directory (for temporary purposes). Use the `odmget` command to get the GODM class attributes. If you want to refer to the active GODM class attributes, you must change the ODMDIR environment variable from default /etc/objrepos to /usr/sbin/cluster/etc/objrepos/active. You can use the `env` command to make a temporary change to the ODMDIR environment variable.

3.1.2.1 Topology-Related GODM Classes

Topology Services running on the HACMP domain and uses the following GODM classes, which contain topology-related information:

- HACMPcluster
- HACMPnode
- HACMPnetwork
- HACMPadapter
- HACMPnim
- HACMPtopsvcs

These GODM classes were changed in HACMP/ES V4.2.2.

The information contained in these GODM classes is explained in the following examples.

HACMPcluster

```
# odmget HACMPcluster

HACMPcluster:
        id = 2
        name = "clusterb"
        nodename = "risc71"
        sec_level = "Standard"
        last_node_ids = ""
        highest_node_id = 0
        last_network_ids = "1"
        highest_network_id = 0
        handle = 3
        cluster_version = 1
```

The HACMPcluster GODM class contains information about the cluster ID (id), cluster name (name), local node name (nodename), security level (sec_level: use normal rsh or Kerberos), local node handle (handle), cluster version (cluster_version: HACMP version), and so forth. The handle and cluster_version information represent new attributes. Refer to “handle (node_handle)” on page 25 and “cluster_version (version)” on page 26 for further details. The id and name attributes are set by the SMIT Add a Cluster Definition menu. sec_revel is set by the SMIT Change/Show Cluster Security menu. Other attributes are set by HACMP automatically.

HACMPnode

```
# odmget -q name=risc71 HACMPnode

HACMPnode:
        name = "risc71"
        object = "VERBOSE_LOGGING"
        value = "high"
        node_id = 3
        node_handle = 3
        version = 1

HACMPnode:
        name = "risc71"
        object = "NAME_SERVER"
        value = "false"
        node_id = 3
        node_handle = 3
        version = 1
```

The HACMPnode GODM class contains information about all node names (name), all node IDs (node_id), all node handles (node_handle), and so forth. The node_handle and version information represent new attributes. Refer to “handle (node_handle)” on page 25 and “cluster_version (version)” on page 26 for further details. The name attribute is set by the SMIT Add Cluster Nodes menu, and object and value are set by the SMIT Configure Run Time Parameters menu.

HACMPnetwork

```
# odmget -q name=ethernet2 HACMPnetwork

HACMPnetwork:
        name = "ethernet2"
        attr = "public"
        network_id = 0
        globalname = ""
```

The HACMPnetwork GODM class contains information about all network names (name), network attributes (attr), the global network name (globalname), and so forth. The globalname represents a new attribute. Refer to “globalname” on page 27 for further details. The name and attr attribute are set by the SMIT Add an Adapter menu, and the globalname attribute is set by the SMIT Change/Show a Global Network menu. For more about the global network configuration, refer to 6.1, “Global Network” on page 131.

HACMPadapter

```
# odmget -q ip_label=risc71_boot HACMPadapter

HACMPadapter:
        type = "ether"
        network = "ethernet2"
        nodename = "risc71"
        ip_label = "risc71_boot"
        function = "boot"
        identifier = "80.7.6.20"
        haddr = ""
        interfacename = "en0"
```

The HACMPadapter GODM class contains information about all adapter names and attributes such as network type (type), network name (network), node name (nodename), ip label name (ip_label), adapter function (function: boot, service, or standby), identifier (identifier: usually IP address without the serial network case), adapter hardware address (haddr), and network interface name (interfacename). The network interface represents a new attribute. Refer to “interfacename” on page 27 for further details. These attributes are set by the SMIT Add an Adapter menu.

HACMPnim

```
# odmget -q name=ether HACMPnim

HACMPnim:
        name = "ether"
        desc = "Ethernet Protocol"
        addrtype = 0
        path = "/usr/sbin/cluster/nims/nim_ether"
        para = ""
        grace = 30
        hbrate = 500000
        cycle = 4
```

The HACMPnim GODM class contains information about all network types (name), heartbeat rates (hbrate), and so forth. HACMP/ES V4.2 does not have the HACMPnim GODM class and cannot set heartbeat rates for each network type. In contrast, HACMP/ES V4.3 does have the HACMPnim GODM class, in order to set up the heartbeat rate by network types. These attributes are set by the SMIT Change/Show a Cluster Network Module menu.

HACMPtopsvcs

```
# odmget HACMPtopsvcs

HACMPtopsvcs:
        hbInterval = 1
        fibrillateCount = 4
        runFixedPri = 1
        fixedPriLevel = 38
        tsLogLength = 5000
        gsLogLength = 5000
        instanceNum = 18
```

The HACMPtopsvcs GODM contains information about the default heartbeat rate (hbInterval, fibrillateCount), log length limitation (tsLogLength, gsLogLength), instance numbers (instanceNum), and so forth. The instanceNum represents a new attribute. Refer to “instanceNum” on page 27 for further details. Some instance numbers are set by the SMIT Change/Show Topology and Group Services Configuration menu.

3.1.2.2 Resource-Related Global ODM Classes

The following HACMP GODM classes contain the resource-related information:

- HACMPgroup
- HACMPserver
- HACMPresource

These GODM classes have not been changed since HACMP/ES V4.2.2.

The information contained in these GODM classes is explained in the following examples.

HACMPgroup

```
# odmget -q group=risc71rg HACMPgroup

HACMPgroup:
        group = "risc71rg"
        type = "cascading"
        nodes = "risc71 sp21n11"
```

The HACMPgroup GODM class contains the resource group name (group), resource group relationship type (type: it can be cascading, rotating, or concurrent), and participating nodes (nodes). These attributes are set by the SMIT Add a Resource Group menu.

HACMPserver

```
# odmget HACMPserver

HACMPserver:
        name = "AppServer1"
        start = "/usr/sbin/cluster/local/start_AppServer1"
        stop = "/usr/sbin/cluster/local/stop_AppServer1"
```

The HACMPserver GODM class contains the application server name (name), and start (start) and stop (stop) script names for the application. These attributes are set by the SMIT Add Application Server menu.

HACMPresource

```
# odmget -q 'group=risc71rg and name=SERVICE_LABEL' HACMPresource

HACMPresource:
        group = "risc71rg"
        name = "SERVICE_LABEL"
        value = "risc71_svc"
```

The HACMPresource GODM class contains resource information regarding all resource groups. This example shows only the service IP label resource in the risc71rg resource group. However, HACMPresource has many other resource attributes, such as filesystems, filesystems to export, volume group, application server, and so forth. These attributes are set by the SMIT Configure a Resource Group menu.

3.1.2.3 Remaining Global ODM Classes

The remaining HACMP GODM classes are as follows:

HACMPcommand This GODM class contains HACMP command information.

HACMPcustom This GODM class contains user-defined customized methods, such as the customized verification method, the customized snapshot method, and a pre- or post-event script.
These attributes are set by the SMIT Add a Custom Verification Method, Add a Custom Snapshot Method, and Add a Custom Cluster Event, respectively.

HACMPdaemons	This GODM class contains HACMP daemon information.
HACMPevent	This GODM class contains HACMP event (recovery script) information. With HACMP V4.3, you can specify multiple pre- or post-event scripts. Before defining these scripts, you need to register them to the HACMPcustom GODM class, and specify their symbolic names instead of their full path names. The attributes are set by the SMIT Change/Show Cluster Events menu. For more details about multiple pre- or post-event scripts, refer to “Cluster Resources” on page 10.
HACMPfence	This GODM class contains fence information in the concurrent environment.
HACMPsp2	This GODM class contains information on how to use HPS Eprimary node takeover control.

3.1.3 GODM Changes

This section explains the following new or changed HACMP GODM attributes:

- handle (node_handle)
- cluster_version (version)
- globalname
- interfacename
- instanceNum

handle (node_handle)

Topology Services of HACMP/ES V4.2 is an enhancement of PSSP Topology Services, so it uses SP node numbers as the Cluster Node Number (CNN), which uniquely identifies that node within the cluster.

Topology Services of HACMP/ES V4.3 does not use SP node numbers. Instead, it uses a node number assigned during cluster configuration. Because of this independence from the SDR information, HACMP/ES V4.3 can be configured as a cluster that includes RS/6000 SP nodes in different partitions and/or RS/6000s.

The new node number, handle value, is kept in a new field in the HACMPcluster GODM class handle attribute for local node handle (see “HACMPcluster” on page 20), and in the HACMPnode GODM class

node_handle attribute for all node handles (see “HACMPnode” on page 21). You can get the node number by using the new `clhandle` or `odmget` command. In the following example, the node number, handle value, is 3.

```
# /usr/sbin/cluster/utilities/clhandle
3 risc71

# odmget HACMPcluster | grep handle
        handle = 3

# odmget -q name=risc71 HACMPnode | grep node_handle | uniq
        node_handle = 3
```

Note that the handle value is assigned automatically in alphanumeric order during topology synchronization. If you add a node to an existing running cluster, then it gets the lowest free handle value. You cannot assign a specific handle value. For more information, refer to 4.2.2.3, “clhandle” on page 61.

HACMP/ES V4.3 uses a *node priority* concept. With node priority, if a cluster is divided into two cluster partitions by some failure such as network down, grpsvcs determines the smallest partition and notifies its clients that it is going down, which kills the nodes in the smaller partition. If the cluster consists of only two nodes, the combination of these partitions causes the death of the node that has a large handle value. For example, if you want to configure a cascading hot standby configuration, it is a good idea to make the server node have higher priority, in other words, low handle value.

At the SP Switch (SPS) or High Performance Switch (HiPS) IP address takeover configuration, the previous version of HACMP/ES uses the SDR at start-up time. The HACMP start-up script, rc.cluster, tries to read the SPS configuration from the SDR in the Control Workstation (CWS). If the script cannot read the information because of a CWS failure, or an SP Ethernet failure for example, then the previous version of HACMP cannot start up.

In contrast, HACMP/ES V4.3 and the RSCT running in the HACMP domain do not depend on the SDR. Therefore, HACMP/ES V4.3 can start, even if there is a problem in the SDR. However, if SPS is in trouble (because of a CWS failure, for example), then HACMP cannot use the SPS network.

cluster_version (version)
The cluster version number is the version of HACMP/ES that is running on a node. The local node cluster version number is kept in the HACMPcluster GODM class cluster_version attribute (see “HACMPcluster” on page 20), and the value of each node is kept in the HACMPnode GODM class version

attribute (see “HACMPnode” on page 21). The value is set to the current HACMP/ES version value.

globalname
HACMP/ES V4.3 can combine several networks into one global logical network. The global network name is kept in the HACMPnetwork GODM class globalname attribute (see “HACMPnetwork” on page 21). By using the global network, you can make a heartbeat ring across routers. This function supports SP Ethernet environments that use boot installation servers as routers.

Note that you cannot configure a global network that consists of a mix of different type networks such as Ethernet and token ring, and IP address takeover across routers is not allowed. The global network is only used by Topology Services to decide the heartbeat route.

For more information on global networks refer to 4.2.2.5, “claddnetwork” on page 63 and 6.1, “Global Network” on page 131.

interfacename
The interface name is the network interface name for each boot, service (which has no boot adapter), and standby adapter. The value is kept in the HACMPadapter GODM class interfacename attribute (see “HACMPadapter” on page 22). It is used to make the machine.lst file. You cannot specify the interface value directly; it is stored by HACMP during topology synchronization automatically.

instanceNum
The instance number is the number of times topology synchronization has occurred. It is kept in the HACMPtopsvcs GODM class (see “HACMPtopsvcs” on page 23). It is increased by +1 when topology synchronization occurs. The Topology Services running on the HACMP domain has the instance number that refers to the GODM topology-related synchronized version. When a node joins a cluster, the instance number of GODM must match the Topology Services instance value on the other running cluster nodes.

You can get these values by using the `odmget` or `lssrc -ls topsvcs` command.

```
# odmget HACMPtopsvcs | grep instanceNum
        instanceNum = 23

# lssrc -ls topsvcs | grep Instance
   Configuration Instance = 23
```

In this example, topology synchronization has been done 23 times, and the running Topology Services shows the current GODM topology information.

3.1.4 Topology Services and GODM

The Topology Services running on the HACMP domain uses the topology information in the GODM at start time. At Topology Services start time, the System Resource Controller (SRC) subsystem calls the /usr/sbin/rsct/bin/topsvcs script, which creates the machine.lst file and starts the Topology Services daemon. In order to get the necessary information from GODM, the script uses such HACMP commands as `cllsif` and the AIX `odmget` command.

Following is an example of a machine.lst file. It contains information about the instance number, the node handle, the network interface name, IP addresses, and so on. The file name is /var/ha/run/topsvcs.CLUSTERNAME/machines.CLUSTERID.lst. In this example, the CLUSTERNAME is `clusterb` and CLUSTERID is `2`.

```
# cat /var/ha/run/topsvcs.clusterb/machines.2.lst
*InstanceNumber=23
*configId=1936237799
*!TS_realm=HACMP
*!TS_EnableIPAT
TS_Frequency=1
TS_Sensitivity=4
TS_FixedPriority=38
TS_LogLength=5000
Network Name ethernet2_0
Network Type ether
*!TS_Sensitivity=4
*
*Node Type Address
    1 en1  80.7.6.10
    3 en0  80.7.6.20
*!Service Address=80.7.6.11
*!Service Address=80.7.6.71
Network Name ethernet2_1
Network Type ether
*!TS_Sensitivity=4
*
*Node Type Address
    1 en2  80.9.9.1
    3 en1  80.9.9.2
```

3.2 Daemons

The group names of the RSCT daemons belonging to the PSSP domain all start with the letters ha. The following command combination lists all of these daemons:

```
# lssrc -a | grep ha
 hats             hats             11874   active
 hags             hags             8942    active
 hagsglsm         hags             13380   active
 haem             haem             11192   active
 haemaixos        haem             8274    active
```

Similarly, the names of the RSCT daemons belonging to the HACMP domain all end with svcs. The following command combination lists all of these daemons:

```
# lssrc -a | grep svcs
 topsvcs          topsvcs          5428    active
 grpsvcs          grpsvcs          5832    active
 grpglsm          grpsvcs          4584    active
 emsvcs           emsvcs           14318   active
 emaixos          emsvcs           5972    active
```

The group name of the HACMP/ES daemons is cluster. The following command combination lists these HACMP/ES daemons:

```
# lssrc -a | grep cluster
 clstrmgr         cluster          15450   active
 clsmuxpd         cluster          14014   active
 clinfo           cluster          6392    active
```

There are three daemons newly implemented for HACMP/ES V4.3:

- emsvcs
- emaixos
- haemaixos

The following sections discuss these.

3.2.1 Event Management

Until HACMP/ES V4.2.2, HACMP/ES shared Event Management with PSSP. More precisely, the HACMP and PSSP domains shared Event Management. HACMP/ES V4.3 must run not only on RS/6000 SP nodes but also on RS/6000s. Therefore, HACMP/ES V4.3 needs its own Event Management. emsvcs is an Event Management daemon that works only for HACMP/ES V4.3 and does not communicate to any RSCT daemons belonging to the PSSP domain, for example, hags.

3.2.2 aixos Resource Monitor

The purpose of the AIX resource monitor is to take values for selected AIX statistics out of SPMI shared memory and feed that data to the Event Management daemon through the RMAPI. The resource monitor is needed as part of the work to make the Event Management daemon as independent of Performance ToolBox (PTX) as possible. The aixos resource monitor differs from most resource monitors in that data structures specific to the resource monitor are included in the Event Management Configuration Database (EMCDB). In High Availability Infrastructure (HAI), the resource monitor was an internal resource monitor of the Event Management daemon. The Event Management daemon uses the configuration data structures directly in its address space for the entire life of the daemon. The format of the data structures is intimately tied to the algorithms used by the Event Management daemon. The configuration database in HAI Event Management defines the aixos resource monitor as an internal resource monitor. To aid in migration and coexistence, it will continue to be defined in the SDR and the configuration database as an internal resource monitor. The RMAPI and Event Management daemon for RSCT will internally convert the type internal to server for this resource monitor. To prevent the hardcoding of a path to the resource monitor, the resource monitor will be a server type resource monitor that is not started by the Event Management daemon. The aixos resource monitor will be controlled by the SRC, and will use the reliable daemon's library. emaixos is an AIX resource monitor that works for only HACMP/ES V4.3, and haemaixos is an AIX resource monitor that works for only PSSP.

3.2.3 Event Management Configuration DataBase (EMCDB)

In PSSP environment, Event Management configuration data is loaded into the System Data Repository (SDR) by the `haemloadcfg` command. This command is executed on the Control Workstation by the `haemctrl` script when the Event Management subsystem is added. Then the `haemctrl` script executes the `haemcfg` command to create an Event Management Configuration Database (EMCDB), a binary version of the data that is in the SDR. This EMCDB is placed into a staging area on the CWS. When the EM

daemons start, the EMCDB is copied from the staging area to a local directory, if necessary.

For HACMP/ES V4.3, an EMCDB is already created in the rsct.hacmp installation fileset. The following resource monitors are defined in this EMCDB:

- IBM.PSSP.harmpd
- aixos
- Membership

During the configuration of HACMP/ES V4.3, the EMCDB is copied from /usr/sbin/rsct/install/config/em.HACMP.cdb to /etc/ha/cfg/em.*domain_name*.cdb, where *domain_name* is the domain name of the HACMP cluster. When the EM daemon starts in the HACMP domain, it looks only for a local copy of the EMCDB and does not attempt to copy an EMCDB from the staging area, even if the version of the local EMCDB does not match the version string in the EM daemon peer group state.

On an RS/6000 node, you can see only one EMCDB that is for the HACMP domain:

```
# cd /etc/ha/cfg
# ls -al
total 64
drwxr-xr-x   2 root     system       512 Jul 09 13:44 .
drwxr-xr-x   3 root     system       512 Jul 02 14:59 ..
-r--r--r--   1 root     haemrm     18268 Jul 02 15:25 em.HACMP.cdb
-rw-r--r--   1 root     haemrm        22 Jul 20 15:35 em.clusterb.cdb_vers
#cat em.clusterb.cdb_vers
887653462,846074368,0
```

On an RS/6000 SP node, you can see two EMCDBs. One is for the HACMP domain and the other is for the PSSP domain:

```
# cd /etc/ha/cfg
# ls -al
total 272
drwxr-xr-x   2 root     system       512 Jul 09 13:57 .
drwxr-xr-x   3 root     system       512 Jul 07 14:36 ..
-r--r--r--   1 root     haemrm     18268 Jul 20 17:47 em.HACMP.cdb
-rw-r--r--   1 root     haemrm        22 Jul 20 15:26 em.clusterb.cdb_vers
-rw-r--r--   1 root     haemrm    100428 Jul 07 14:51 em.sp21en0.cdb
-rw-r--r--   1 root     haemrm        22 Jul 20 17:30 em.sp21en0.cdb_vers
# cat em.clusterb.cdb_vers
887653462,846074368,0
# cat em.sp21en0.cdb_vers
899417748,130272768,0
```

3.3 Cluster Manager

This section describes the Cluster Manager. It includes Cluster Manager control flow, initial formation, and so on.

3.3.1 Control Flow

The Cluster Manager runs a finite state machine to sequence its processing. The states and transitions are shown in Figure 6 on page 34. It starts out in the INIT state and the machine is run given a state and a finite state machine event (FSM-event). The FSM-event is a Cluster Manager internal transition trigger and has no relationship to HACMP events or user-defined events. The FSM-events are shown in the transition arrows in Figure 6 on page 34. Each state and FSM-event pair results in a new state. After the specified function is executed, the current state is updated to the next state as shown in Table 2 on page 36.

3.3.1.1 State Diagram

When HACMP/ES starts, the Cluster Manager changes its state to the STABLE state from the INIT state via the JOINING state, and enters the main loop waiting for events to run recovery actions.

When recovery events that need to run recovery actions happen, the Cluster Manager usually changes its state to the VOTING state via the UNSTABLE state. And the Cluster Manager starts the vote of approve or reject to the strongest queued recovery event by using VOTE_MSG FSM-event.

Note that the FSM-events and recovery events are not directly related. The recovery events like TE_JOIN_NODE, TE_FAIL_NODE, TE_SWAP_ADAPTER are related to the recovery programs. You can see

them in the /usr/sbin/cluster/events/rules.hacmprd file. Refer to 6.2.3, “The rules.hacmprd File” on page 141 for more detail.

The Cluster Managers do the recovery program, usually using RP_RUNNING, BARRIER, and the CBARRIER states. RP_RUNNING is the event script execute phase, the BARRIER state is the synchronization phase of the recovery program execution in the cluster, and the CBARRIER state is the synchronization phase after the recovery program ends in the cluster. Refer to 6.2.2, “The Recovery Program” on page 139 for more detail.

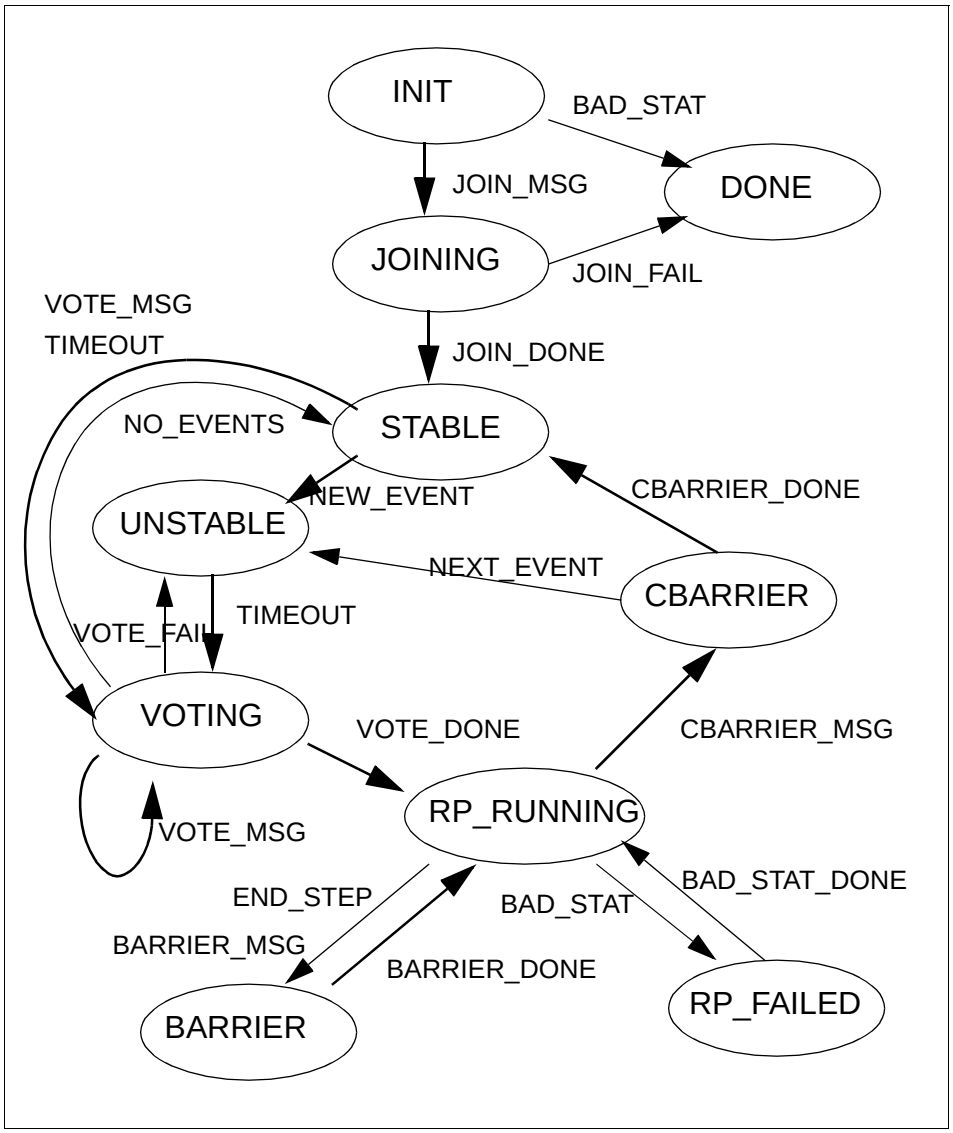


Figure 6. Cluster Manager State Diagram

3.3.1.2 State Transition Table

The Cluster Manager contains functions to run a finite state machine (FSM) based on the state transition table shown in Table 2 on page 36. From the CURRENT STATE and FSM-EVENT pair, the Cluster Manager executes the specified function. Then the state is updated to the NEXT STATE field in the

table. The * matches any state or FSM-event. The = represents the same (current) state.

Table 1. Cluster Manager State Transition Table (1 of 2)

CURRENT STATE	FSM-EVENT	NEXT STATE
DONE	*	=
*	QUIT	DONE
INIT	BAD_STAT	DONE
INIT	JOIN_MSG	JOINING
JOINING	JOIN_FAIL	DONE
JOININIG	JOIN_DONE	STABLE
STABLE	NEW_EVENT	UNSTABLE
STABLE	VOTE_MSG	UNSTABLE
STABLE	BARRIER_MSG	=
STABLE	CBARRIER_MSG	=
STABLE	TIMEOUT	VOTING
UNSTABLE	TIMEOUT	VOTING
UNSTABLE	VOTE_MSG	VOTING
UNSTABLE	NEW_EVENT	=
UNSTABLE	BARRIER_MSG	=
UNSTABLE	CBARRIER_MSG	=
VOTING	TIMEOUT	=
VOTING	BARRIER_MSG	=
VOTING	CBARRIER_MSG	=
VOTING	VOTE_MSG	=
VOTING	VOTE_FAIL	UNSTABLE
VOTING	VOTE_DONE	RP_RUNNING

Table 2. Cluster Manager State Transition Table (2 of 2)

CURRENT STATE	FSM-EVENT	NEXT STATE
RP_RUNNING	NEW_EVENT	=
RP_RUNNING	BARRIER_MSG	BARRIER
RP_RUNNING	BAD_STAT	RP_FAILED
RP_RUNNING	CBARRIER_MSG	CBARRIER
RP_RUNNING	END_RP	CBARRIER
BARRIER	NEW_EVENT	=
BARRIER	BARRIER_MSG	=
BARRIER	BARRIER_DONE	RP_RUNNING
RP_FAILED	NEW_EVENT	=
RP_FAILED	CBARRIER_MSG	=
RP_FAILED	BARRIER_MSG	=
RP_FAILED	BAD_STAT_DONE	RP_RUNNING
CBARRIER	NEW_EVENT	=
CBARRIER	CBARRIER_MSG	=
CBARRIER	CBARRIER_DONE	STABLE
CBARRIER	NEXT_EVENT	UNSTABLE

3.3.2 Initial Cluster Formation

This section describes the steps taken by the HACMP/ES Cluster Manager when the user starts it. It shows Cluster Manager state changes, and how the HACMP/ES Cluster Manager interacts with Group Services.

In this scenario, the user starts up the Cluster Manager on Node A first, then starts up the Cluster Manager on Node B.

1. INIT state (Node A)

   The HACMP/ES Cluster Manager is started on Node A.

   When the HACMP/ES Cluster Manager starts on a node, it starts in the INIT state. See “Cluster Manager State Diagram” in Figure 6 on page 34. If there is an error in reading the configuration or allocating memory for

data structures, the Cluster Manager enters the DONE state and terminates.

The HACMP/ES Cluster Manager registers with Group Services and creates the CLSTRMGR_xxx provider group. The provider group name will be the string CLSTRMGR_ concatenated with the HACMP Cluster ID assigned by the administrator when the cluster was configured. This will allow multiple clusters within a single Group Services domain and namespace. If no HACMP/ES configuration is found on a node, the HACMP/ES Cluster Manager exits with an error message. If the GODM configuration indicates HACMP/ES has network adapters to monitor, it subscribes to those Group Services adapter groups. Node A is ready to join the provider group and sends a join request to Group Services.

2. JOINING state (Node A)

   Once the join request is sent to Group Services, the Cluster Manager enters the JOINING state. It joins its Group Services provider group using a multiphase join protocol. Refer to 4.2.3.11, “hagsvote” on page 85 for more information on protocols. During the join protocol, Node A sends the state of its adapters to the running cluster (there is none in this case) and the running cluster sends the state of its adapters to the joining node (Node A in this case). If there are many nodes trying to join a cluster at the same time, any node within the established cluster will not allow the join until their event queue is empty for a time-out period (30 sec. +(10 * nodehandle milliseconds)).

3. STABLE state (Node A)

   When Group Services notifies the Cluster Manager of its membership in the group, the Cluster Manager enters the STABLE state.

4. UNSTABLE state (Node A)

   The Cluster Manager puts its join event on the event queue. Once an event goes on the event queue, the Cluster Manager enters the UNSTABLE state.

5. VOTING state (Node A)

   The Cluster Manager waits for a time interval for the queue to stabilize, and then enters the VOTING state. A two-phase event voting protocol is initiated to reach consensus on the next event to process.

6. RP_RUNNING state (node_up.rp phase1) (Node A)

   The node_up event for Node A is voted to be the next event to process and Node A enters the RP_RUNNING state and executes its node_up recovery program *(*node_up.rp*)*. The node_up recovery program usually

runs the node_up event script on the other nodes first (there are none in this case).

The phase number used in this discussion and the node_up recovery program (node_up.rp) structure are shown in Table 3 on page 38. Refer to 6.2.2, “The Recovery Program” on page 139 for more details.

This RP_RUNNING state is related to phase 1 in this table, and the next step BARRIER state is related to phase 2, and so on.

Table 3. Phase Number and node_up Recovery Program Structure

Phase	Node Set	Recovery Command	Expected Status
1	other	node_up event script	0
2		barrier	
3	event	node_up event script	0
4		barrier	
5	all	node_up_complete event script	X

7. BARRIER state (node_up.rp phase2) (Node A)

   The node_up recovery program encounters the `barrier` command, which causes the Cluster Manager to enter the BARRIER state and a two-phase barrier protocol is initiated. In this case, there is only one node, so the barrier protocol does not wait for the other nodes.

8. RP_RUNNING state (node_up.rp phase3) (Node A)

   When the barrier protocol is complete, the Cluster Manager enters the second RP_RUNNING state. The node_up recovery program runs the node_up event script on the local node. This script causes the local node to claim all of the resources (except application server) for which it is configured.

9. BARRIER state (node_up.rp phase4) (Node A)

   The node_up recovery program encounters the `barrier` command again and the Cluster Manager enters the BARRIER state.

10. RP_RUNNING state (node_up.rp phase5) (Node A)

    When Group Services indicates the barrier protocol is complete, the Cluster Manager enters the RP_RUNNING state and runs the node_up_complete event script. This script causes the local node to claim application server resources for which it is configured.

11. CBARRIER state (Node A)

When the end-of-file is encountered in the recovery program, the Cluster Manager enters the CBARRIER state and runs a two-phase cbarrier protocol.

12. STABLE state (Node A)

    When the cbarrier protocol is complete the Cluster Manager enters the STABLE state if the event queue is empty; otherwise it enters the UNSTABLE state.

At this point, Node A has started. The next steps are the joining procedure of the other node (Node B):

13. INIT state (Node B)

    The Cluster Manager is started on Node B in the INIT state.

14. JOINING state (Node B)

    It registers with Group Services and joins the CLSTRMGR_xxx provider group. Once the join request is sent to Group Services, the Cluster Manager enters the JOINING state.

    If the GODM configuration indicates HACMP has network adapters to monitor, it subscribes those to Group Services adapter groups. If there is an error in reading the configuration or allocating memory for data structures, the Cluster Manager enters the DONE state and terminates.

15. STABLE state (Node B)

    When Group Services notifies the Cluster Manager of its membership in the group, the Cluster Manager enters the STABLE state.

16. UNSTABLE state (Node B)

    When the Cluster Managers on Nodes A and B complete the join protocol for Node B, Node B adds a join event for itself to its event queue and enters the UNSTABLE state.

17. VOTING state (Node B)

    After the event queue stabilizes, the Cluster Manager on Node B enters the VOTING state and initiates a two-phase voting protocol between nodes A and B.

18. UNSTABLE, VOTING state (Node A)

    Node A enters the VOTING state via the UNSTABLE state when it is notified that a Voting protocol has been initiated.

19. RP_RUNNING state (node_up.rp phase1) (Node A,B)

The node_up event for Node B is voted the next event to process and the Cluster Managers on both nodes enter the RP_RUNNING state and execute the node_up recovery program (node_up.rp). The node_up recovery program runs the node_up event script on all nodes in the membership before Node B joined (Node A in this case). The shell scripts run by the recovery program may release resources currently held by Node A, if both are in the resource chain for one or more resources and Node B has a higher priority for any of the resources.

20. BARRIER state (node_up.rp phase2) (Node A,B)

    The node_up recovery program encounters a `barrier` command, which causes the Cluster Manager to enter the BARRIER state and initiates a two-phase barrier protocol with all nodes, which in turn causes all nodes to wait until everyone reaches the `barrier` command in the recovery program.

21. RP_RUNNING state (node_up.rp phase3) (Node A,B)

    When the barrier protocol is complete, the Cluster Manager enters the second RP_RUNNING state. The Cluster Manager on Node B executes the node_up recovery program, which runs the node_up event script. This node_up event script causes Node B to claim all of the resources (except application server) for which it is configured.

22. BARRIER state (node_up.rp phase4) (Node A,B)

    A `barrier` command is encountered, resulting in another two-phase barrier protocol and state transitions as previously described.

23. RP_RUNNING state (node_up.rp phase5) (Node A,B)

    The state transitions are as previously described, leaving all nodes in the RP_RUNNING state. The node_up recovery program executes the node_up_complete event script on all nodes. This script causes the local nodes to claim the application server resources for which it is configured. In this case Node B claims its application server resources.

24. CBARRIER state (Node A,B)

    The end-of-file is reached in the node_up recovery program. The Cluster Manager enters the CBARRIER state and runs a two-phase cbarrier protocol.

25. STABLE state (Node A,B)

    When the cbarrier protocol is complete, the Cluster Manager enters the STABLE state if the event queue is empty, otherwise it enters the UNSTABLE state.

At this point, all resources configured for either or both Node A and Node B are available to cluster clients.

If the Cluster Manager on Node N started, it would follow steps 13 to 25.

The Cluster Manager leaves messages in the /tmp/clstrmgr.debug file. You can trace the state of the Cluster Manager by this file. For more details about this file, refer to 4.1.2.4, “The /tmp/clstrmgr.debug File” on page 53 and 5.6, “Cluster Manager” on page 116.

3.3.3 Node Departure

The Cluster Manager uses Group Services to keep track of the status of nodes within the cluster. If a node fails or the Cluster Manager on the node is stopped purposefully, Group Services detects this and a multi-phased protocol is initiated. Then the peer nodes take the necessary actions to get critical applications up and running and to ensure that data remains available.

3.3.3.1 User-Controlled Stops

You can stop the HACMP/ES Cluster Manager in two different ways:

Graceful	The Cluster Manager sends a message to the other nodes indicating this is a graceful down. It shuts down after the last phase node_down_complete event script has run and the node has released its resources. The surviving nodes do *not* take over these resources.
Graceful with Takeover	The Cluster Manager shuts down after the last phase node_down_complete event script has run and the node has released its resources. The surviving nodes take over these resources.

3.3.3.2 Node Failure

When a node fails, the Cluster Managers on the surviving nodes recognize that the node_down has occurred when Group Services initiates a membership protocol and they execute the node_down recovery program.

3.3.4 Node Rejoining the Cluster

When a node rejoins the cluster, the Cluster Managers running on the existing nodes recognize a node_up event when Group Services initiates a membership protocol. All nodes, including the rejoining node execute their node_up recovery programs, which program runs the node_up script on all nodes except the rejoining node, to acknowledge that the returning node is

up and to release any resources belonging to it. The returning node then runs its node_up script so it can resume providing cluster resources. Whether or not resources are actually released in this situation depends on how the resources are configured for takeover.

3.4 User-Defined Events

This section is intended to give you a brief overview of user-defined events.

An HACMP cluster is event driven. In HACMP/ES you have two kinds of events: predefined events and user-defined events. The predefined events are triggered by Group Services. These events are part of the installation code and their function is the same as in HACMP for AIX. User defined events are only available in HACMP/ES. These events are triggered by the Event Management subsystem.

When an event occurs, the following steps are processed:

1. Group Services or Event Management notifies the cluster manager.
   - Group Services handles the predefined events
   - Event Management handles the user-defined events
2. The Cluster Manager uses the information in the rules.hacmprd file to map the recovery actions to the event.
3. The Cluster Manager executes the scripts specified in the recovery program file.
4. The Cluster Manager receives the return codes and compares them with the values in the recovery program file.
5. When the last script mentioned in the recovery program file has finished on all cluster nodes, the cluster manager is able to handle the next event.

When you start the first node (HACMP start) in your cluster, this node is be the subscriber node to Event Management. All the other nodes will not subscribe to Event Management again. The only exception is that if the first node dies the second one becomes the subscriber node to Event Management.

If an event happens on a node (in this case not the subscriber node), the following steps happen. The flow is also shown in Figure 7 on page 43.

1. The Resource Monitor detects the event (problem) and sends an event to the Event Manager.

2. The Event Manager receives the event and sends it to the Event Manager of the node where the subscription for this event is.
3. The Event Manager on the subscriber node receives it and sends it to the HACMP/ES Cluster Manager.
4. The HACMP/ES Cluster Manager receives it and communicates it to the other Cluster Managers.
5. Each Cluster Manager reads the Recovery Program file for this event and executes the necessary scripts.

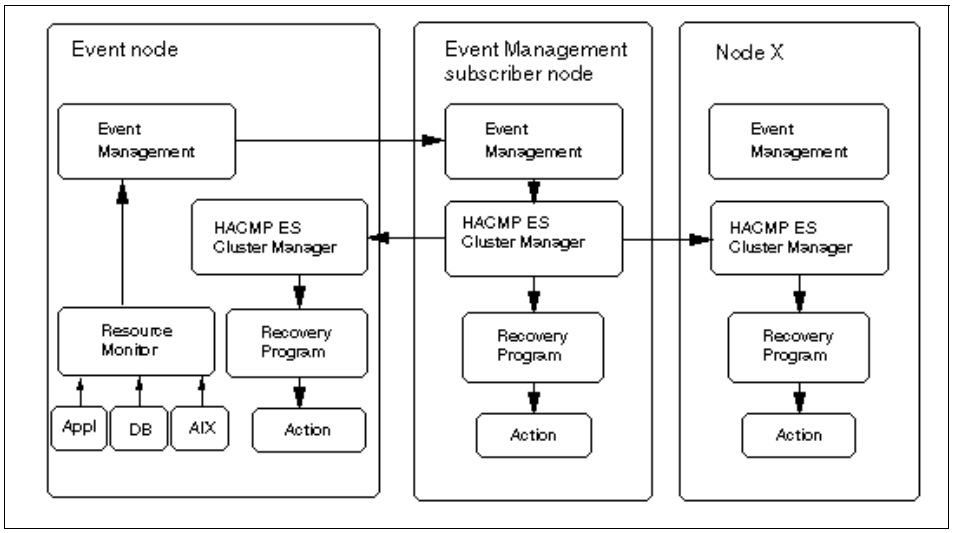


Figure 7. User-Defined Event Detection

For more details about the rules.hacmprd file, recovery programs, and script files, refer to 6.2, “User-Defined Events” on page 138.

Chapter 4. Log Files and Commands

This chapter describes useful log files and commands to better understand IBM HACMP Enhanced Scalability (HACMP/ES) V4.3 and IBM RS/6000 Cluster Technology (RSCT).

4.1 Log Files

This section describes the different log files used by HACMP/ES V4.3, making a distinction between the log files that are common with HACMP for AIX and those that are specific to HACMP/ES.

The first approach to diagnosing a problem affecting a cluster is to examine the log files. It is important to understand that every component (AIX, Topology Services, Group Services, Event Management, and HACMP/ES) has its own log files.

Every time a problem occurs on an HACMP/ES V4.3 cluster, it is the responsibility of the system administrator to save *all* the relevant log files and eventual core files from *all* cluster nodes and provide them to the IBM support personnel. In order to save disk space, many log files are removed by HACMP/ES on a regular basis, so it is very important to save them immediately after the problem has occurred.

Note: When a problem occurs in HACMP/ES V4.3, often many commands in the same event shell script will fail, generating multiple error messages. The system administrator must always debug a problem by scanning the log file *from the top to the bottom*. Most of the time the errors found at the end of the file are just a consequence of the first, original problem at the top of the log file.

Chapter 5, “Problem Determination” on page 89 shows some real HACMP/ES problems and how to debug them looking at the log files presented in this section.

4.1.1 Log Files in Common with HACMP for AIX

HACMP/ES V4.3 and HACMP for AIX V4.3 have in common the following log files:

- /tmp/hacmp.out
- /var/adm/cluster.log
- /usr/sbin/cluster/history/cluster.mmdd

- /tmp/cspoc.log
- /tmp/emuhacmp.out
- System Error Log

In the following sections we examine these files and describe the kind of information that is written inside each one.

4.1.1.1 The /tmp/hacmp.out Log File

As with all previous versions of HACMP for AIX, /tmp/hacmp.out is the primary log file to look at when you suspect a problem related to one of its event shell scripts. By default, all shell scripts have the Debug Level set to High, which means activating the `set -x` command at the beginning of the Korn shell script. We strongly recommend that you leave the Debug Level set to High, as shown in Figure 8 on page 46:

```
Change/Show Run Time Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Node Name                                      sp21n13

  Debug Level                                    high                    +
  Host uses NIS or Name Server                   false                   +

F1=Help            F2=Refresh         F3=Cancel          F4=List
Esc+5=Reset        F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

Figure 8. HACMP/ES Runtime Parameters

In addition to the /tmp/hacmp.out file, which refers to the current day, HACMP/ES V4.3 also maintains a copy of /tmp/hacmp.out of the last seven days: /tmp/hacmp.out.1 contains debug information from the previous day, /tmp/hacmp.out.2 from two days ago, /tmp/hacmp.out.3 from three days ago and so on. The shell script /usr/sbin/cluster/utilities/clcycle, executed automatically from crontab every night, is responsible for renaming these files.

4.1.1.2 The /var/adm/cluster.log File

The /var/adm/cluster.log file provides a high-level view of the event shell scripts executed by the cluster manager. HACMP/ES V4.3 writes two lines in this file for every event it executes. The first line is logged when an event starts its execution, and it contains the string EVENT START plus additional information, while the second line is logged when the event has finished. If the event shell script has completed without any errors, the Cluster Manager logs EVENT COMPLETED. If the shell script does not exit with a return code of zero, then HACMP/ES logs EVENT FAILED instead.

During the installation of HACMP/ES V4.3, the file /etc/syslogd.conf is updated with new entries in order for the syslogd daemon to write additional debug and information messages in cluster.log, as shown in Figure 9 on page 47:

```
# tail /etc/syslog.conf

#  mail.debug           /usr/spool/mqueue/syslog
#  *.debug              /dev/console
#  *.crit                       *
daemon.notice           /var/adm/SPlogs/SPdaemon.log
# HACMP/ES for AIX Critical Messages from HACMP/ES for AIX
local0.crit /dev/console
# HACMP/ES for AIX Informational Messages from HACMP/ES for AIX
local0.info /usr/adm/cluster.log
# HACMP/ES for AIX Messages from Cluster Scripts
user.notice /usr/adm/cluster.log
```

Figure 9. The /etc/syslog.conf File

The file /var/adm/cluster.log can become quite large after HACMP/ES has been running for some time. It is the responsibility of the system administrator to clean it out periodically.

Figure 10 on page 48 is an example of the cluster.log file.

```
# pg /var/adm/cluster.log

Jul  9 13:34:44 sp21n13 HACMP for AIX: EVENT START: node_down sp21n13 graceful
Jul  9 13:34:44 sp21n13 HACMP for AIX: EVENT START: node_down_local rg13
Jul  9 13:34:44 sp21n13 HACMP for AIX: EVENT START: release_vg_fs
Jul  9 13:34:45 sp21n13 HACMP for AIX: EVENT COMPLETED: release_vg_fs
Jul  9 13:34:45 sp21n13 HACMP for AIX: EVENT COMPLETED: node_down_local rg13
Jul  9 13:34:45 sp21n13 HACMP for AIX: EVENT COMPLETED: node_down sp21n13 gracef
ul
Jul  9 13:34:45 sp21n13 HACMP for AIX: EVENT START: node_down_complete sp21n13 g
raceful
Jul  9 13:34:46 sp21n13 HACMP for AIX: EVENT START: node_down_local_complete
Jul  9 13:34:46 sp21n13 HACMP for AIX: EVENT COMPLETED: node_down_local_complete
Jul  9 13:34:46 sp21n13 HACMP for AIX: EVENT COMPLETED: node_down_complete sp21n
13 graceful
```

Figure 10. The /var/adm/cluster.log File

4.1.1.3 The /usr/sbin/cluster/history/cluster.mmdd File

The system creates a cluster history file for every day of the month that event shell scripts have been executed. Each file is identified by its file name extension, where *mm* indicates the month and *dd* indicates the day. The contents of this file are basically identical to the contents of the /var/adm/cluster.log file.

This file can become useful in case there is a need to look at what occurred to the cluster a long time back. However, HACMP/ES V4.3 does not periodically remove the old cluster.*mmdd* files; that is a system administrator responsibility in order to preserve disk space.

4.1.1.4 The /tmp/cspoc.log File

The Cluster Single Point of Control facility (C-SPOC) logs its output and error messages in the file /tmp/cspoc.log, which is file is only created on the cluster node where the C-SPOC command is invoked.

4.1.1.5 The /tmp/emuhacmp.out File

HACMP/ES V4.2.2 has introduced a new functionality called Event Emulation, which allows the system administrator to emulate specific cluster events. To emulate a cluster event means that the cluster manager runs the event shell script by just emulating the commands, without actually executing them. The log file /tmp/emuhacmp.out is a text file residing on the node from which the HACMP/ES Event Emulator was invoked. The file contains information from log files generated on all nodes in the cluster. When the emulation is complete, the information in these files is transferred to the

/tmp/emuhacmp.out file on the node from which the emulation was invoked, and all other files are deleted.

The example that follows shows the SMIT menu to simulate the event `fail_standby` for the standby adapter called `n13_stdby` on cluster node sp21n13.

```
                        Emulate Fail Standby Event

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Node Name                                        [sp21n13]                +
* IP Label                                         [n13_stdby]              +









F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

When the event emulation is finished, the /tmp/emuhacmp.out file contains log information of all cluster nodes, in our case sp21n13 and sp21n15, as shown in the next screen:

```
********************************************************************
******************START OF EMULATION FOR NODE sp21n13***************
********************************************************************

Jul 16 10:48:58 EVENT START: fail_standby sp21n13 n13_stdby

+ set -u
+ + dspmsg scripts.cat 335 Adapter n13_stdby is no longer available for use as a
 standby,\n due to either a standby adapter failure or IP address takeover.\n n1
3_stdby
MSG=Adapter n13_stdby is no longer available for use as a standby,
 due to either a standby adapter failure or IP address takeover.
+ /bin/echo Adapter n13_stdby is no longer available for use as a standby, due t
o either a standby adapter failure or IP address takeover.
+ 1> /dev/console
+ exit 0
Jul 16 10:48:59 EVENT COMPLETED: fail_standby sp21n13 n13_stdby

************************************************************
**************END OF EMULATION FOR NODE sp21n13*****************
************************************************************
********************************************************************
******************START OF EMULATION FOR NODE sp21n15***************
********************************************************************

Jul 16 10:48:59 EVENT START: fail_standby sp21n13 n13_stdby

+ set -u
+ + dspmsg scripts.cat 335 Adapter n13_stdby is no longer available for use as a
 standby,\n due to either a standby adapter failure or IP address takeover.\n n1
3_stdby
MSG=Adapter n13_stdby is no longer available for use as a standby,
 due to either a standby adapter failure or IP address takeover.
+ /bin/echo Adapter n13_stdby is no longer available for use as a standby, due t
o either a standby adapter failure or IP address takeover.
+ 1> /dev/console
+ exit 0
Jul 16 10:48:59 EVENT COMPLETED: fail_standby sp21n13 n13_stdby

************************************************************
**************END OF EMULATION FOR NODE sp21n15*****************
************************************************************
```

4.1.1.6 System Error Log

The HACMP/ES V4.3 messages in the System Error Log follow the same format as that used by other AIX subsystems. In case of problems, the topsvcs, grpsvcs, and emsvcs daemons write entries in the System Error Log.

4.1.2 Log Files Specific to HACMP/ES V4.3

HACMP/ES V4.3 has these additional log files that are typically used by Topology Services, Group Services, and Cluster Manager:

- /var/ha/log/topsvcs
- /var/ha/log/grpsvcs
- /var/ha/log/grpglsm
- /tmp/clstrmgr.debug

One important thing to remember is that under certain conditions the topsvcs, grpsvcs, or grpglsm may core dump. The core file is placed in the directory under /var/ha/run/topsvcs.clustername for topsvcs, under /var/ha/run/grpsvcs.clustername for grpsvcs and under /var/ha/run/grpglsm.clustername for grpglsm, where *clustername* is the cluster name used during HACMP/ES V4.3 configuration. The core files are extremely important when debugging problems and must be provided to the IBM support personnel in addition to the log files.

In the following sections we examine these files and describe the kind of information that is written inside each one.

4.1.2.1 The /var/ha/log/topsvcs File

First of all, the real file name is not /var/ha/log/topsvcs but rather /var/ha/log/topsvcs.day.hhmmss.cluster_name, where *day* is the day of the month and *hhmmss* is the hour, minute, and second the file was created. *cluster_name* is the HACMP/ES V4.3 cluster name used during the configuration. This log file contains time-stamped messages that track the execution of the internal activities of the topsvcs daemon.

This log file is intended to be used mainly by IBM support personnel for troubleshooting. However, some information can be interpreted quite easily, as shown in the following example.

```
07/10  18:18:00 hatsd[0]: My New Group ID = (128.100.10.30:0x85a69314) and is Un
stable.
        My Leader is              (128.100.10.30:0x85a69314).
        My Crown Prince is        (128.100.10.30:0x85a69314).
        My upstream neighbor is   (128.100.10.30:0x85a69314).
        My downstream neighbor is (128.100.10.30:0x85a69314).
07/10  18:18:00 hatsd[1]: My New Group ID = (128.200.30.3:0x85a69315) and is Uns
table.
        My Leader is              (128.200.30.3:0x85a69315).
        My Crown Prince is        (128.200.30.3:0x85a69315).
        My upstream neighbor is   (128.200.30.3:0x85a69315).
        My downstream neighbor is (128.200.30.3:0x85a69315).
07/10  18:18:00 hatsd[2]: My New Group ID = (192.168.4.13:0x45a69316) and is Uns
table.
        My Leader is              (192.168.4.13:0x45a69316).
        My Crown Prince is        (192.168.4.13:0x45a69316).
        My upstream neighbor is   (192.168.4.13:0x45a69316).
        My downstream neighbor is (192.168.4.13:0x45a69316).
```

This information shows how Topology Services builds a ring for each physical network in order to exchange heartbeats. All the HACMP/ES cluster nodes connected to the physical network belong to the ring. Each node sends heartbeats to its downstream neighbor and receives heartbeat from its downstream neighbor. The Leader is the cluster node in the ring having the highest IP address and is usually called the Group Leader. The Crown Prince is the node that takes over the responsibility of the Group Leader when the Group Leader leaves the ring and has the next highest IP address. For additional information about the concept of Group Leader, refer to 4.2.3.2, “Group Leader” on page 66.

4.1.2.2 The grpsvcs Log Files

The grpsvcs daemon creates two log files: /var/ha/log/grpsvcs_X_Y.*clustername*. The other is called and /var/ha/log/grpsvcs.default.X_Y, where X is the HACMP/ES node number, Y is the instance number and *clustername* is the cluster name as defined during the HACMP/ES configuration.

Note: It is important not to confuse the HACMP/ES node number and the SP node number. The HACMP/ES node number is a number associated with the cluster node name. The `clhandle -a` command shows the correspondence between each cluster node name and its node number. Refer to 4.2.2.3, “clhandle” on page 61 for more information on the `clhandle` command. The SP node number is a number assigned to each RS/6000 SP node. The `splstdata -n` command shows the correspondence between each RS/6000 SP node name and its node number.

These two files contain time-stamped messages to track the execution of internal activities of the grpsvcs daemon. They are intended to be used mainly by IBM support personnel for troubleshooting.

4.1.2.3 The grpglsm Log Files

The grpglsm daemon creates two log files: /var/ha/log/grpglsm_X_Y.*clustername,* and /var/ha/log/grpglsm.default.X_Y*,* where X is the HACMP/ES node number, Y is the instance number and *clustername* is the cluster name as defined during the HACMP/ES configuration. These two files contain time-stamped messages to track the execution of internal activities of the grpglsm daemon. They are intended to be used mainly by IBM support personnel.

4.1.2.4 The /tmp/clstrmgr.debug File

The /tmp/clstrmgr.debug log file contains time-stamped messages generated by the clstrmgr daemon. When HACMP/ES V4.3 is started on a cluster node, this is the first file where the Cluster Manager daemon writes information about its activity. So in case no information is written in the /tmp/hacmp.out log file after starting the cluster, this file is a good place to look to understand what is occurring with HACMP/ES.

Every time you start HACMP/ES V4.3, the Cluster Manager daemon creates a new /tmp/clstrmgr.debug file. In case one already exists, it is saved and renamed to /tmp/clstrmgr.debug.1.

This log file is intended to be used mainly by IBM support personnel for troubleshooting. However, some information can be used to know the state of Cluster Manager. Section 5.6, “Cluster Manager” on page 116 gives you more information.

4.2 Commands

In this section we look at the various commands and utilities that are available to administrators which enable them to examine in further detail the current or changing state of the daemons and clusters involved. Not only do they provide a greater understanding of how the daemons interact, but in conjunction with the log files, they provide a means to perform problem determination.

4.2.1 Commands for AIX

First we look at how existing standard AIX commands can be used to extract information about the various subsystems running on the system.

4.2.1.1 The lssrc Command

The `lssrc` command can only be issued to daemons or subsystems that are listed as being controlled by the System Resource Controller (SRC).

The `lssrc` command sends a request to the SRC to get status on a subsystem, a group of subsystems, or all subsystems. It sends a subsystem request packet to the daemon to be forwarded to the subsystem for a subserver status or a long subsystem status.

Let us take a look at some sample outputs for the various RSCT components when using the `lssrc` command.

Topology Services Daemon

First we examine the Topology Services daemon (topsvcs).

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          18342   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.1    128.100.10.3
 ethernet1_0    [ 0]              0x85a6258a      0x85a6276d
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    2  S 128.200.20.2    128.200.30.3
 ethernet1_1    [ 1]              0x85a6253c      0x85a6271b
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.15    192.168.4.15
 spether_0      [ 2]              0x45a6253d      0x45a6271b
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.15   192.168.14.15
 basecss_0      [ 3]              0x45a6253e      0x45a6271b
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.15     140.40.4.15
 aliascss_0     [ 4]              0x45a6258f      0x45a62793
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd(  3794) hagsd(  6184)
   Configuration Instance = 7
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
```

So what does this output tell us?

We can see which subsystem is being interrogated, its current process identifier (PID) and status, normally shown for all `lssrc` output.

Next, we have the daemon-specific information:

Network Name This is the network label we assigned during the initial configuration of our HACMP/ES cluster. Each network

label has a suffix attached to differentiate between the current heartbeat networks.

Indx This is the system-assigned index array, which can be cross-referenced with the current topsvcs log file in the directory /var/ha/log. Refer to 4.1.2.1, “The /var/ha/log/topsvcs File” on page 51.

Defd This is the total number of adapters defined to Topology Services on this network that we would expect to be heartbeating under normal operation.

Mbrs This is the actual number of adapters currently heartbeating on this network.

St This is the state of this network, for which valid values are:

- **S** The group is stable. Currently no protocols are running and no nodes are either joining or being declared dead.
- **U** The group is unstable. This should be a temporary state where a new node or adapter is being included into the group or a node or adapter has been declared dead and a new group is being declared.
- **D** The local adapter is disabled. This means that topsvcs knows that it has a locally defined adapter that should be included in the group, however, it cannot currently use the local adapter to heartbeat. An example of this might be when the node is fenced from the switch.

Adapter ID This is the adapter identifier, which, on IP networks, is the adapter IP address.

Group ID This is the group identifier, which is the same value as the Group Leader’s adapter identifier.

HB Interval, Sensitivity

For each network type we can set the tunable parameters that will affect the rate at which a change in the group is detected. In the previous version of HACMP/ES 4.2.2, when the software only ran on SP systems, the tunables were controlled through the System Data Repository (SDR), and were system wide. Now with HACMP/ES 4.3, we are able to set the parameters for each network type through HACMP/ES to bring the software in line with HACMP for AIX, where each Network Interface Module

(NIM) has an option to set the failure detection rate or fibrillate count.

Group Services Daemon

Now let us examine the `lssrc` output when run against Group Services (grpsvcs).

```
# lssrc -ls grpsvcs
Subsystem         Group            PID     Status
 grpsvcs          grpsvcs          6184    active
 3 locally-connected clients.  Their PIDs:
 3794 16692 6062
 HA Group Services domain information:
 Domain established by node 2.
 Number of groups known locally: 3
                   Number of   Number of local
 Group name        providers   providers/subscribers
 cssMembership            2           1           0
 ha_em_peers              2           1           0
 CLSTRMGR_1               2           1           0
```

Figure 11. The lssrc Command Against the grpsvcs Daemon

Again, we first see which subsystem is being examined, that is, grpsvcs.

Next we see that there are three locally connected clients. To understand what this means, we need to clarify what a client, a subscriber, and a provider are:

Client A generic term applied to both subscribers and providers.

Provider Any process that is a member of a defined group within the grpsvcs domain.

Subscriber A process in the domain that requests to monitor a group. It is notified of group activities but is not a listed member.

Therefore, we have three processes that are either group members or monitoring the status of a group. The PIDs for the clients are listed, so we can find out which processes they are.

```
# ps -ef|egrep "3794|16692|6062"|grep -v grep
root  3794  6970   0 10:29:22      -  0:00 haemd HACMP 2 cluster_a SECNOSUPP
ORT
root  6062  6970   0 10:29:28      -  0:12 /usr/sbin/cluster/clstrmgr
root 16692  6970   0 10:29:19      -  0:00 hagsglsmd grpglsm
```

The information displayed next in the `lssrc` output tells us which node established the domain. In this example, node 2 established the domain. Formerly, with HACMP/ES V4.2.2, the node number listed matched the actual node number in the SDR. However, since we are now able to include systems outside of the RS/6000 SP and span multiple RS/6000 SP systems, the node number no longer matches the SDR node number. Instead, HACMP/ES allocates incremental node numbers starting at one. These are held in the HACMPnode class of the Global Object Data Manager (GODM) and can be interrogated by the `clhandle` command. See “HACMPnode” on page 21 and 4.2.2.3, “clhandle” on page 61 for more details.

At the end of the `lssrc` output we see that three groups have been established. We also have the following fields:

Group name This is the name assigned by Group Services for clients to use when they wish to join or monitor a group.

Number of providers

Shows the total number of active providers in the domain for this group.

Number of local providers/subscribers

Lists the number of local providers or subscribers on this node. In our example we see that each group has one local provider, which matches the processes we listed. For example, the local providing process for cssMembership group is PID 16692 hagsglsmd grpglsm. We cannot directly determine this association at this stage. However, we will look at how this can be achieved with the `hagscl` and `hagsgr` commands. Refer to 4.2.3.6, “hagscl” on page 69 and 4.2.3.7, “hagsgr” on page 73 for more detail.

Switch Daemon

Another `lssrc` output that may be useful is when it is run against the switch daemon (grpglsm), as follows:

```
# lssrc -ls grpglsm
Subsystem         Group            PID     Status
 grpglsm          grpsvcs          14332   active
 Status information for subsystem grpglsm:
 Connected to Group Services.
 Subscribed: Yes  Joined: Yes  (Number of aliases: 4)
 Switch device /dev/css0 not currently open on this node.
  Switch device opened 23 times, closed 23 times.
```

The basic function that the grpglsm daemon provides is to report locally whether the node is functioning over the css interface. The grpglsm daemon subscribes to the *Adapter Membership Group* cssRawMembership, which is provided internally by topsvcs, and if it finds that it is locally connected, it joins the cssMembership group, which we see in the `lssrc -ls grpsvcs` output.

The information listed in the `lssrc` output when run against grpglsm is telling us the current status of the subscription and join, specifically, in the fields Subscribed and Joined.

The last important piece of information is the number of aliases. This refers to the total number of IP aliases that grpglsm is attempting to heartbeat over. In our case, we have the following defined on our node:

```
# netstat -in -I css
Name  Mtu   Network     Address              Ipkts Ierrs    Opkts Oerrs  Coll
css0  65520 link#5                         346824     0   322082     0     0
css0  65520 192.168.14  192.168.14.13      346824     0   322082     0     0
css0  65520 140.40.4    140.40.4.13        346824     0   322082     0     0
```

The same setup applies to the only other node in the cluster, making a total of four aliases that are currently attempting to heartbeat.

Event Management Daemon

Similar output can also be obtained from the Event Management daemon (emsvcs) when we run the `lssrc` command against it.

Here we see the top of the output from the event management daemon (emsvcs):

```
Subsystem         Group            PID     Status
 emsvcs           emsvcs           4038    active

Trace flags set:  None

 Configuration Data Base version: 887653462,846074368,0(SDR)

 Daemon started on 07/20/98 at 17:14:28.889124864
    running 0 days, 23 hours, 44 minutes and 23 seconds
 Daemon connected to group services: TRUE
 Daemon has joined peer group:       TRUE
 Daemon communications enabled:      TRUE
 Peer count:                         1
```

We see that the output from emsvcs is similar to the output of the grpglsm daemon when `lssrc` was run against it. The emsvcs daemon has connected to Group Services and has joined its own group, ha_em_peers.

AIX Operating System Resource Monitor Daemon

Finally, we see the output from the AIX operating system resource monitor daemon (emaixos):

```
# lssrc -ls emaixos
Subsystem         Group            PID     Status
 emaixos          emsvcs           4922    active

Trace Level:         None
Domain Type:         HACMP
Domain Name:         cluster_a
RMAPI Initialized:   TRUE
Data Initialized:    TRUE
Data Init. Attempts: 1
Data Init. Delay:    5
Inst. Interval:      600
Inst. Count:         143
SRC FD:              3
Server FD:           7
Class Count:         7
Variable Count:      41
```

The sample output shows us that the emaixos daemon has correctly identified that it is running in an HACMP domain and lists the Domain Name as the original cluster name we specified when configuring HACMP, that is, cluster_a.

4.2.2 Commands for HACMP/ES

In this section we look at the commands more specific to HACMP/ES. They reside in the directories /usr/es/sbin/cluster and /usr/es/sbin/cluster/utilities. For compatibility with HACMP for AIX, symbolic links have been established to the previous locations under /usr/sbin/cluster and /usr/sbin/cluster/utilities.

For the most part, the commands that we examine are called from higher levels, like the System Management Interface Tool (SMIT), and are not normally executed on their own. However, for the purposes of problem determination, it is useful to know what the commands being called under the covers are actually doing.

The specific commands and utilities we examine, which are either new to HACMP/ES or considered relevant, are:

- clstat
- clinfo
- clhandle
- clmixver
- claddnetwork
- cldomain
- clgetesdbginfo
- cllsgnw
- claddcustom

4.2.2.1 clstat

The `clstat` utility used for monitoring the status of HACMP clusters is functionally the same as the previous versions included in HACMP/ES V4.2.2 and HACMP for AIX V4.2.2.

The 32-node support remains the same. Compatibility with previous versions of HACMP is also maintained.

Further information regarding the `clstat` utility can be found in *HACMP for AIX, Version 4.3: Administration Guide*, SC23-4279.

4.2.2.2 clinfo

The Client Information Daemon (clinfo) is also functionally coded the same as the previous versions from HACMP/ES and HACMP for AIX.

Clinfo provides information about the current state of the cluster or multiple clusters to client applications, for example the `clstat` command. Clinfo uses a shared memory segment to track the topology of specified clusters. It consists of a daemon that updates the shared memory by communicating with the Cluster Smux Peer Daemon (clsmuxpd), and an Application Programming Interface (API) that client applications communicate with to obtain information about the cluster.

There are two types of broadcast events received by Clinfo:

- Topology Maps
- Topology Events

Topology Maps contain information that completely describes a node, while Topology Events track the transition of the cluster through various states. States that are tracked include:

- The cluster has become stable.
- The cluster has become unstable.
- A network has failed.
- An interface on a node has failed.
- A node has joined the cluster.
- A node has failed or left the cluster.
- A new primary node has been selected.

The main body of Clinfo is a select() loop that processes incoming events. During inactive cycles Clinfo calls a context checking function. This function does one of the following:

- Communicates with at least one node in an "up" cluster to ensure the cluster has not been lost.
- Attempts to communicate with "down" clusters to determine if they have become active or reachable.

Currently the daemon allocates a memory segment based on an 8-node cluster by default. If more storage is required, Clinfo increases the shared memory segment dynamically. The current upper limits of the segment are:

- Maximum number of clusters is 16.
- Maximum number of nodes within a cluster is 128.
- Maximum number of interfaces per node is 128.

These limits are likely to be increased to accommodate greater than 128-node clusters.

Further information regarding Clinfo can be found in *HACMP for AIX, Version 4.3: Administration Guide*, SC23-4279.

4.2.2.3 clhandle

The `clhandle` command is used to obtain the *Cluster Node Handle* of the node it is executed on, or return the *Cluster Node Handle* associated with a given *Cluster Node Name* or vice versa. It can also be used to display the *Cluster Node Handle* and *Cluster Node Name* of all nodes in the cluster configuration. The Cluster Node Handle is now the field that HACMP/ES V4.3 uses to identify the node. In previous versions HACMP used the node_id field in the ODM.

This information is extracted from the HACMPnode and HACMPcluster object classes of the Global Object Data Manager (GODM), and is required in order for the user to define instance vectors for user-defined events. Cluster node numbers will be in the range of 1 to 2048. Refer to “HACMPnode” on page 21 and “HACMPcluster” on page 20, for more detail.

The `clhandle` command will be an undocumented command. By which we mean it will not appear in the InfoExplorer or manual pages and will be provided on an as-is basis only.

The `clhandle` accepts the following flags:

-a	Displays every node handle and node name defined in the cluster.
-c	Displays the node handle and node name in colon-separated output for this node.
-**n [nodename]**	Display the node handle for the specified nodename.
-h [nodehandle]	Display the node name for the specified nodehandle.
[no flags]	Display the node handle and node name for this node.

The possible return codes are:

0	Success
!0	Failure

4.2.2.4 clmixver

The `clmixver` command is a new command that is used to determine if the local node that the command is executing on is at the same or later level than the cluster manager that ran the last topology synchronization.

The `clmixver` command will also be an undocumented command provided on an as-is basis only.

The `clmixver` command does not have any flags that can be passed to it, and has the following return codes:

1	This node is executing a version greater than the version of clstrmgr that performed the last topology synchronization.
0	This node is executing the same version as the clstrmgr that performed the last topology synchronization.
-1	An error has occurred. An example could be that the cluster version is greater than the version running on this node.

The information concerning the cluster version and the node version is held in the HACMPcluster and the HACMPnode GODM class. Refer to “HACMPcluster” on page 20 and “HACMPnode” on page 21, for more detail.

4.2.2.5 claddnetwork

The `claddnetwork` command is a new command in HACMP/ES V4.3. After configuring the cluster with all of the adapters, the individual subnets of a network can be combined to create a global network for heartbeat using the `claddnetwork` command. Therefore, for a global network, when the last adapter within a subnet goes down, a network down event will not occur. Likewise, should the first adapter on a subnet of the global network come up, a network up event will not run.

The concept of a global network was designed most specifically for the SP Ethernet. It was created for specific situations where, with some larger configurations, a node in each frame acts as the boot install server (BIS) for that frame (an example of this can be seen in Figure 12 on page 64). It was decided, however, not to limit the global network functionality to the SP Ethernet, as it may be useful on other networks.

The problem we are trying to avoid is a partitioned cluster. If we attempt to bring all the nodes up at the same time, we may encounter a partitioned system due to their subnetting. This subsequently means that one of the partitions, the one with the least number of nodes, will die and have to rejoin the cluster. The larger the system becomes, the more cumbersome and time consuming this becomes. If, however, we define the entire SP Ethernet as a global network, then all nodes will be able to heartbeat to each other, even over the subnetworks. Ultimately avoiding the partitioned system.

Refer to 6.1, “Global Network” on page 131 for details on the procedure involved when defining a global network.

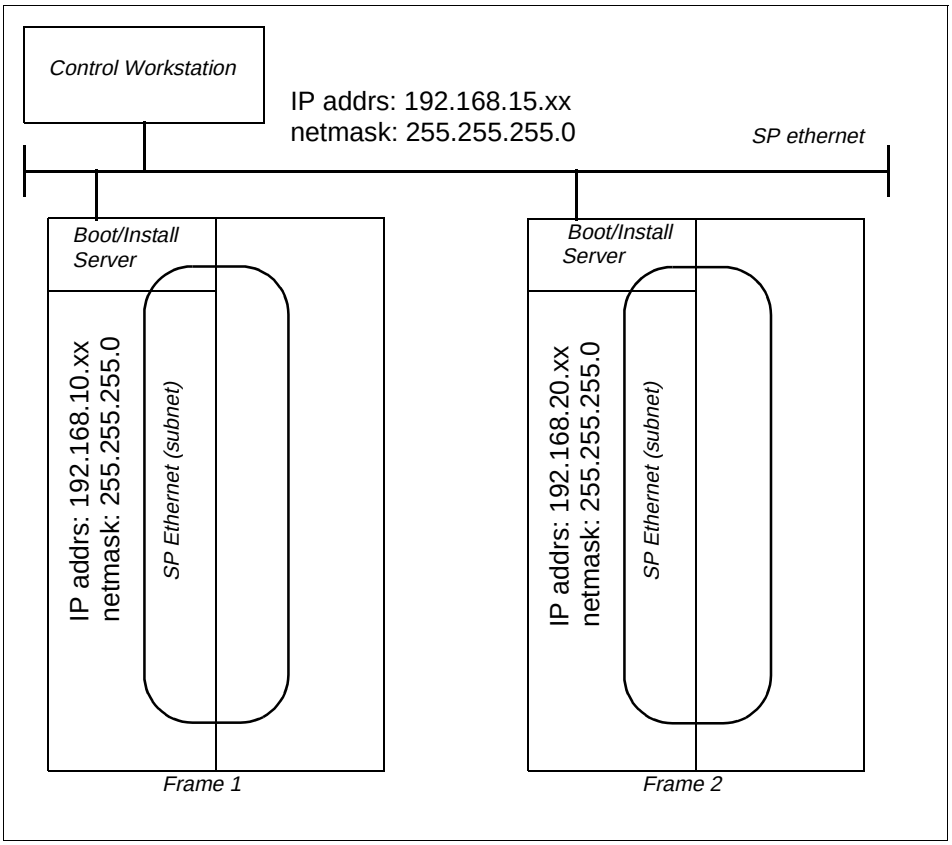


Figure 12. An Example of a Global Network

The `claddnetwork` command accepts the following flags:

-u local_network[:global_network] Add or remove a global network.

The return codes are:

0 Success

!0 Failure

4.2.2.6 cldomain

The `cldomain` command is another new command for HACMP/ES V4.3 that is also undocumented. The command is used to interrogate the GODM, specifically the HACMPcluster GODM class, to obtain the Cluster Name, which defines the domain for the HACMP infrastructure. If the command is

successful, the Cluster Name is displayed, otherwise a non-zero failure code is displayed.

4.2.2.7 clgetesdbginfo

The `clgetesdbginfo` script is for future use and is not used.

4.2.2.8 cllsgnw

The `cllsgnw` command is a new command with HACMP/ES V4.3 used to obtain information about local and global networks. The command interrogates the HACMPnetwork object class to determine the names of the local and global network names.

The command accepts the following flags:

-a	Displays all defined local and global network names.
-c	Displays the defined local and global network names in colon-separated output for this node.
-n [networkname]	Displays the defined local and global network names for the specified networkname.
-g [globalnetwork]	Displays the defined local and global network names for the specified global network.

4.2.2.9 claddcustom

The `claddcustom` command existed previously under HACMP for AIX V4.2.2 and HACMP/ES V4.2.2 but was undocumented. This command is used for adding a custom verification method.

The command accepts the following flags:

-n	Verification method name
[-I description]	Verification method description
[-v value]	Verification script file name

4.2.3 Commands for RSCT

In this final command section we examine the commands and scripts that are specific to RSCT. Some of the scripts, in particular the daemon control scripts, formally resided with the HACMP/ES commands and utilities. However, since the actual code that is now executing for daemons such as topsvcs and High Availability Topology Services (hats) is now exactly the same, but just executing in a separate domain, they are located with the rest of the RSCT-specific commands.

Before we examine the commands and scripts in more detail we need to explain some of the nomenclature we are going to use. We have two terms that need to be defined, Name Server (NS) and Group Leader (GL).

4.2.3.1 Name Server

Group Services must be able to track all the groups that its clients want to form. To do this, a group name server is established within a domain. In this case, each domain's boundary is defined by the HACMP/ES cluster. There will be one name server for each domain, responsible for keeping track of all groups created in that domain.

The node with the lowest node handle is elected as the name server for the domain. To ensure that only one node becomes the name server, Group Services waits for Topology Services to inform it which nodes are currently running. Based on this information, each grpsvcs daemon finds the node with the lowest node handle. It then compares the value with its own node handle value and either sends a message to the name server, or awaits messages if it is the name server.

Once the Group Services domain has been established, requests from client processes to create, join, or subscribe to groups can be processed.

4.2.3.2 Group Leader

For each group that is registered with Group Services, one of the grpsvcs daemons is designated as the group leader. Each group leader is responsible for the following operations within its own group:

- Managing the group information, such as membership lists and group state information.
- Managing the group meta data, such as node addresses and group protocol information.

The group leader is elected by the first node joining or forming the group. Subsequent joins form the hierarchy for taking over as the group leader should it fail. We can see the hierarchy with the `hagsmg` command, which is examined in more detail in 4.2.3.8, “hagsmg” on page 78. Group Services maintains replicated data across nodes on which the providers to the group reside, so that another node is able to take over should the group leader fail.

Now that we have clarified this terminology, we can examine the commands available. The specific commands and scripts we are going to look at in more detail are:

- topsvcsctrl

- grpsvcsctrl
- emsvcsctrl
- hagscl
- hagsgr
- hagsmg
- hagsns
- hagspbs
- hagsvote
- hagscounts
- hagsp
- hagsreap

First, we look at the daemon control scripts. An important point to note is that some of will be familiar with the original control scripts that originated with IBM Parallel System Support Programs for AIX (PSSP). These were often used on systems to correct minor problems within the groups and may still be utilized in this fashion. However, this is not the case for the HACMP/ES V4.3 versions of these scripts. They are called when a change is propagated through HACMP, and should not be invoked manually; otherwise we may see some disruptive effect. A typical example of this is: if the topsvcs daemon is stopped without cluster manager invocation, we no longer receive any heartbeats and HACMP causes the node to halt.

4.2.3.3 topsvcsctrl

The `topsvcsctrl` script is used to manage the topsvcs daemon as an SRC subsystem. It is based on the `hatsctrl` script formerly used in the High Availability Infrastructure (HAI) or Pheonix Infrastructure that was originally introduced with PSSP V2.2. The `hatsctrl` script still exists in RSCT today, but only manages the Topology Services daemon in the PSSP domain, and its name is hats.

The `topsvcsctrl` script is called with one or more of the following flags:

-a Add Topology/Group Services

-d Delete Topology Services

-k Stop Topology Services

-c Clean Topology Services

-r Refresh the Topology Services configuration

-s Start Topology Services

-t Turn on tracing for Topology Services

-o Turn off tracing for Topology Services

-h|? This help message

4.2.3.4 grpsvcsctrl

As with the `topsvcsctrl` script, the `grpsvcsctrl` script was originally created in the PSSP code to manage the Group Services daemon (hags) in the PSSP domain, called `hagsctrl`. Again, this script still exists in RSCT but is now used to manage the Group Services daemon running in the PSSP domain. Hence the `grpsvcsctrl` script was created to manage the Group Services daemon (grpsvcs) running in the HACMP domain.

The `grpsvcsctrl` script is called with at least one of the following flags:

-a Add the Group Services subsystem to this partition

-s Start the Group Services subsystem in this partition

-k Stop the Group Services subsystem in this partition

-d Delete the Group Services subsystem from this partition

-c Remove the Group Services subsystem from all partitions (but do not remove SDR objects)

-u Remove the Group Services subsystem from all partitions

-t Start the Group Services subsystem trace in this partition

-o Stop the Group Services subsystem trace in this partition

-r Refresh the Group Services subsystem in this partition (currently, Group Services refresh does nothing)

-b Tell the Group Services subsystem this partition has begun to be migrated

-g Tell the Group Services subsystem this partition has completed being migrated

-h Display this usage statement

4.2.3.5 emsvcsctrl

The `emsvcsctrl` script was, again, originally created in the PSSP code to control and manage the Event Management daemon (haem) running in the PSSP domain, called `haemctrl`. Like the other control scripts, both the HACMP domain and PSSP domain versions exist in RSCT. The `emsvcsctrl` script manages the Event Management daemon (emsvcs) running in the

HACMP domain and the AIX Operating System Resource Monitor (emaixos) as SRC subsystems.

The `emsvcsctrl` script is called with one of the following flags:

- **-a** Add the Event Management subsystem to the HACMP domain
- **-A** Add the Event Management subsystem and AIX OS Resource Monitor to the HACMP domain
- **-s** Start the Event Management subsystem in the HACMP domain
- **-k** Stop the Event Management subsystem in the HACMP domain
- **-d** Delete the Event Management subsystem from the HACMP domain
- **-c** Remove the Event Management subsystem from the HACMP domain
- **-t** Start the Event Management subsystem trace in the HACMP domain
- **-o** Stop the Event Management subsystem trace in the HACMP domain
- **-r** Refresh the Event Management subsystem in the HACMP domain (currently, Event Management refresh does nothing)
- **-h** Display this usage statement

4.2.3.6 hagscl

The `hagscl` command is another command that existed previously within PSSP. However, it was, and still is, undocumented and is provided without support.

The purpose of this command is to extract further information concerning the client processes attached to Group Services. If we refer back to the sample output from `lssrc` in Figure 11 on page 56, we see that there are three locally connected clients and one local provider per group. However, we cannot link the exact processes directly to the existing groups or obtain any further information concerning these processes. The `hagscl` command provides us with this type of information, for example:

```
# hagscl -ls grpsvcs
Client Control layer summary:
  Number of clients connected: 3
  Cumulative number of clients connected: 3
  Total number of client requests: 3
  Number of client hash table conflicts: 0

--------------------------------------------------
Client: socketFd[10] pid[16538]Total number of Clients: 3
 Client initialized: pid: 16538
  uid/gid/version: [0/200/4]
  client directory: [SuppName: length: 26 value:
/var/ha/run/haem.cluster_a

Number of local providers/subscribers: 1/0
Responsiveness information for Client: socketFd[10] pid[16538]
Type[ type[HA_GS_PING_RESPONSIVENESS]] interval[120] response time limit[120]
Checks done/bypassed[56/0] lastResponse[OK]]
Results(good/bad/late)[56/0/0]
Membership list:
slot    info
0       [{provider}Member token[0] Client: socketFd[10] pid[16538]ProviderId[1/1
]]
--------------------------------------------------
Client: socketFd[11] pid[16144]Total number of Clients: 3
 Client initialized: pid: 16144
  uid/gid/version: [0/0/4]
  client directory: [SuppName: length: 29 value:
/var/ha/run/grpglsm.cluster_a

Number of local providers/subscribers: 1/1
Responsiveness information for Client: socketFd[11] pid[16144]
Type[ type[HA_GS_PING_RESPONSIVENESS]] interval[3630] response time limit[10]
Checks done/bypassed[1/0] lastResponse[OK]]
Results(good/bad/late)[1/0/0]
Membership list:
slot    info
0       [{subscriber}Member token[0] Client: socketFd[11] pid[16144]]
1       [{provider}Member token[1] Client: socketFd[11] pid[16144]ProviderId[0/1
]]
--------------------------------------------------
Client: socketFd[12] pid[17008]Total number of Clients: 3
 Client initialized: pid: 17008
  uid/gid/version: [0/0/3]
  client directory: [SuppName: length: 1 value:
/

Number of local providers/subscribers: 1/2
Membership list:
slot    info
2       [{provider}Member token[2] Client: socketFd[12] pid[17008]ProviderId[0/1
]]
3       [{subscriber}Member token[3] Client: socketFd[12] pid[17008]]
4       [{subscriber}Member token[4] Client: socketFd[12] pid[17008]]
```

In the output we have a number of fields that are of interest. First, we have the overall summary, similar to that seen in the `lssrc` output.

Next, we have the information by each group within Group Services. This starts with the socket file descriptor (socketFd) and process identifier (PID) along with the total number of clients for the group. We can also see at this point which client has *initialized*. By initialized we mean the client process has successfully executed the ha_gs_init() subroutine, which is part of the Group Services Application Programming Interface (GSAPI). All processes that wish to register with Group Services must call the ha_gs_init() subroutine to set up a connection with GSAPI and subsequently Group Services. For further information on the GSAPI subroutines and how to utilize them, refer to *RS/6000 SP High Availability Infrastructure*, SG24-4838 or *IBM RS/6000 Cluster Technology for AIX: Group Services Programming Guide and Reference,* SA22-7355.

We also see the User Identifier (UID), Group Identifier (GID) and version number of the process running, along with the default directory for the process. It is worth noting here that the UID will always be 0, since only processes with root authority can make use of the ha_gs_init() subroutine.

Next, we have a summary for the group showing the total number of subscribers and providers. This entry is followed by the client responsiveness, which tells us how often GSAPI checks the client for a response. This is defined when the process registers with GSAPI with the ha_gs_init() subroutine. Let us take a closer look:

```
Responsiveness information for Client: socketFd[10] pid[16538]
Type[ type[HA_GS_PING_RESPONSIVENESS]] interval[120] response time limit[120]
Checks done/bypassed[56/0] lastResponse[OK]]
Results(good/bad/late)[56/0/0]
```

Note that the process 16538, which happens to be the Event Monitoring daemon for HACMP, has an interval of 120 seconds between checks, with 56 checks passed so far.

The final part of our output is the membership list. This displays the current number of groups this client process is providing and subscribing to. For example:

```
2       [{provider}Member token[2] Client: socketFd[12] pid[17008]ProviderId[0/1
]]
3       [{subscriber}Member token[3] Client: socketFd[12] pid[17008]]
4       [{subscriber}Member token[4] Client: socketFd[12] pid[17008]]
```

The output is telling us that PID 17008 is a provider to one group but is also subscribing to two groups. On its own we cannot gather much information. However, once we involve the hagsgr command (refer to 4.2.3.7, “hagsgr” on page 73), we are able to distinguish which groups the client process is actually subscribed to by using the socketFd and the node number.

The question that may arise is, why would a process be a provider and a subscriber, and, as in the previous example, more than once?

To answer this question we need to take a look from a higher level. In Figure 13 on page 72, we see an overall schematic of how the different daemons interact with each other.

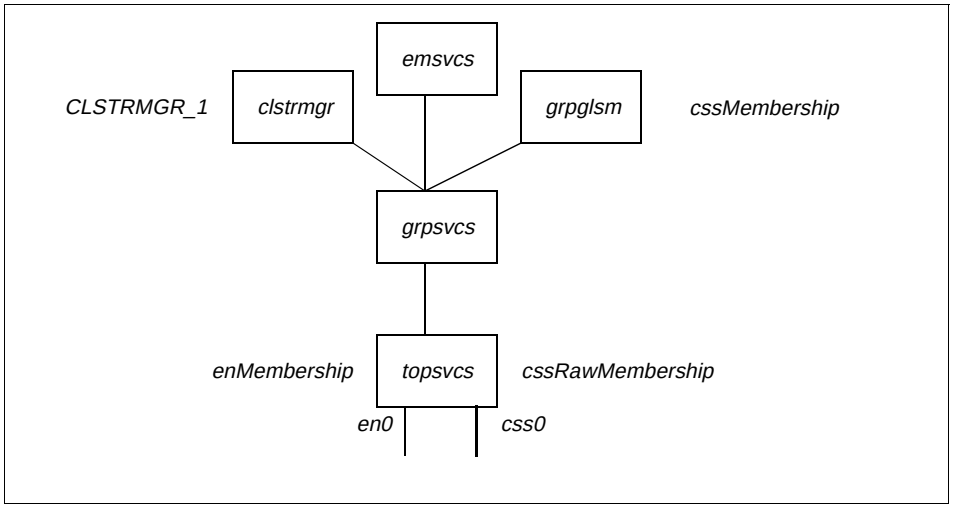


Figure 13. Interaction of Daemons

The groups listed, enMembership and cssRawMembership, are groups we have not seen until now. They are known as *Adapter Membership Groups,* provided internally by topsvcs. They are subscribed to by the appropriate daemons, grpglsm for cssRawMembership and cluster manager (clstrmgr) for enMembership and cssRawMembership. If the local adapters are in the Adapter Membership Groups, they will join or create their own group,

cssMembership for grpglsm and CLSTRMGR_1 for clstrmgr. It is these groups that we see listed in the output from the `lssrc` command, that is, the groups that have a listed process providing to them.

The `hagscl` command accepts the following flags:

[-h host] [-l] -g group_name Gets group status

[-h host] [-l] -s subsystem_name Gets subsystem status

[-h host] [-l] -p subsystem_pid Gets status by PID

4.2.3.7 hagsgr

The `hagsgr` command is also an undocumented command that originally existed for PSSP Group Services. It can be used to display information concerning the groups within Group Services, their status, and their providers and subscribers. The command only lists the groups in which the node specified is a provider, a subscriber, or both.

An example output from `hagsgr` may look similar to:

```
# hagsgr -s grpsvcs
  Number of: groups: 6
Group slot # [0] Group name[HostMembership] group state[Not Inserted |]
Providers[]
Local subscribers[]

Group slot # [1] Group name[ha_em_peers] group state[Inserted |Idle |]
Providers[[1/1][1/2]]
Local subscribers[]

Group slot # [2] Group name[cssRawMembership] group state[Idle |]
Providers[[1/1][3/1][1/2][3/2]]
Local subscribers[[11/1][12/1]]

Group slot # [3] Group name[cssMembership] group state[Inserted |Idle |]
Providers[[0/1][0/2]]
Local subscribers[]

Group slot # [4] Group name[enMembership] group state[Idle |]
Providers[[2/1][0/1][1/1][0/2][2/2][1/2]]
Local subscribers[[12/1]]

Group slot # [5] Group name[CLSTRMGR_1] group state[Inserted |Idle |]
Providers[[0/1][1/2]]
Local subscribers[]
```

What do each of the fields tell us?

The group slot is not important - it is an internal index array number not used externally. This is followed by the group name, which is the external name that processes wishing to provide or subscribe to must use.

We also have the most important field, the group state, which can have the following values:

Not inserted This indicates that this group is not active on this node and therefore should not have any providers or subscribers.

Insert Pending This means the node is currently trying to insert into the group. It has sent the Group Services name server (NS) a *lookup* and is awaiting a response, or alternatively the node has received a response and is waiting to be inserted into the *shadow* of the group, known as the meta group (see “What is a meta group?” on page 78 for more details). We can use the hagsmg or hagspbs commands to determine which of these we are actually waiting upon. However, this is a temporary state that under normal circumstances should not remain.

Inserted This tells us that the group is currently active on the node specified and may have providers, subscribers, or both. If the status is inserted, we may also see a further substatus of the following:

Idle Indicates that a group is not currently running any protocols.

Running Protocol

Indicates the group is running a protocol. If we use the command with the long (-l) flag, we can see which protocol is running. As an example, here we can see that a sample group we have created, *the SourceGroup,* is running a protocol:

```
# /usr/sbin/rsct/bin/hagsgr -a theSourceGroup -s grpsvcs
Wed Jul 22 15:31:34 EDT 1998
  Number of: groups: 7
Group name[theSourceGroup] group state[Inserted |Running Protocol |]
Providers[]
Local subscribers[]
```

If we take the corresponding output from hagsgr command with long (-l) flag, and examine the

protocol information, this shows us the following output:

```
Wed Jul 22 15:31:34 EDT 1998
  Number of: groups: 7
Information for SGroup: Group name[theSourceGroup]
.
.
.
Protocol Manager summary information:
Current count: 0
total count: executed/approved/rejected[2/1/1]
failure count: executed/approved/rejected(explicit/implicit)[0/0/0(0/0)]
join count: executed/approved/rejected[1/0/1]
expel count: executed/approved/rejected[0/0/0]
attribute change count: executed/approved/rejected[0/0/0]
leave count: executed/approved/rejected[0/0/0]
state change count: executed/approved/rejected[0/0/0]
PBM count: executed/approved/rejected[0/0/0]
source reflection count: executed/approved/rejected[0/0/0]
subscription count: executed/approved/rejected[0/0/0]
announcement count: executed/approved/rejected[1/1/0]
--------------------
No transient protocol
--------------------
Currently executing protocol: SJoinProtocol: requested by: [{provider}Member tok
en[0] Client: socketFd[13] pid[9686]ProviderId[100/1]]
SVProtocol: state[Submitted+Executing+ExecutingPostBroadcast+ExecutingNeedsVotes
+Continuing]
ProtocolToken[7/13]
[proposer:[{provider}Member token[0] Client: socketFd[13] pid[9686]ProviderId[10
0/1]]
][group:Group name[theSourceGroup]]
Number phases[2] this phase[1]
summary code[0] time limit[60]

[Batching allowed]
Changing count [1] local changing count [1] changers removed count [0]
Have [32] changing member slots, list:
SProvider(ProviderId[100/1]conditionalListPosition[0]
SVSuppMember: [owned by:Client: socketFd[13] pid[9686]] token[0] status[NotIn ]
name[SMemberName: (min/max)length: (1/16)10 value: SourceJoin]
 supp ptr: 0x300be7c8 group ptr: 0x300bf288 groupListPosition: -1 nodeListPositi
on: -1 Need Vote/Voted Yet[1/0]
Voting Protocol:0x300c05e8
 [votingParticipant])[end SProvider]
--------------------
Unsent queue:[No entries]
--------------------
Sent queue:[No entries]
--------------------
Failure queue:[No entries]
--------------------
Join queue:[No entries]
--------------------
Subscribe queue:[No entries]
--------------------
Announcement queue:[No entries]
```

Figure 14. The hagscl Command Output

We can see that the current executing protocol is `SJoinProtocol`, which is obviously a group join request in progress.

Needs Priming The node is attempting to become active within the group and is waiting to find out the current status regarding the membership and group state values. Again, during normal operations this is a temporary state.

Waiting for BroadcastSent

The node is currently sending a broadcast.

Resending Requests

The group's Group Leader (GL) has failed, and Group Services is currently in recovery status moving to a new GL.

The last piece of information supplied to us for each group, is the subscriber and provider information. This is the point at which we can link up the information seen from the `hagscl` output in Figure 14 on page 76.

The providers are listed as:

[provider_instance_number / provider_node_number]

The provider_instance_number is specified by the client process when it joins the group, and the provider_node_number is the HACMP node number that the client process is running on.

The providers are listed in the order that they joined the group. Therefore, the oldest is first and the youngest last.

The local subscribers are listed as:

[socket_file_descriptor / subscriber_node_number]

The socket file descriptor is the Group Services daemon socket file descriptor that connects to the client process that subscribed to the group. This will match the socketFd listed in the output from `hagscl` for the same node. Thereby, providing a means to determine which process is subscribing to which group. The subscriber node number is, again, the HACMP node number on which the subscribing process is running.

We can also see from the output that we are *Inserted* into the three groups that were originally displayed in the `lssrc -ls grpsvcs` output in Figure 11 on page 56: the ha_em_peers, cssMembership and CLSTRMGR_1 groups. Each of these groups has one provider process for each node in the cluster.

For the two groups that we are not *Inserted* into, we have a number of providers for each node. As previously mentioned, these providers come internally from Topology Services and are defined by the number of networks we configure to HACMP. As an example, let us look specifically at the enMembership group information:

```
Group slot # [4] Group name[enMembership] group state[Idle |]
Providers[[2/1][0/1][1/1][0/2][2/2][1/2]]
Local subscribers[[12/1]]
```

This shows that we have three provider instances on each node in the cluster. We can match this information with the local Ethernet-specific information provided from the `lssrc` output on Topology Services:

```
Network Name   Indx Defd Mbrs St Adapter ID      Group ID
ethernet1_0    [ 0]    2    2  S 128.100.10.1    128.100.10.3
ethernet1_0    [ 0]              0x85a6258a      0x85a6276d
HB Interval = 1 secs. Sensitivity = 4 missed beats
ethernet1_1    [ 1]    2    2  S 128.200.20.2    128.200.30.3
ethernet1_1    [ 1]              0x85a6253c      0x85a6271b
HB Interval = 1 secs. Sensitivity = 4 missed beats
spether_0      [ 2]    2    2  S 192.168.4.15    192.168.4.15
spether_0      [ 2]              0x45a6253d      0x45a6271b
HB Interval = 1 secs. Sensitivity = 4 missed beats
```

We have three networks listed, hence three heartbeating networks and our three providers from Topology Services to Group Services.

The `hagsgr` command accepts the following flags:

[-h host] [-l] [-a argument] -g group_name	Gets group status
[-h host] [-l] [-a argument] -s subsystem_name	Gets subsystem status
[-h host] [-l] [-a argument] -p subsystem_pid	Gets status by PID

4.2.3.8 hagsmg

The `hagsmg` command is another command that existed undocumented in PSSP. It is used to list all the *meta groups* and the nodes that belong to them.

What is a meta group?

A group and a meta group are ultimately one and the same. However, they may appear very different depending on how they are viewed. There are two ways to view a group: from the client level or from the Group Services level.

In Figure 15 on page 79 we see how the different layers of IBM RS/6000 Cluster Technology (RSCT) interact with each other.

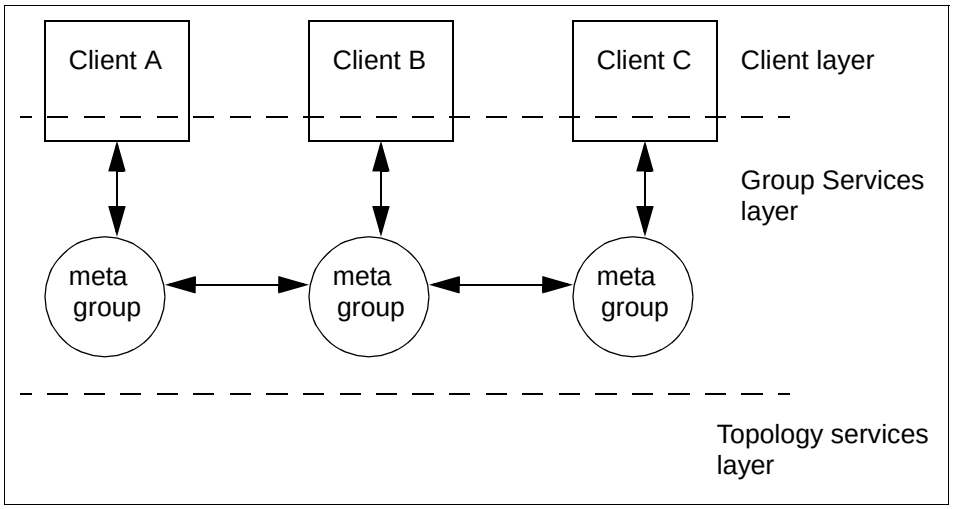


Figure 15. The Layers of RSCT

When we refer to clients in Figure 15 on page 79, we mean any process that is a provider or subscriber to a group in Group Services, for example the cluster manager for HACMP/ES.

When we view the group from the client layer, with commands such as `lssrc`, we only see the information relevant to that layer. Again, this is why we do not see information concerning the default groups provided by Topology Services.

For each client group spread across the nodes, we have an equivalent meta group in the Group Services layer. The meta group has two components, the list of nodes that belong to the group, managed by the meta group layer, and a *reliable broadcast message stream*, which includes notifications, join requests and so on, managed by the Pheonix Broadcast System (PBS). This information is replicated across all members of the group, so that should the group leader fail, another member will be able to take over with minimum disruption. The group state is not managed by the meta group, it is managed in the client layer.

It is worth noting that only groups where the node, queried with the `hagsmg` command, has been inserted, are displayed.

A sample output from using the `hagsmg` command is shown here:

```
# hagsmg -s grpsvcs
1.1 cssMembership: 1 2
2.1 ha_em_peers: 1 2
3.1 CLSTRMGR_1: 1 2
0.Nil ZtheNameServerXY: 1.8 2.15
0.Nil theGROVELgroup:
```

The first column provides the internal group identifier for each meta group and the second column provides the group name. The group name is used externally by processes wishing to join the group, and along with the group identifier can be matched to output in the log file in /var/ha/log.

We can see from the output that there are two groups that have not appeared elsewhere. These are:

ZtheNameServerXY This group is used by the Group Services name server to manage the domain and should have an entry for all nodes. Some entries may have the node number followed by the word *Nil*. This is normal since this information is only relevant to the Name Server.

theGROVELgroup This is an internally used group that does not list any nodes.

After each group identifier and group name we see the list of nodes inserted into the group. The list was generated as the nodes joined the group with the oldest first, down to the youngest. The list also defines the hierarchy for takeover, by which we mean that the first node in the list is the group leader, and should this node fail, the next node listed in the group will assume the role of group leader.

The hagsmg command accepts the following flags:

[-h host] -g group_name Gets group status

[-h host] -s subsystem_name Gets subsystem status

[-h host] -p subsystem_pid Gets status by PID

4.2.3.9 hagsns

Once again, the hagsns command existed as an undocumented command of the PSSP high availability infrastructure, now part of RSCT. The command is most likely one of our first stages in problem determination as it displays output pertaining to the Group Services name server status. It is especially useful in ascertaining which nodes have connected correctly to the nameserver, thereby establishing the domain.

Let us take a look at a sample output from an HACMP/ES cluster node:

```
# hagsns -s grpsvcs
We are: 1.8 pid: 4290 domaindId = 1.8 noNS = 0 inRecovery = 0 CodeLevel = RSCT 1
.0
NS::ENsState(7):kBecomeNS protocolInProgress = NS::ENsProtocol(0):kNoProtocol ou
tstandingBroadcast = NS::ENsBroadcast(0):kNoBcast
Process started on Jul 14 12:10:37, (22:57:12) ago.  HB connection took (0:0:0).
Initial NS certainty on Jul 14 12:11:38, (22:56:12) ago, taking (0:1:0).
Our current epoch of certainty started on Jul 14 12:11:38, (22:56:12) ago.
1 UP nodes: 1
1.1 cssMembership: GL: 1 seqNum: 0 theIPS: 1 lookupQ:
2.1 ha_em_peers: GL: 1 seqNum: 0 theIPS: 1 lookupQ:
3.1 CLSTRMGR_1: GL: 1 seqNum: 0 theIPS: 1 lookupQ:
```

Figure 16. The hagsns Command Output

Because the `lssrc` command is a daemon summary, it may not always provide us with the information we require when Group Services is not functioning as expected. This is an occasion when the `hagsns` command can be used to extract more detailed information. The previous screen shot shows us a typical output.

In the first line we see which node and instance number of Group Services is running locally, in this case `1.8`, followed by the PID of the Group Services daemon. We can also see that the domainID has the same value as the server responsible for maintaining the domain. This is corroborated by the next fields, `noNS` and `inRecovery`, which have a value of 0, meaning we have a Name Server and we are not in recovery.

The second set of fields provides us with information regarding the current state of the Name Server and any broadcasts or protocols being executed.

Currently no protocols or broadcasts are being sent by the name server. In the next example we see what the temporary state of the group looks like when a node fails to respond, in this case the name server:

```
# hagsns -s grpsvcs
Wed Jul 22 16:17:52 EDT 1998
We are: 1.15 pid: 14582 domaindId = 2.25 noNS = 1 inRecovery = 1 CodeLevel = RSC
T 1.0
NS::ENsState(11):kRecoverAscend protocolInProgress = NS::ENsProtocol(0):kNoProto
col outstandingBroadcast = NS::ENsBroadcast(0):kNoBcast
Process started on Jul 22 16:11:17, (0:6:34) ago.  HB connection took (0:0:0).
Initial NS certainty on Jul 22 16:11:24, (0:6:27) ago, taking (0:0:6).
Our current epoch of uncertainty started on Jul 22 16:17:44, (0:0:7) ago.
1 UP nodes: 1
Coronation Timer has NOT popped!
Domain not recovered for (0:0:7).  Waiting to hear from (check hats & hags on) t
hese 1 nodes: 2
```

We see that we are now *inRecovery* since the value equals 1, and our current state has changed to (11):KRecoverAscend. This should be a temporary state, possibly caused, as it was in this case, by stopping HACMP/ES on our second node.

Finally, referring back to our original output from hagsns command in Figure 16 on page 81, we see the various data for the overall group status. For each statement we have an initial time stamp and a total length of time with the format (HH:MM:SS).

We also have a stable group at this time with only one node, our current node, which is the name server. The node is a provider in each of the groups listed. The fields have the following meanings:

<**group identifier**> Is the internally allocated identifier given to each Group Services group.

<**group name**> Is the external name allocated to each group, used by clients when they wish to join or subscribe to the group.

GL Is the HACMP node number of the current group leader, which manages the primary functions of the group.

seqNum Is a sequence number for messages within each group. Each group maintains its own sequence of message broadcasts. The value shown here was the next message in the sequence when the current group leader took ownership of the group. Since our value is 0, it is likely we have had the same group leader since Group Services was started.

theIPS Is the list of nodes that have expressed an interest in a group. Normally this is a result of a client process requesting to join or subscribe to a group. A node expresses an interest in a group by sending a *lookup* request to the domain's name server, and is placed on the `theIPS` list when the name server sends out the response to the lookup.

lookupQ Is another list of nodes, displaying which nodes have sent a lookup request, but have not yet had a response from the domain's name server sent out. There should only be a lookup queue list when there is no group leader for the group, for example, when the group is first being formed or when the group leader has failed and recovery is in progress.

On a node that is not the name server, we receive similar output, except that no group information is displayed since it is not responsible for collecting this information. A typical example may look like:

```
# hagsns -s grpsvcs
We are: 3.2 pid: 9108 domaindId = 1.2 noNS = 0 inRecovery = 0 CodeLevel = RSCT 0
NS::ENsState(6):kCertain protocolInProgress = NS::ENsProtocol(0):kNoProtocol out
Process started on Jul 13 13:20:14, (1d 22:52:57) ago.  HB connection took (0:0.
Initial NS certainty on Jul 13 13:20:17, (1d 22:52:55) ago, taking (0:0:1).
Our current epoch of certainty started on Jul 13 13:20:17, (1d 22:52:55) ago.
2 UP nodes: 1 3
```

The hagsns command accepts the following flags:

[-h host] -g group_name Gets group status

[-h host] -s subsystem_name Gets subsystem status

[-h host] -p subsystem_pid Gets status by PID

4.2.3.10 hagspbs

Another former component of PSSP high availability infrastructure, the hagspbs command is used to display the status of the broadcast services subcomponent, formally known as the Phoenix Broadcast Services (PBS), an internal part of Group Services. A sample output from the hagspbs command may look similar to:

```
# hagspbs -s grpsvcs
1.1 cssMembership: HWM 8 LWM 7
  pendingAckCount=0 kNotExpectingAcks  pendingRecoverCount=0
        1: HWM=Nil: lastType1=Nil
        2: HWM=6: lastType1=Nil
2.1 ha_em_peers: HWM 11 LWM 10
  pendingAckCount=0 kNotExpectingAcks  pendingRecoverCount=0
        1: HWM=Nil: lastType1=Nil
        2: HWM=8: lastType1=Nil
3.1 CLSTRMGR_1: HWM 55 LWM 53
  pendingAckCount=0 kNotExpectingAcks  pendingRecoverCount=0
        1: HWM=Nil: lastType1=Nil
        2: HWM=45: lastType1=Nil
0.Nil ZtheNameServerXY: HWM 7 LWM 7
  pendingAckCount=0 kNotExpectingAcks  pendingRecoverCount=0
        1.Nil kUp: HWM=6: lastType1=Nil
        2.16 kUp: HWM=6: lastType1=Nil
0.Nil  theGROVELgroup: HWM Nil LWM Nil
  pendingAckCount=0 kNotExpectingAcks  pendingRecoverCount=0
```

The output shows the group identifier and group name followed by the *broadcast sequence numbers* High Water Mark (HWM) and Low Water Mark (LWM), for each group. These refer to the reliable broadcast message stream values described on page 79, with the HWM being the next message number in the sequence and the LWM being the last message to be acknowledged (ACK). Not all messages are acknowledged, since this might place too much strain on the network with larger systems. Instead, approximately every fifth message is a ACKed, meaning that the difference between the values for the HWM and LWM should be anywhere between one and five.

In general, these sequence numbers should match for each group on each of the nodes that are inserted into the groups. However, it may be that due to delays in the messages passed between the nodes (possibly due to network traffic), that some nodes may not have received all the messages. Assuming the traffic subsides, all nodes should eventually catch up.

If one or more groups are running a series of protocols, through failures or additions, a large amount of information may be seen when this command is run.

If a node, domain, or group appears to be hung, the hagspbs command should be run on the group leader to ascertain which nodes we are waiting for.

The hagspbs command accepts the following flags:

[-h host] -g group_name	Gets group status
[-h host] -s subsystem_name	Gets subsystem status

[-h host] -p subsystem_pid Gets status by PID

4.2.3.11 hagsvote

The hagsvote command is, again, converted for use in RSCT from the original PSSP high availability infrastructure. It is used to display information about the groups when they are in the middle of sending and receiving voting protocols. If the group is stable and no current changes are taking place, then very little information is displayed, similar to the following output:

```
root@sp21n13 >hagsvote -ls grpsvcs
  Number of: groups: 6
Group slot # [0] Group name[HostMembership] voting data:
No protocol is currently executing in the group.
--------------------------------------------------

Group slot # [1] Group name[ha_em_peers] voting data:
No protocol is currently executing in the group.
--------------------------------------------------

Group slot # [2] Group name[cssRawMembership] voting data:
No protocol is currently executing in the group.
--------------------------------------------------

Group slot # [3] Group name[cssMembership] voting data:
No protocol is currently executing in the group.
--------------------------------------------------

Group slot # [4] Group name[enMembership] voting data:
No protocol is currently executing in the group.
--------------------------------------------------

Group slot # [5] Group name[CLSTRMGR_1] voting data:
No protocol is currently executing in the group.
--------------------------------------------------
```

Since this provides us with no information to work with, we will have to stimulate some activity by inserting or removing a node from the group. This was achieved by using the sample_test program provided with RSCT. Refer to 5.7, “Using the sample_test Utility” on page 118, for further details on how to start sample_test.

The output we receive from the hagsvote command differs depending upon which type of node we run the command on: either the group leader or a non-group leader node. On a group leader node we can display a summary of all the collected voting responses as well as which nodes have or have not voted. On a non-group leader node, we can only display information concerning the local providers, and whether they have or have not voted.

First let us examine some sample output from the group leader during the course of some protocol activity:

```
# hagsvote -a theSourceGroup -s grpsvcs
Number of: groups: 9
Group name[theSourceGroup] voting data:
GL in phase [1] of an n-phase protocol of type[Join].
Local voting data:
Local provider count [1] Number not yet voted [0](vote submitted).
 Given vote:[Approve vote]Default vote:[No vote value]
Global voting data:
Number of nodes in group [3] Number not yet voted [1]
Given vote:[Approve vote]Default vote:[No vote value]
```

We are looking specifically at our test group called *theSourceGroup*. The output is fairly self-explanatory, with the current protocol type of join, meaning that a new member wishes to join the group. We can see the term *n-phase protocol* in this output, which needs to be explained. In basic terms, the protocols are messages that are the synchronizing mechanism behind the distributed group. There are two types of protocols:

1. A *1-phase protocol* is simply a single broadcast that commits a change to the group data.
2. An *n-phase protocol* is a series of barrier synchronization steps that each provider must reach, before they continue to the next.

Therefore, in our example, we are still in the first phase of the join process, awaiting one of the providers to vote on the join request. We have voted locally, and so has one of the other nodes. In this scenario, we have not yet submitted a vote from the process requesting to join the group.

Now let us examine the output from a non-group leader node during the same period of activity:

```
Wed Jul 22 17:41:59 EDT 1998
  Number of: groups: 6
Group name[theSourceGroup] voting data:
Not GL in phase [1] of an n-phase protocol of type[Join].
Local voting data:
Local provider count [1] Number not yet voted [1](vote not submitted).
 Given vote:[No vote value]Default vote:[No vote value]
```

The output is similar to that seen on the group leader, however, we only see the local voting data, not the global voting data that may be required.

In this output we can see that we have not yet submitted our vote. The field entitled *default vote* will come into play if no vote is submitted in the time period allocated to respond for this group.

The hagsvote command accepts the following flags:

[-h host] [-l] [-a argument] -g group_name	Gets group status
[-h host] [-l] [-a argument] -s subsystem_name	Gets subsystem status
[-h host] [-l] [-a argument] -p subsystem_pid	Gets status by PD

4.2.3.12 hagscounts

The hagscounts command was used internally by the developers as an internal debugging tool. The information provided is used to count the number of internal memory blocks allocated and deallocated by Group Services. To the user or administrator this information is not much use. The tool has been included in case it is required for data collection when reporting problems to IBM.

4.2.3.13 hagsp

The hagsp script is the script to start the Group Services Daemon as an SRC subsystem. It is a PERL script, which is called when Group Services is started and is used to determine whether the daemon should be running in the PSSP domain or in the HACMP domain. The script is not intended to be executed manually.

4.2.3.14 hagsreap

The hagsreap script is another internal PERL script. It is called from hagsp and is used to maintain the log files and directories that Group Services makes use of. It attempts to keep approximately the last 5MB of data, removing the oldest logs and core files if this value is breached.

All of the commands described in the previous section are tools that can be used in conjunction with the daemon's logs, located in /var/ha/log, to obtain detailed information on how they interact with each other and to perform some form of problem determination.

Chapter 5. Problem Determination

This chapter presents some common problems that can occur in an HACMP/ES running cluster. The objective is to put in to practice the knowledge acquired in the previous chapters. For each problem we show the usage of some commands, explain the data displayed by these commands, and look at the data written in the log files. Whenever possible, we also give some suggestions on how to avoid these problems in the future.

This chapter also presents some hints and tips on exploiting the internal mechanisms of HACMP/ES and RSCT.

First, we give a brief description of the environment used to recreate the problems described later on in the chapter. We have two cluster nodes, called *sp21n13* and *sp21n15*, connected by the SP Ethernet (the network name is spether), the SP Switch (the network names are basecss for the SP Switch base address and aliascss for the SP Switch IP alias addresses), and an additional Ethernet network (the network name is ethernet1). Figure 17 on page 89 shows a summary of all the adapters configured in the cluster.

```
Adapter    Type     Network    Type   Attribute Node      IP Address     Name
------------------------------------------------------------------------------
sw13_boot  boot     aliascss   hps    private   sp21n13   140.40.4.33    css0
sw13_svc   service  aliascss   hps    private   sp21n13   140.40.4.13    css0
sp21sw13   service  basecss    hps    private   sp21n13   192.168.14.13  css0
n13_boot   boot     ethernet1  ether  public    sp21n13   128.100.10.30  en1
n13_svc    service  ethernet1  ether  public    sp21n13   128.100.10.3   en1
n13_stdby  standby  ethernet1  ether  public    sp21n13   128.200.30.3   en2
sp21n13    service  spether    ether  private   sp21n13   192.168.4.13   en0
sw15_boot  boot     aliascss   hps    private   sp21n15   140.40.4.55    css0
sw15_svc   service  aliascss   hps    private   sp21n15   140.40.4.15    css0
sp21sw15   service  basecss    hps    private   sp21n15   192.168.14.15  css0
n15_boot   boot     ethernet1  ether  public    sp21n15   128.100.10.10  en1
n15_svc    service  ethernet1  ether  public    sp21n15   128.100.10.1   en1
n15_stdby  standby  ethernet1  ether  public    sp21n15   128.200.20.2   en2
sp21n15    service  spether    ether  private   sp21n15   192.168.4.15   en0
```

Figure 17. Adapter Configuration

For more details of the environment, refer to Appendix A, “Our Environment” on page 189.

5.1 Adapter Failure

In this section we simulate the failure of an Ethernet adapter and then check the most useful HACMP/ES log files and commands to troubleshoot the problem.

After starting HACMP/ES on both cluster nodes, the `lssrc -ls topsvcs` command on node sp21n13 shows the information shown in Figure 18 on page 90.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          12426   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.3    128.100.10.3
 ethernet1_0    [ 0]              0x85b750cc      0x85b750e6
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    2  S 128.200.30.3    128.200.30.3
 ethernet1_1    [ 1]              0x85b74fbc      0x85b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.13    192.168.4.15
 spether_0      [ 2]              0x45b74fbd      0x45b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.13   192.168.14.15
 basecss_0      [ 3]              0x45b750c0      0x45b750c9
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.13     140.40.4.15
 aliascss_0     [ 4]              0x45b750d7      0x45b75113
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 13192) hagsd(  7356)
   Configuration Instance = 12
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 18. lssrc -ls topsvcs Output on Node sp21n13 before Failure

Look at the lines for the network called ethernet1_1. From a Topology Services point of view, this network represents the ring where the standby adapters (n13_stdby and n15_stdby) exchange heartbeats. The relevant information here is the data under the columns `Defd` (Defined), `Mbrs` (Members), and `St` (State). The number `2` under `Defd` and `Mbrs` means that there are two adapters in this ring and both are exchanging heartbeats correctly. The letter `S` under `St` means the ring is stable.

For more detail, about the `lssrc` command, refer to “Topology Services Daemon” on page 54.

The same command, `lssrc -ls topsvcs`, but this time on node sp21n15, shows the information shown in Figure 19 on page 91.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          14206   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.1    128.100.10.3
 ethernet1_0    [ 0]              0x85b74f17      0x85b750e6
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    2  S 128.200.20.2    128.200.30.3
 ethernet1_1    [ 1]              0x85b74ec6      0x85b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.15    192.168.4.15
 spether_0      [ 2]              0x45b74ec7      0x45b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.15   192.168.14.15
 basecss_0      [ 3]              0x45b74ec8      0x45b750c9
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.15     140.40.4.15
 aliascss_0     [ 4]              0x45b74f1c      0x45b75113
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 12812) hagsd( 13246)
   Configuration Instance = 12
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 19. lssrc -ls topsvcs Output on Node sp21n15 before Failure

Just as on node sp21n13, for network ethernet1_1 we see the number `2` under the columns `Defd` and `Mbrs`, and the letter `S` under the column `St`.

At this point we simulate the failure of adapter n13_stdby, which is the standby adapter of node sp21n13, by pulling its Ethernet cable. HACMP/ES recognizes this failure immediately, and records the information in the /tmp/hacmp.out log file shown in Figure 20 on page 92.

```
# tail -f /tmp/hacmp.out
Jul 23 11:11:15 EVENT START: fail_standby sp21n13 128.200.30.3

+ set -u
+ + dspmsg scripts.cat 335 Adapter 128.200.30.3 is no longer available for use a
s a standby,\n due to either a standby adapter failure or IP address takeover.\n
 128.200.30.3
MSG=Adapter 128.200.30.3 is no longer available for use as a standby,
 due to either a standby adapter failure or IP address takeover.
+ /bin/echo Adapter 128.200.30.3 is no longer available for use as a standby, du
e to either a standby adapter failure or IP address takeover.
+ 1> /dev/console
+ exit 0
Jul 23 11:11:15 EVENT COMPLETED: fail_standby sp21n13 128.200.30.3
#
```

Figure 20. /tmp/hacmp.out

The /var/adm/cluster.log file contains the information, shown in Figure 21 on page 92.

```
# tail -f /var/adm/cluster.log
Jul 23 11:11:15 sp21n13 HACMP for AIX: EVENT START: fail_standby sp21n13 128.200
.30.3
Jul 23 11:11:15 sp21n13 HACMP for AIX: EVENT COMPLETED: fail_standby sp21n13 128
.200.30.3
#
```

Figure 21. /var/adm/cluster.log

We now execute the `lssrc -ls topsvcs` command again on node sp21n13 to see how the output has changed from the previous time. The output is shown in Figure 22 on page 93.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          12426   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.3    128.100.10.3
 ethernet1_0    [ 0]              0x85b750cc      0x85b750e6
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    0  D
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.13    192.168.4.15
 spether_0      [ 2]              0x45b74fbd      0x45b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.13   192.168.14.15
 basecss_0      [ 3]              0x45b750c0      0x45b750c9
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.13     140.40.4.15
 aliascss_0     [ 4]              0x45b750d7      0x45b75113
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 13192) hagsd(  7356)
   Configuration Instance = 12
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 22. lssrc -ls topsvcs Output on Node sp21n13 after Failure

Looking at the lines for the ethernet1_1 network, we see the number `0` under the column `Mbrs`, which means node sp21n13 does not detect any working adapters. This makes sense, because by disconnecting the Ethernet cable from adapter n13_stdby, node sp21n13 cannot send or receive any heartbeats. As a consequence, the letter `D` (Disabled) is shown under the column `St`.

Executing the `lssrc -ls topsvcs` command on node sp21n15, we see the data shown in Figure 23 on page 94.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          14206   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.1    128.100.10.3
 ethernet1_0    [ 0]              0x85b74f17      0x85b750e6
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    1  S 128.200.20.2    128.200.20.2
 ethernet1_1    [ 1]              0x85b74ec6      0x85b75287
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.15    192.168.4.15
 spether_0      [ 2]              0x45b74ec7      0x45b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.15   192.168.14.15
 basecss_0      [ 3]              0x45b74ec8      0x45b750c9
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.15     140.40.4.15
 aliascss_0     [ 4]              0x45b74f1c      0x45b75113
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 12812) hagsd( 13246)
   Configuration Instance = 12
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 23. lssrc -ls topsvcs Output on Node sp21n15 after Failure

This time the output of the `lssrc -ls topsvcs` command on node sp21n15 is different from that on node sp21n13. In fact, under the `Mbrs` column we see the number `1` and under `St` we see the letter `S` (Stable). This means that node sp21n15 has recognized the failure of adapter n13_stdby, but from its point of view the ring is still working properly because the adapter n15_stdby is up.

On node sp21n13, HACMP/ES also records the failure of the standby adapter n13_stby in the log file /var/ha/log/topsvcs. Figure 24 on page 94 is an extract of the information written in this log file.

```
07/23  11:11:02 hatsd[1]: Node (128.200.20.2:0x85b74ec6) is dead.
07/23  11:11:02 hatsd[1]: Notifying leader (128.200.30.3:0x85b74fbc) of death.
07/23  11:11:02 hatsd[1]: Received a DEATH IN FAMILY message from (128.200.30.3:
0x85b74fbc) in group (128.200.30.3:0x85b74fbf).
```

Figure 24. /var/ha/log/topsvcs

The most important line of this figure is the line `Received a DEATH IN FAMILY message from 128.200.30.3`, which means Topology Services has detected the

failure of IP address 128.200.30.3, which is indeed our standby adapter n13_stdby.

Another useful command is `hagsgr -s grpsvcs`. Figure 25 on page 95 shows the output of this command before the failure.

```
# hagsgr -s grpsvcs
  Number of: groups: 6
Group slot # [0] Group name[HostMembership] group state[Not Inserted |]
Providers[]
Local subscribers[]

Group slot # [1] Group name[ha_em_peers] group state[Inserted |Idle |]
Providers[[1/2][1/1]]
Local subscribers[]

Group slot # [2] Group name[cssRawMembership] group state[Idle |]
Providers[[1/2][3/2][1/1][3/1]]
Local subscribers[[11/2][12/2]]

Group slot # [3] Group name[cssMembership] group state[Inserted |Idle |]
Providers[[0/2][0/1]]
Local subscribers[]

Group slot # [4] Group name[enMembership] group state[Idle |]
Providers[[2/2][0/2][1/2][0/1][1/1][2/1]]
Local subscribers[[12/2]]

Group slot # [5] Group name[CLSTRMGR_1] group state[Inserted |Idle |]
Providers[[1/2][0/1]]
Local subscribers[]
#
```

Figure 25. hagsgr-s grpsvcs Output before Failure

Look at the lines about the enMembership group in order to compare them with Figure 26 on page 96, which contains the output of the `hagsgr -s grpsvcs` command after the failure.

```
# hagsgr -s grpsvcs
  Number of: groups: 6
Group slot # [0] Group name[HostMembership] group state[Not Inserted |]
Providers[]
Local subscribers[]

Group slot # [1] Group name[ha_em_peers] group state[Inserted |Idle |]
Providers[[1/2][1/1]]
Local subscribers[]

Group slot # [2] Group name[cssRawMembership] group state[Idle |]
Providers[[1/2][3/2][1/1][3/1]]
Local subscribers[[11/2][12/2]]

Group slot # [3] Group name[cssMembership] group state[Inserted |Idle |]
Providers[[0/2][0/1]]
Local subscribers[]

Group slot # [4] Group name[enMembership] group state[Idle |]
Providers[[2/2][0/2][1/2][0/1][1/1]]
Local subscribers[[12/2]]

Group slot # [5] Group name[CLSTRMGR_1] group state[Inserted |Idle |]
Providers[[1/2][0/1]]
#
```

Figure 26. hagsgr-s grpsvcs Output after Failure

We have to concentrate on the lines regarding the enMembership group. The `hagsgr` command just reinforces the concept that the n13_stdby Ethernet adapter has failed. We can see that we have three providers for the cluster node with node number `2` (sp21n15), but only two providers for the cluster node with node number 1 (sp21n13).

For more details about `the hagsgr` command, refer to 4.2.3.7, “hagsgr” on page 73.

To find out the relationship between cluster node names (sp21n13 and sp21n15) and cluster node numbers (1 and 2, respectively) we can use the `clhandle -a` command shown in Figure 27 on page 96.

```
# clhandle -a
1 sp21n13
2 sp21n15
#
```

Figure 27. clhandle -a Output

For more details about the `clhandle` command, refer to 4.2.2.3, “clhandle” on page 61.

5.2 Node Failure

In this section we see what happens when a cluster node fails. In our HACMP/ES configuration we have two cluster nodes, node sp21n13 and node sp21n15. First we look at the configuration after having started HACMP/ES on both nodes. It will be interesting to compare the data shown by these commands with the data collected after the failure of node sp21n13 has occurred.

In this HACMP/ES configuration, node sp21n13 acquires two Service IP addresses, n13_svc on the en1 network interface and sw13_svc on the css0 network interface. The `netstat -i` command shows the IP addresses configured on node sp21n13 shown in Figure 28 on page 97.

```
# netstat -i
Name  Mtu   Network     Address              Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                           58512     0    58793     0     0
lo0   16896 127         loopback             58512     0    58793     0     0
lo0   16896 ::1                              58512     0    58793     0     0
en0   1500  link#2      2.60.8c.2e.7c.1e    251342     0   201625     0     0
en0   1500  192.168.4   sp21n13             251342     0   201625     0     0
en1   1500  link#3      2.60.8c.2f.ac.d1    132102     0   107741     0     0
en1   1500  128.100     n13_svc             132102     0   107741     0     0
en2   1500  link#4      2.60.8c.2e.60.63    128749     0   104048     0     0
en2   1500  128.200     n13_stdby           128749     0   104048     0     0
css0  65520 link#5                          353644     0   331344     0     0
css0  65520 192.168.14  sp21sw13            353644     0   331344     0     0
css0  65520 140.40.4    sw13_svc            353644     0   331344     0     0
#
```

Figure 28. netstat -i Output On Node sp21n13 before Failure

Node sp21n15 also acquires two Service IP addresses, n15_svc on the en1 network interface and sw15_svc on the css0 network interface. The same command, `netstat -i`, shows the information for node sp21n15 shown in Figure 29 on page 98.

```
# netstat -i
Name  Mtu   Network     Address             Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                          6790     0     6843     0     0
lo0   16896 127         loopback            6790     0     6843     0     0
lo0   16896 ::1                             6790     0     6843     0     0
en0   1500  link#2      2.60.8c.2e.78.c9    11246     0     8571     0     0
en0   1500  192.168.4   sp21n15             11246     0     8571     0     0
en1   1500  link#3      2.60.8c.2c.e2.32     2179     0     1602     0     0
en1   1500  128.100     n15_svc              2179     0     1602     0     0
en2   1500  link#4      2.60.8c.2e.87.b1     2145     0      613     0     0
en2   1500  128.200     n15_stdby            2145     0      613     0     0
css0  65520 link#5                          7288     0    14007     0     0
css0  65520 192.168.14  sp21sw15             7288     0    14007     0     0
css0  65520 140.40.4    sw15_svc             7288     0    14007     0     0
#
```

Figure 29. netstat -i Output On Node sp21n15 before Failure

The `lssrc -ls topsvcs` command shows the information for node sp21n13 shown in Figure 30 on page 98.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          12226   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.3    128.100.10.3
 ethernet1_0    [ 0]              0x85b8ac62      0x85b8ac7b
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    2  S 128.200.30.3    128.200.30.3
 ethernet1_1    [ 1]              0x85b8ac26      0x85b8ac29
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.13    192.168.4.15
 spether_0      [ 2]              0x45b8ac27      0x45b8ac29
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.13   192.168.14.15
 basecss_0      [ 3]              0x45b8ac28      0x45b8ac29
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.13     140.40.4.15
 aliascss_0     [ 4]              0x45b8ac66      0x45b8aca2
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 14324) hagsd( 16516)
   Configuration Instance = 13
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 30. lssrc -ls topsvcs Output On Node sp21n13 before Failure

Figure 31 on page 99 contains the output of the `lssrc -ls topsvcs` command on node sp21n15. We have to look closely at the data under the `Mbrs` column, because this is what is going to change after the failure.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          7814    active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.1    128.100.10.3
 ethernet1_0    [ 0]              0x85b8abd5      0x85b8ac7b
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    2  S 128.200.20.2    128.200.30.3
 ethernet1_1    [ 1]              0x85b8ab83      0x85b8ac29
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.15    192.168.4.15
 spether_0      [ 2]              0x45b8ab84      0x45b8ac29
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.15   192.168.14.15
 basecss_0      [ 3]              0x45b8ab85      0x45b8ac29
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.15     140.40.4.15
 aliascss_0     [ 4]              0x45b8abdb      0x45b8aca2
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd(  3628) hagsd(  6840)
   Configuration Instance = 13
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 31. lssrc -ls topsvcs Output On Node sp21n15 before Failure

Figure 32 on page 99 shows the output of the `lssrc -ls grpsvcs` command on node sp21n15.

For more details about the `lssrc` command, refer to “Group Services Daemon” on page 56.

```
# lssrc -ls grpsvcs
Subsystem         Group            PID     Status
 grpsvcs          grpsvcs          6840    active
 3 locally-connected clients.  Their PIDs:
 3628 2206 5178
 HA Group Services domain information:
 Domain established by node 2.
 Number of groups known locally: 3
                    Number of   Number of local
 Group name         providers   providers/subscribers
 cssMembership            2           1           0
 ha_em_peers              2           1           0
 CLSTRMGR_1               2           1           0
#
```

Figure 32. lssrc -ls grpsvcs Output on Node sp21n15 before Failure

The next figure shows an extract of the /var/ha/log/topsvcs log file captured on node sp21n15.

```
07/24  11:45:47 hatsd[2]: My New Group ID = (192.168.4.15:0x45b8ac29) and is Sta
ble.
        My Leader is              (192.168.4.15:0x45b8ab84).
        My Crown Prince is        (192.168.4.13:0x45b8ac27).
        My upstream neighbor is   (192.168.4.13:0x45b8ac27).
        My downstream neighbor is (192.168.4.13:0x45b8ac27).
```

Figure 33. /var/ha/log/topsvcs on Node sp21n15 before Failure

This data tells us how the Topology Services daemon, topsvcs, has built the ring to exchange heartbeats on the SP Ethernet network. We can see that we have two adapters on the ring with IP addresses 192.168.4.15, which corresponds to node sp21n15, and 192.168.4.13, which corresponds to node sp21n13. The adapter with the highest IP address is the Group Leader and the adapter with the next highest IP address in the ring is the Crown Prince.

For more details about the Group Leader, refer to 4.2.3.2, “Group Leader” on page 66.

Now we simulate a failure of node sp21n13. The surviving node, sp21n15, acquires the cluster resources during takeover. At this point we again execute the commands we have already seen so far and point out the differences. All commands that we see from now on were executed on node sp21n15, since it is the only node in the cluster still functioning at this time.

We start by looking at the output of the `netstat -i` command, shown in the Figure 34 on page 101.

```
# netstat -i
Name  Mtu   Network     Address             Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                          2485     0     2514     0     0
lo0   16896 127         loopback            2485     0     2514     0     0
lo0   16896 ::1                             2485     0     2514     0     0
en0   1500  link#2      2.60.8c.2e.78.c9    5871     0     4707     0     0
en0   1500  192.168.4   sp21n15             5871     0     4707     0     0
en1   1500  link#3      2.60.8c.2c.e2.32    1255     0      768     0     0
en1   1500  128.100     n15_svc             1255     0      768     0     0
en2   1500  link#4      2.60.8c.2e.87.b1    1120     0      490     0     0
en2   1500  128.100     n13_svc             1120     0      490     0     0
css0  65520 link#5                          4330     0     6812     0     0
css0  65520 192.168.14  sp21sw15            4330     0     6812     0     0
css0  65520 140.40.4    sw15_svc            4330     0     6812     0     0
css0  65520 140.40.4    sw13_svc            4330     0     6812     0     0
#
```

Figure 34. netstat -i Output on Node sp21n15 after Failure

We can see that in the HACMP/ES takeover, node sp21n15 has acquired the n13_svc Service IP address on the en2 network interface and the sw13_svc Service IP alias address on the css0 network interface. This can also be seen by looking at the /var/adm/cluster.log log file on node sp21n15. During the takeover, the entries shown in Figure 35 on page 101 were written in this log file.

```
# tail -f /var/adm/cluster.log
Jul 24 11:52:42 sp21n15 HACMP for AIX: EVENT START: node_down sp21n13
Jul 24 11:52:42 sp21n15 HACMP for AIX: EVENT START: node_down_remote rg13
Jul 24 11:52:43 sp21n15 HACMP for AIX: EVENT START: acquire_takeover_addr n13_sv
c sw13_svc
Jul 24 11:52:48 sp21n15 HACMP for AIX: EVENT COMPLETED: acquire_takeover_addr n1
3_svc sw13_svc
Jul 24 11:52:48 sp21n15 HACMP for AIX: EVENT START: get_disk_vg_fs /fs13
Jul 24 11:53:02 sp21n15 HACMP for AIX: EVENT COMPLETED: get_disk_vg_fs /fs13
Jul 24 11:53:02 sp21n15 HACMP for AIX: EVENT COMPLETED: node_down_remote rg13
Jul 24 11:53:03 sp21n15 HACMP for AIX: EVENT COMPLETED: node_down sp21n13
Jul 24 11:53:03 sp21n15 HACMP for AIX: EVENT START: node_down_complete sp21n13
Jul 24 11:53:04 sp21n15 HACMP for AIX: EVENT START: node_down_remote_complete rg
13
Jul 24 11:53:04 sp21n15 HACMP for AIX: EVENT COMPLETED: node_down_remote_complet
e rg13
Jul 24 11:53:04 sp21n15 HACMP for AIX: EVENT COMPLETED: node_down_complete sp21n
13
#
```

Figure 35. /var/adm/cluster.log on Node sp21n15 after Failure

The cluster.log file records all the event shell scripts executed by HACMP/ES. The shell script acquire_takeover_addr is the event responsible for configuring on node sp21n15. the two IP addresses n13_svc and sw13_svc that were originally owned by node sp21n13.

Figure 36 on page 102 shows the output of the `lssrc -ls topsvcs` command.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          7814    active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    1  S 128.100.10.1    128.100.10.1
 ethernet1_0    [ 0]              0x85b8abd5      0x85b8adee
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    1  S 192.168.4.15    192.168.4.15
 spether_0      [ 2]              0x45b8ab84      0x45b8adee
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    1  S 192.168.14.15   192.168.14.15
 basecss_0      [ 3]              0x45b8ab85      0x45b8adee
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    1  S 140.40.4.15     140.40.4.15
 aliascss_0     [ 4]              0x45b8abdb      0x45b8ae06
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd(  3628) hagsd(  6840)
   Configuration Instance = 13
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
#
```

Figure 36. lssrc -ls topsvcs Output on Node sp21n15 after Failure

The interesting information in this figure is that we now see the number 1 under the column `Mbrs`, while before the failure we had 2. Since node sp21n13 has failed, all its adapters are not exchanging heartbeats anymore and for this reason Topology Services has taken them out of the ring.

Figure 37 on page 102 shows the output of the `lssrc -ls grpsvcs` command.

```
# lssrc -ls grpsvcs
Subsystem         Group            PID     Status
 grpsvcs          grpsvcs          6840    active
 3 locally-connected clients.  Their PIDs:
 3628 2206 5178
 HA Group Services domain information:
 Domain established by node 2.
 Number of groups known locally: 3
                    Number of   Number of local
 Group name         providers   providers/subscribers
 cssMembership            1           1           0
 ha_em_peers              1           1           0
 CLSTRMGR_1               1           1           0
#
```

Figure 37. lssrc -ls grpsvcs Output on Node sp21n15 after Failure

The difference from before the failure is the data under the column `Number of providers`. Before the failure we had 2, now we only have 1, which means that

in each of the three groups (cssMembership, ha_em_peers, and CLSTRMGR_1) there is only one member.

Figure 38 on page 103 and Figure 39 on page 103 show an extract of the information written in the /var/ha/log/topsvcs log file on node sp21n15.

```
07/24  11:53:18 hatsd[2]: Node (192.168.4.13:0x45b8ac27) is dead.
07/24  11:53:18 hatsd[2]: Notifying leader (192.168.4.15:0x45b8ab84) of death.
07/24  11:53:18 hatsd[3]: Node (192.168.14.13:0x45b8ac28) is dead.
07/24  11:53:18 hatsd[3]: Notifying leader (192.168.14.15:0x45b8ab85) of death.
07/24  11:53:18 hatsd[3]: Node (192.168.14.13:0x45b8ac28) is dead.
07/24  11:53:18 hatsd[3]: Notifying leader (192.168.14.15:0x45b8ab85) of death.
07/24  11:53:18 hatsd[3]: Received a DEATH IN FAMILY message from (192.168.14.15
:0x45b8ab85) in group (192.168.14.15:0x45b8ac29).
```

Figure 38. /var/ha/log/topsvcs on Node sp21n15 after Failure (1)

From this information we see that Topology Services has detected the failure of the sp21n13 adapter, which has IP address 192.168.4.13 and is the SP Ethernet of node sp21n13.

```
07/24  11:53:18 hatsd[3]: My New Group ID = (192.168.14.15:0x45b8adee) and is St
able.
        My Leader is              (192.168.14.15:0x45b8ab85).
        My Crown Prince is        (192.168.14.15:0x45b8ab85).
        My upstream neighbor is   (192.168.14.15:0x45b8ab85).
        My downstream neighbor is (192.168.14.15:0x45b8ab85).
```

Figure 39. /var/ha/log/topsvcs on Node sp21n15 after Failure (2)

This data shows how Topology Services builds a new ring to exchange heartbeats after having excluded the sp21n13 adapter. The only adapter in the ring is sp21n15 with IP address 192.168.14.15, and it is now both the Group Leader and the Crown Prince.

5.3 Deadman Switch

The term Deadman Switch (DMS) describes the AIX kernel extension that causes a cluster node to crash with 888 flashing on the LEDs. The Deadman Switch is simply a timer that expires if not reset periodically by the cluster manager daemon, hence causing the cluster node to panic. There are many reasons why a cluster node may crash because of the Deadman Switch. The

most common ones are a large amount of I/O traffic, a high priority process not releasing control of the CPU, and heavy paging space activity.

When a cluster node crashes with 888 flashing, it is very easy to determine if the system dump was caused by the Deadman Switch or not. In case it was really the Deadman Switch, HACMP/ES writes the entry in the System Error Log shown in Figure 40 on page 104.

```
---------------------------------------------------------------------------
LABEL:          KERNEL_PANIC
IDENTIFIER:     225E3B63

Date/Time:       Wed Jul 22 17:15:19
Sequence Number: 394
Machine Id:      000089847000
Node Id:         000089847000
Class:           S
Type:            TEMP
Resource Name:   PANIC

Description
SOFTWARE PROGRAM ABNORMALLY TERMINATED

        Recommended Actions
        PERFORM PROBLEM DETERMINATION PROCEDURES

Detail Data
ASSERT STRING

PANIC STRING
HACMP for AIX dms timeout - halting hung node
---------------------------------------------------------------------------
```

Figure 40. System Error Log Entry Written by HACMP/ES

The last line in Figure 40 on page 104, `HACMP for AIX dms timeout - halting hung node`, demonstrates that this is a DMS crash. Another way to determine if the panic was caused by the Deadman Switch is to use the `crash` command to look at the system dump, as shown in Figure 41 on page 105.

```
# crash /dev/lv00 /unix
> stat
        sysname: AIX
        nodename: sp21n13
        release: 3
        version: 4
        machine: 000089847000
        time of crash: Wed Jul 22 17:15:19 1998
        age of system: 2 hr., 5 min.
        xmalloc debug: disabled
        dump code: 700
        csa: 0x320eb0
        exception struct:
                0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
        panic: HACMP for AIX dms timeout - ha

>
```

Figure 41. Crash Output

The last line at the bottom of Figure 41 on page 105, `panic: HACMP for AIX dms timeout - ha`, is the confirmation that this system dump was caused by the Deadman Switch.

5.3.1 How to Prevent the Deadman Switch Problem

In this section we give some suggestions on how to prevent the Deadman Switch problem. These hints must not be considered the complete solution, but rather a way to reduce the risk of incurring the DMS problem.

The first suggestion is to increase the frequency the *syncd* daemon is invoked from the default value of sixty to ten seconds. This daemon is responsible for flushing to disk all the data residing in the system buffers. The syncd daemon is started at IPL from the /sbin/rc.boot configuration file, as shown in Figure 42 on page 105.

```
if [ -f /etc/rc.B1 ]; then
        /etc/rc.B1 label
fi

echo "Starting the sync daemon" | alog -t boot
nohup /usr/sbin/syncd 60 > /dev/null 2>&1 &

# deleting error notification objects that should not survive reboot
odmdelete -o errnotify -q "en_persistenceflg=0" >/dev/null 2>&1
```

Figure 42. Extract from /sbin/rc.boot

When the syncd daemon is invoked after sixty seconds, it can occasionally cause large amounts of I/O traffic that can then trigger the Deadman Switch. Reducing the syncd interval down to ten seconds makes the system less vulnerable to big I/O transfers and hence reduces the risk of DMS. After modifying the /sbin/rc.boot file, it is necessary to rebuild the system boot images by issuing the `bosboot` command, as shown in Figure 43 on page 106.

```
# bosboot -s -d /dev/hdisk0
#
# shutdown -Fr
```

Figure 43. bosboot and shutdown Commands

Since the system boot image is only read at IPL, it is also necessary to reboot the cluster node after the `bosboot` command.

For a detailed description of the syncd daemon, see *AIX Version 3.2 and 4 Performance Tuning Guide*, SC23-2365

The second suggestion for preventing DMS regards I/O pacing, which is configured via the SMIT menu shown in Figure 44 on page 107. The command `smit chgsys` is the fastpath to reach this menu directly:

```
                Change / Show Characteristics of Operating System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Maximum number of PROCESSES allowed per user       [256]                +#
  Maximum number of pages in block I/O BUFFER CACHE  [20]                 +#
  Maximum Kbytes of real memory allowed for MBUFS    [0]                  +#
  Automatically REBOOT system after a crash           false               +
  Continuously maintain DISK I/O history              true                +
  HIGH water mark for pending write I/Os per file    [33]                 +#
  LOW water mark for pending write I/Os per file     [24]                 +#
  Enable memory SCRUBBING                             false               +
  Amount of usable physical memory in Kbytes          262144
  State of system keylock at boot time                normal
  Enable full CORE dump                               false               +
  Use pre-430 style CORE dump                         false               +


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 44. SMIT Menu for I/O Pacing

The two parameters that affect I/O pacing are `HIGH water mark for pending write I/Os per file` and `LOW water mark for pending write I/Os per file`. Our suggestion is to set these two parameters to the values of 33 and 24, respectively.

For a detailed description of I/O pacing, see *AIX Version 3.2 and 4 Performance Tuning Guide*, SC23-2365

The third suggestion for preventing DMS is to change the parameters that affect the exchange of heartbeats between the cluster nodes. To determine which parameters HACMP/ES is using, we execute the `lssrc -ls topsvcs` command, as shown in Figure 45 on page 108.

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          14206   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet1_0    [ 0]    2    2  S 128.100.10.1    128.100.10.3
 ethernet1_0    [ 0]              0x85b74f17      0x85b750e6
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 ethernet1_1    [ 1]    2    2  S 128.200.20.2    128.200.30.3
 ethernet1_1    [ 1]              0x85b74ec6      0x85b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 spether_0      [ 2]    2    2  S 192.168.4.15    192.168.4.15
 spether_0      [ 2]              0x45b74ec7      0x45b74fbf
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 basecss_0      [ 3]    2    2  S 192.168.14.15   192.168.14.15
 basecss_0      [ 3]              0x45b74ec8      0x45b750c9
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 aliascss_0     [ 4]    2    2  S 140.40.4.15     140.40.4.15
 aliascss_0     [ 4]              0x45b74f1c      0x45b75113
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 12812) hagsd( 13246)
   Configuration Instance = 12
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
```

Figure 45. lssrc -ls topsvcs Output

Looking at Figure 45 on page 108, the two parameters that affect the heartbeat exchange are `HB Interval` and `Sensitivity`. `HB Interval` specifies the time interval, in seconds, between heartbeats. The default value is one second. `Sensitivity` specifies the number of successive heartbeats that can be missed. The default value is four heartbeats. HACMP/ES V4.3 uses the following formula to calculate the time needed to detect a failure: (HB Interval) * (Sensitivity) * 2 seconds. For example, one cluster node is considered dead after no heartbeat exchange occurs for eight consecutive seconds, assuming the default values.

Our suggestion is to leave unchanged the `HB Interval` parameter and increase the `Sensitivity` value. So, in order for an HACMP/ES cluster node to be less vulnerable to the Deadman Switch problem, one suggestion is to change the Sensitivity parameter to 6. In this way, the cluster node would be considered failed if it does not exchange heartbeats for 12 seconds.

5.4 User-Defined Event Problem

In this section we show what happens when a user-defined event is configured incorrectly. First, we explain how the problem occurs, then we

discuss the troubleshooting. For a detailed explanation of user-defined events, see 6.2, “User-Defined Events” on page 138.

We have two nodes in the cluster, sp21n13 and sp21n15. We start HACMP/ES on node sp21n15, and while the command is still running, all of a sudden the screen hangs. The node seems isolated from the network as well, since it does not respond to a `ping` or `telnet` command. By looking at Perspectives, host responds has become red. At this point, the only thing left is to power off and then on again.

When the IPL has finished, we start investigating the problem. The /tmp/hacmp.out file is empty. In the System Error Log no relevant entries have been recorded. So we decide to look at the /tmp/clstrmgr.debug log file, which contains the few lines shown in Figure 46 on page 109.

```
Thu Jul 30 14:57:30 HACMP/PE Cluster Manager Version 4.0
Thu Jul 30 14:57:30 ReadTopsvcs: called.
Thu Jul 30 14:57:30 GetObjects: Called
Thu Jul 30 14:57:30 ReadTopsvcs: hbInterval = 1, fibrillateCount = 4, fixedPriLe
vel = 38, runFixedPri = 1 instanceNum = 12
Thu Jul 30 14:57:30 ReadTopsvcs: Calculated fixed priority is 39
Thu Jul 30 14:57:30 GetObjects: Called
Thu Jul 30 14:57:30 ReadClVersion: Setting CLversion = 1 from HACMPcluster
Thu Jul 30 14:57:30 HACMP/PE Cluster Manager Started
Thu Jul 30 14:57:30 MakeDeadman: Trying to set up deadman connection (clm_smux)
Thu Jul 30 14:57:30 MakeDeadman: Trying to set up deadman connection (clm_lkm)
Thu Jul 30 14:57:30 RdInit: Called
RdInit: Event Manager event defined
     Name =
     State = 0
     Resource Variable = 0
     Instance Vector = IBM.PSSP.aixos.FS.%totused
     Predicate = NodeNum=3;LV=hd3;VG=rootvg
     Rearm Predicate = X>90

Thu Jul 30 14:57:30 RdInit: Bad format in rules file at line 218
Thu Jul 30 14:57:30 RdInit error
Thu Jul 30 14:57:30 clstrmgr: Disable DMS timer
Thu Jul 30 14:57:30 clstrmgr on node 0 is exiting with code 3
```

Figure 46. Extract from the /tmp/clstrmgr.debug Log File

The line towards the end, `RdInit: Bad format in rules file at line 218`, seems to indicate a problem regarding the rules.hacmprd file, which is the file where all the cluster events are defined.

The next step is to look at the /var/adm/cluster.log file. Figure 47 on page 110 shows its contents.

```
Jul 30 14:57:30 sp21n15 clstrmgr[5950]: Thu Jul 30 14:57:30 HACMP/PE Cluster Man
ager Started
Jul 30 14:57:30 sp21n15 clstrmgr[5950]: Thu Jul 30 14:57:30 RdInit: Bad format i
n rules file at line 218
Jul 30 14:57:30 sp21n15 clstrmgr[5950]: Thu Jul 30 14:57:30 RdInit error
Jul 30 14:57:30 sp21n15 clstrmgr[5950]: Thu Jul 30 14:57:30 clstrmgr on node 0 i
s exiting with code 3
Jul 30 14:57:31 sp21n15 HACMP for AIX: clexit.rc : Unexpected termination of cls
trmgr.
Jul 30 14:57:31 sp21n15 HACMP for AIX: clexit.rc : Halting system immediately!!!
```

Figure 47. Extract from the /var/adm/cluster.log Log File

This information is basically the same as the one found in the /tmp/clstrmgr.debug file before, and again it points to the rules.hacmprd file. The date and timestamps are also identical.

We now look at the rules.hacmprd file under the /usr/sbin/cluster/events directory to make sure it has the correct format. At the end of this file we find a user-defined event, which is shown in Figure 48 on page 110.

```
# start of User Event tmpfull

UE_TMPFULL
0
/usr/local/tmpfull.rp
2
0
IBM.PSSP.aixos.FS.%totused
NodeNum=2;LV=hd3;VG=rootvg
X>90
X<80
# end of User Event tmpfull
```

Figure 48. Extract from the /usr/sbin/cluster/events/rules.hacmprd File

The problem is clear now. The `UE_TMPFULL` user-defined event has not been configured correctly. Each event must be *exactly* nine lines. Looking closely at Figure 48 on page 110, we can see there are ten lines instead. We can ignore the lines starting with the pound sign since they are simply comments. The problem is the extra, blank line just above the event name, `UE_TMPFULL`. After removing this blank line, HACMP/ES now starts correctly.

For more details of the rules.hacmprd file, see 6.2.3, “The rules.hacmprd File” on page 141.

5.5 The emsvcs Daemon Problem

In this section we examine a problem regarding the emsvcs daemon. The problem manifests itself in the following way. After starting HACMP/ES, it is always good practice to check that everything has been initialized correctly. So we look at the /tmp/hacmp.out log file and do not see any errors, we monitor the cluster resources and see that all of them have been acquired as expected, we execute the `lssrc` command to make sure all the daemons are running - and here we find an anomaly: the emsvcs daemon is not in the active state, as can be seen in Figure 49 on page 111.

```
# lssrc -a | grep svcs
 topsvcs          topsvcs          13180   active
 grpsvcs          grpsvcs          13004   active
 grpglsm          grpsvcs          13692   active
 emaixos          emsvcs           14476   active
 emsvcs           emsvcs                   inoperative
#
```

Figure 49. lssrc Output

So we start to investigate this problem. First we check the System Error Log using the `errpt -a` command and we find three entries related to our problem, having ERROR LABELs of SRC, CORE_DUMP and HA002_ER. They are shown in detail in Figure 50 on page 112 through Figure 53 on page 115.

```
# errpt -a
---------------------------------------------------------------------------
LABEL:          SRC
IDENTIFIER:     E18E984F

Date/Time:       Thu Jul 23 09:59:18
Sequence Number: 478
Machine Id:      000106837000
Node Id:         sp21n15
Class:           S
Type:            PERM
Resource Name:   SRC

Description
SOFTWARE PROGRAM ERROR

Probable Causes
APPLICATION PROGRAM

Failure Causes
SOFTWARE PROGRAM

        Recommended Actions
        PERFORM PROBLEM RECOVERY PROCEDURES

Detail Data
SYMPTOM CODE
           0
SOFTWARE ERROR CODE
       -9020
ERROR CODE
         134
DETECTING MODULE
'srchevn.c'@line:'281'
FAILING MODULE
emsvcs
---------------------------------------------------------------------------
```

Figure 50. Extract from System Error Log Labeled SRC

```
---------------------------------------------------------------------------
LABEL:          CORE_DUMP
IDENTIFIER:     C60BB505

Date/Time:       Thu Jul 23 09:59:17
Sequence Number: 477
Machine Id:      000106837000
Node Id:         sp21n15
Class:           S
Type:            PERM
Resource Name:   SYSPROC

Description
SOFTWARE PROGRAM ABNORMALLY TERMINATED

Probable Causes
SOFTWARE PROGRAM

User Causes
USER GENERATED SIGNAL

        Recommended Actions
        CORRECT THEN RETRY

Failure Causes
SOFTWARE PROGRAM

        Recommended Actions
        RERUN THE APPLICATION PROGRAM
        IF PROBLEM PERSISTS THEN DO THE FOLLOWING
        CONTACT APPROPRIATE SERVICE REPRESENTATIVE

Detail Data
SIGNAL NUMBER
           6
USER'S PROCESS ID:
       15876
FILE SYSTEM SERIAL NUMBER
           7
INODE NUMBER
        2083
PROGRAM NAME
haemd
```

Figure 51. Extract from System Error Log Labeled CORE_DUMP (1/2)

```
ADDITIONAL INFORMATION
raise 4C
??
abort B8
emd_exit 38
parse_joi 10C
approve_c DC
ha_gs_dis F4C
dispatch_ 14
ctrl_loop 4CC
main 63C
??

Symptom Data
REPORTABLE
1
INTERNAL ERROR
0
SYMPTOM CODE
PCSS/SPI2 FLDS/haemd SIG/6 FLDS/emd_exit VALU/38
-------------------------------------------------------------------------------
```

Figure 52. Extract from System Error Log Labeled CORE_DUMP (2/2)

```
---------------------------------------------------------------------------
LABEL:          HA002_ER
IDENTIFIER:     12081DC6

Date/Time:       Thu Jul 23 09:59:17
Sequence Number: 476
Machine Id:      000106837000
Node Id:         sp21n15
Class:           S
Type:            PERM
Resource Name:   haemd

Description
SOFTWARE PROGRAM ERROR

Probable Causes
SUBSYSTEM

Failure Causes
SUBSYSTEM

        Recommended Actions
        REPORT DETAILED DATA
        CONTACT APPROPRIATE SERVICE REPRESENTATIVE

Detail Data
DETECTING MODULE
LPP=PSSP,Fn=emd_gsi.c,SID=1.4.1.28,L#=1389,
DIAGNOSTIC EXPLANATION
haemd(cluster_a): 2521-007 Internal error (65551).
---------------------------------------------------------------------------
```

Figure 53. Extract from System Error Log Labeled HA002_ER

The most important information we find by looking at these entries is that the emsvcs daemon has created a core file. The IBM RS/6000 Cluster Technology daemons (topsvcs, grpsvcs, grpglsm and emsvcs) create core files under the /var/ha/run directory. Here each daemon has its own directory. The directory names are called *daemonname.clustername,* where *daemonname* is simply the name of the daemon and *clustername* is the HACMP/ES cluster name assigned during configuration. However, there is one exception to this rule, and it regards the emsvcs daemon, whose directory is called *haem.clustername* instead of *emsvcs.clustername*. In the next figure we look at the contents of this directory:

```
# cd /var/ha/run/haem.cluster_a
# ls -l
total 10547
drwxr-xr-x   2 root     haemrm       512 Jul 10 10:29 Rcache_local
drwxr-xr-x   2 root     haemrm       512 Jul 10 10:29 Rcache_remote
drwxr-xr-x   2 root     system       512 Jul 10 10:29 aixos
-rw-rw-rw-   1 root     haemrm   2695975 Jul 23 09:59 core
#
```

Figure 54. Files in the /var/ha/run/haem.cluster_a Directory

As expected, we find a core file here, and its date and timestamp match the date and timestamp of the three errors written in the System Error Log that we have just examined.

In this case, we cannot suggest any action to solve the problem, so the next step would be to call IBM support.

5.6 Cluster Manager

This section provides information to trace the state of the Cluster Manger. Basically, the Cluster Manager monitors events. If an event happens, the Cluster Manager looks in the rules.hacmprd file. According to the name of the event, the Cluster Manager decides which recovery program should be executed. In the recovery program, there are definitions how to run recovery scripts. The Cluster Manager changes its state according to its action against events. When the Cluster Manager changes its state, it leaves a message in the /tmp/clstrmgr.debug internal log file. This file includes Finite State Machine-event (FSM-event) and Cluster Manager state change information. If you examine this file, it is possible to learn what recovery script the Cluster Manager executed.

The following example uses the case of initial cluster formation described in 3.3.2, “Initial Cluster Formation” on page 36. Figure 55 on page 117 is the Cluster Manager state change example while the first node comes up. The steps of this example are from step 1 “INIT state (Node A)” on page 36 to step 12 “STABLE state (Node A)” on page 39.

There is a lot of information written to this log file. To concentrate on information about the state change, we used the `grep` command, which picks up the FMS-event related lines.

```
# grep FSM-event /tmp/clstrmgr.debug
Mon Jul 27 17:48:17 FSMrun: running state = ST_INIT, FSM-event = EV_JOIN_MSG
Mon Jul 27 17:48:17 FSMrun: next state = ST_JOINING, FSM-event = FSM_NONE
Mon Jul 27 17:48:42 FSMrun: running state = ST_JOINING, FSM-event = EV_JOIN_DONE
Mon Jul 27 17:48:42 FSMrun: next state = ST_STABLE, FSM-event = FSM_NONE
Mon Jul 27 17:48:42 FSMrun: running state = ST_STABLE, FSM-event = EV_NEW_EVENT
Mon Jul 27 17:48:42 FSMrun: next state = ST_UNSTABLE, FSM-event = FSM_NONE
Mon Jul 27 17:48:44 FSMrun: running state = ST_UNSTABLE, FSM-event = EV_TIMEOUT
Mon Jul 27 17:48:44 FSMrun: next state = ST_VOTING, FSM-event = FSM_NONE
Mon Jul 27 17:48:45 FSMrun: running state = ST_VOTING, FSM-event = EV_VOTE_MSG
Mon Jul 27 17:48:45 FSMrun: next state = ST_VOTING, FSM-event = FSM_NONE
Mon Jul 27 17:48:45 FSMrun: running state = ST_VOTING, FSM-event = EV_VOTE_DONE
Mon Jul 27 17:48:45 FSMqueue: queueing FSM-event = EV_END_STEP
Mon Jul 27 17:48:45 FSMrun: next state = ST_RP_RUNNING, FSM-event = EV_END_STEP
Mon Jul 27 17:48:45 FSMrun: next state = ST_RP_RUNNING, FSM-event = EV_END_STEP
Mon Jul 27 17:48:45 FSMrun: running state = ST_RP_RUNNING, FSM-event = EV_END_STEP
Mon Jul 27 17:48:45 FSMrun: next state = ST_BARRIER, FSM-event = FSM_NONE
Mon Jul 27 17:48:45 FSMrun: running state = ST_BARRIER, FSM-event = EV_BARRIER_MSG
Mon Jul 27 17:48:45 FSMrun: next state = ST_BARRIER, FSM-event = FSM_NONE
Mon Jul 27 17:48:45 FSMrun: running state = ST_BARRIER, FSM-event = EV_BARRIER_DONE
Mon Jul 27 17:48:45 FSMrun: next state = ST_RP_RUNNING, FSM-event = FSM_NONE
Mon Jul 27 17:49:53 FSMrun: running state = ST_RP_RUNNING, FSM-event = EV_BARRIER_MSG
Mon Jul 27 17:49:53 FSMrun: next state = ST_BARRIER, FSM-event = FSM_NONE
Mon Jul 27 17:49:53 FSMrun: running state = ST_BARRIER, FSM-event = EV_BARRIER_DONE
Mon Jul 27 17:49:53 FSMrun: next state = ST_RP_RUNNING, FSM-event = FSM_NONE
Mon Jul 27 17:49:53 FSMrun: next state = ST_RP_RUNNING, FSM-event = FSM_NONE
Mon Jul 27 17:49:54 FSMrun: running state = ST_RP_RUNNING, FSM-event = EV_CBARRIER_MSG
Mon Jul 27 17:49:54 FSMrun: next state = ST_CBARRIER, FSM-event = FSM_NONE
Mon Jul 27 17:49:54 FSMrun: running state = ST_CBARRIER, FSM-event = EV_CBARRIER_DONE
Mon Jul 27 17:49:54 FSMrun: next state = ST_STABLE, FSM-event = FSM_NONE
Mon Jul 27 17:50:00 FSMrun: running state = ST_STABLE, FSM-event = EV_TIMEOUT
Mon Jul 27 17:50:00 FSMqueue: queueing FSM-event = EV_NO_EVENTS
Mon Jul 27 17:50:00 FSMrun: next state = ST_VOTING, FSM-event = EV_NO_EVENTS
Mon Jul 27 17:50:00 FSMrun: running state = ST_VOTING, FSM-event = EV_NO_EVENTS
Mon Jul 27 17:50:00 FSMrun: next state = ST_STABLE, FSM-event = FSM_NONE
```

Figure 55. A Part of the /tmp/clstrmgr.debug File

The line starting with `FSMrun: running state = ...` and the line starting with `FSMrun next state = ...` make a pair. These lines tell you that the FSM-event `JOIN_MSG` happened at `Mon Jul 27 17:48:17`, then the Cluster Manager did the specified action for the state and FSM-event pair, then the state of the Cluster Manager was changed from `INIT` to `JOINING`.

You can see the relationship of Cluster Manager states and FSM-events, in Figure 2 on page 36 and Table 2 on page 36.

This example is an initial cluster formation, so it uses the node_up.rp recovery program. Important messages are line 15, 17, 21, 23, 26, and 28. Line 15, running state = `ST_RP_RUNNING`, shows that the Cluster Manager executed the node_up recovery script for other nodes. Line 17, running state = `ST_BARRIER`, shows that the Cluster Manager waited at the barrier statement. Lline 21, running state = `ST_RP_RUNNING`, shows that the Cluster Manager executed the node_up recovery script for the event node. Line 23, running state = `ST_BARRIER`, shows that the Cluster Manager waited at the barrier statement. Line 26, running state = `ST_RP_RUNNING`, shows that the Cluster

Manager executed the node_up_complete recovery script on all nodes. Line 28, running state = ST_CBARRIER, shows that the Cluster Manager was exiting the node_up.rp recovery program.

The states and their sequence depends on what kind of recovery program the Cluster Manager executed. If something is wrong with the recovery program or recovery script, you will find messages in the log file that might help you determine problems with the HACMP Cluster Manager.

5.7 Using the sample_test Utility

This section may help provide some starting tips for using the sample_test utility provided with the RSCT filesets. This is an unsupported utility provided purely on an as-is basis, which means that if any problems are found either by using the utility or within the utility itself, it is up to the user to take responsibility and perform corrective actions. We found the utility useful in gaining a greater understanding of how groups are formed and some of the protocols that run within them.

The utility and the source code it was compiled from is located in the directory /usr/sbin/rsct/samples/hags, and contains the following files:

```
# ls
Makefile.sample          Sample_Subscribe          sample_deactive_ksh
Sample_Frame.C           Sample_Subscribe.C        sample_schg
Sample_Frame.h           Sample_Subscribe.h        sample_schg.c
Sample_FrameTable.C      Sample_Subscription.C     sample_test
Sample_FrameTable.h      Sample_Subscription.h     sample_test.c
Sample_Node.C            sample_callbacks.c        sample_utility.c
Sample_Node.h            sample_callbacks.h        sample_utility.h
Sample_ProviderTable.C   sample_deactive_c_prog
Sample_ProviderTable.h   sample_deactive_c_prog.c
```

The sample_test file is a precompiled executable (RS/6000) or object module for use on AIX systems. Before it can be used, however, we need to supply the following variables as described in *IBM RS/6000 Cluster Technology for AIX: Group Services Programming Guide and Reference*, SA22-7355.

HA_DOMAIN_NAME This is the domain name for the current HACMP/ES cluster we are using. The name is actually the same as the name of the cluster. If you are unsure of the exact name, run lssrc -ls against the event management operating system resource monitor

(emaixos), which displays the domain name in the output.

HA_GS_SUBSYS This is the subsystem name that we wish to interface with, either the PSSP group services daemon (hats) or the HACMP/ES group services daemon (grpsvcs).

Here is an example of the variables:

```
# lssrc -ls emaixos|grep -i domain
Domain Type:         HACMP
Domain Name:         cluster_a
# export HA_DOMAIN_NAME=cluster_a
# export HA_GS_SUBSYS=grpsvcs
```

Now that we have set our variables, we can execute the `sample_test` script. We found that using the script in interactive mode was easier to follow and provided breakpoints for us for examining the effects of our actions.

The next screen snapshot shows us starting the executable in interactive mode and displaying the help screen, which provides a brief description of the flags and their meanings:

```
# sample_test -i
**********++++++++++==========**********++++++++++==========

My program index is 0 and the program name is: <sample_test0>

**********++++++++++==========**********++++++++++==========
Please enter a command('h' for help): h

Enter: 'i'(ha_gs_init) '1'(1 phase join) 'j'(n phase join)
 'x'(good target join, if 'j' was done) 'm'(bad target join)
 'a'(mismatch attributes, if done by both even and odd indexed sample_tests)
 't'(state value change) 'p'(provider broadcast message)
 's'(subscribe to a group) 'u'(unsubscribe from a group)
 'l'(leave a group) 'e'(expel one or more providers from a group)
 'g'(even index: source 'theSourceGroup', odd index: no source-group)
 '3'(good target join, if 'j' and 'x' were done)
 'b'(build a group by defining its attributes)
 'n'(modify a group's attributes)
 'y'(say goodbye ("exit immediately from") a group)
 'z'(toggle HA_GS_DEACTIVATE_ON_FAILURE)
 'd'(ha_gs_dispatch) 'h' (print these instructions) 'q'(quit)

To automatically get these instructions at each prompt, use
 'verbose' mode via '-v' on the command line.

To interactively specify initialization parameters, specify '-i' on
 the command line.

Write instructions returned rc:[HA_GS_OK]
Please enter a command('h' for help):
```

It may be useful to print out this information, instead of having to refer back each time you want to run a command.

Before we can do anything else, we need to register with the Group Services Application Programming Interface (GSAPI) by using the ha_gs_init() subroutine. We do this by entering the `i` (ha_gs_init) command. We will be prompted to respond to each parameter that needs to be specified with ha_gs_init() if we chose interactive mode, otherwise we will use some default values. An example can be seen here:

```
Please enter a command('h' for help): i
Please enter a command('h' for help): Please specify the client's socket control
 attribute.
Enter: 0 (SIGNAL) anything else (NO SIGNAL): 1
Using HA_GS_SOCKET_NO_SIGNAL!  Good idea.
The default deactivate script is[./sample_deactive_c_prog].
  Keep it (0 [no] 1 [yes])? 1
Keeping default deactivatate script [./sample_deactive_c_prog].
What kind of responsiveness?
 1 [Ping]  2 [Counter]  3 [None]: 1
Using HA_GS_PING_RESPONSIVENESS.  Good idea.
Specify the interval (in seconds) between responsiveness checks: 3000
Specify time limit (in seconds) for response to responsiveness check: 5
Responsiveness parameters specified:  Type[HA_GS_PING_RESPONSIVENESS]
Interval[3000 seconds] Response time limit[5 seconds]

Do you want to respond manually or automatically to all responsiveness
 notifications?  If auto, then we will always return OK.  If manual, you
 will be able to specify OK or NOT_OK at notification time.
 Specify 'A' [auto] or 'M' [manual] (default is auto):
Automatic responsiveness responses!
ha_gs_init returned rc:[HA_GS_OK]
Please enter a command('h' for help):
```

We successfully registered with Group Services. If you receive a return from ha_gs_init() that says "unsuccessful", for example `HA_GS_CONNECT_FAILED` instead of `HA_GS_OK`, then it is likely that something is wrong with the variables that were exported before we executed `sample_test`.

After registering with Group Services, we can perform simple actions, such as joining or subscribing to a group. Here we can see what happens when we join a new group by entering the `j` (n phase join) command:

```
Please enter a command('h' for help): j
Please enter a command('h' for help): **********++++++++++==========**********++
++++++++=========
Attempting JOIN for group[theSourceGroup]:
 My instance #[100]  local name[SourceJoin]
Group attributes:
 version[1] size[40] client version[1]
 Batching[HA_GS_BATCH_BOTH] Join/Fail phases[N_PHASE] Reflection phases[N_PHASE]
 Default vote[HA_GS_VOTE_REJECT] Merge[HA_GS_DONTCARE_MERGE]
 Time limits: join/fail[60] reflection[30]
 Group name[theSourceGroup] Source group name[]
**********++++++++++==========**********++++++++++==========
ha_gs_join so far returned rc:[HA_GS_OK]
Please enter a command('h' for help):
[TOD(Jul 23 14:08:41)]N Phase Callback Zero called
********************  Thu Jul 23 14:08:41 1998  ********************
Type[HA_GS_JOIN] Token[0] TimeLimit[60]
Summary_Code[]
NumPhases[N_PHASE] ThisPhase[1] Proposer[100/2]
whatsChanged[HA_GS_PROPOSED_MEMBERSHIP HA_GS_ONGOING_MEMBERSHIP]
CurrentProviders[No providers in list!]
ChangingProviders[count[1] Members[100/2 ]]
CurrentState[Length[4] Value[0]]
**************************************************************************
Protocol is HA_GS_JOIN
SUMMARY_CODE=
**********++++++++++==========**********++++++++++==========
**********++++++++++==========**********++++++++++==========
Enter a vote value: C [Continue]  A [Approve]  R [Reject]
 or, to force 'nested' proposals: T [State change] P [PBM]
```

Here we are being prompted on what vote we should give. Since we are the first process requesting to join, our vote is the deciding vote.

At this point we can use some commands in another window to interrogate Group Services and list the protocols being used, for example by using the `hagsvote` command.

We can see that we have only 60 seconds to vote before our default vote of `HA_GS_VOTE_REJECT` is submitted. So let us approve the join request and join the group *theSourceGroup* by entering the `A` (Approve) command.

```
Enter a vote value: C [Continue]  A [Approve]  R [Reject]
 or, to force 'nested' proposals: T [State change] P [PBM]
A
You have chosen well, grasshopper. This protocol will be approved here and now!
**********++++++++++==========**********++++++++++==========
Phase[1]vote value[HA_GS_VOTE_APPROVE]default vote value[HA_GS_NULL_VOTE]
**********++++++++++==========**********++++++++++==========
ha_gs_dispatch returned rc:[HA_GS_OK]
Please enter a command('h' for help):
[TOD(Jul 23 14:09:06)]Approved Callback Zero called
********************  Thu Jul 23 14:09:06 1998  ********************
Type[HA_GS_JOIN] Token[0]
Summary_Code[HA_GS_EXPLICIT_APPROVE]
NumPhases[N_PHASE] ThisPhase[1] Proposer[100/2]
WhatsChanged[HA_GS_UPDATED_PROVIDER_MESSAGE HA_GS_UPDATED_MEMBERSHIP]
CurrentProviders[count[1] Members[100/2 ]]
ChangingProviders[count[1] Members[100/2 ]]
>>>> Grabbed my provider id [100/2]
CurrentState[Length[4] Value[0]]
ProviderMessage[Length[67] Value[0<sample_test0>Things do seem to be looking up
now, do they not?]]
**********************************************************************
ha_gs_dispatch returned rc:[HA_GS_OK]
Please enter a command('h' for help):
```

Once we approve the join we can see some of the protocols being exchanged by Group Services and the new group. We do not appear to have had any errors, so let us examine the groups managed by grpsvcs with the hagsgr command.

```
# hagsgr -s grpsvcs
  Number of: groups: 7
Group slot # [0] Group name[HostMembership] group state[Not Inserted |]
Providers[]
Local subscribers[]

Group slot # [1] Group name[ha_em_peers] group state[Inserted |Idle |]
Providers[[1/2][1/1]]
Local subscribers[]

Group slot # [2] Group name[cssRawMembership] group state[Idle |]
Providers[[1/2][3/2][1/1][3/1]]
Local subscribers[[11/2][12/2]]

Group slot # [3] Group name[cssMembership] group state[Inserted |Idle |]
Providers[[0/2][0/1]]
Local subscribers[]

Group slot # [4] Group name[enMembership] group state[Idle |]
Providers[[2/2][0/2][1/2][0/1][1/1][2/1]]
Local subscribers[[12/2]]

Group slot # [5] Group name[CLSTRMGR_1] group state[Inserted |Idle |]
Providers[[1/2][0/1]]
Local subscribers[]

Group slot # [6] Group name[theSourceGroup] group state[Inserted |Idle |]
Providers[[100/2]]
Local subscribers[]
```

We now see our new group, `theSourceGroup (Group slot # [6])`, inserted as the last group. The provider in this case is the `sample_test` executable. If we run `sample_test` on multiple nodes in the cluster and join the group, *theSourceGroup*, we will see multiple providers for this group. Now that we have joined a group, we are able to utilize some of the other functions available, such as subscribing to existing groups, expelling members from the group, or forcing state changes on the group.

For more information on GSAPI and the subroutines available, such as ha_gs_init(), refer to *IBM RS/6000 Cluster Technology for AIX: Group Services Programming Guide and Reference*, SA22-7355.

5.8 Release Notes and Readme Files

Many IBM Products have their own Release Notes, and HACMP/ES is no exception. The Release Notes are simply a document that includes answers to frequently asked questions, solutions to very common problems, as well as

updates and corrections to the Product manuals. The Readme files are very similar to the Release Notes as far as the information they contain.

We highly recommend that you always read the Release Notes and Readme files every time you install HACMP/ES or encounter a problem with HACMP/ES.

5.8.1 HACMP/ES Release Notes and Readme Files

The HACMP/ES Release Notes are located in the file release_notes under the /usr/lpp/cluster/doc directory. Figure 56 on page 126 shows the table of contents of the HACMP/ES V4.2.2 Release Notes.

Note: We show the HACMP/ES V4.2.2 Release Notes because at the time of this writing the HACMP/ES V4.3 Release Notes were not available yet.

```
The release notes consist of the following topics.

=============================================================
New Functionality
 - CLVerify Enhancements
 - Kerberos, Phase II
 - (HACMP, HACMP/ES) Event Emulation
 - (HACMP, HACMP/ES) DARE Resource Migration
 - Fast Recovery
 - HAView Cluster Monitoring Utility
 - Custom Verification Methods
o Product Constraints
 - Required BOS Level
 - Microcode Levels
 - Support Not Available
 - (HANFS) Cluster Configuration Parameters
o Installation and Migration Notes
 - Restoring OLDer HACMPevent Classes
 - Installing only Messages Results in Failed Install
 - Upgrading a Version 4.2 Cluster Using the 4.2.2 Maintenance Level Updates
 - Upgrade to 4.2.2 Doesn't Require cl_convert Utility
 - Saving Cluster Configuration or Customized Event Scripts
 - (HACMP) Concurrent LVM Issues during Cluster Migration to AIX 4.3
 - (HACMP) Client-Only Migration
 - (HANFS) Migrating to HANFS 4.2.2
 - (HACMP/ES) clconvert_snapshot Utility
 - (HACMP/ES) User-editable Files Saved During HACMP/ES Installation
o Post-installation Processing
 - Recovering Configuration Information
 - (HACMP, HACMP/ES) Do not install HC daemon with RVSD
o Configuration Notes
 - IPAT Required in Cascading Resource Groups with NFS Mount Point
 - Nodes Must Have Direct Network Connections for Successful Integration
 - Failure of Single Active Adapter does not Generate Events
 - Setting Up NFS Clients
 - Slight Delay on Clients During Takeover
 - Customizing Network Failure Events
 - Check Automount Attribute of File Systems
 - (HACMP) 9333 and SSA Disk Fencing
 - (HACMP, HACMP/ES) Loopback Addresses no Longer Needed
 - (HACMP, HACMP/ES) Synchronization of Event Customization
```

Figure 56. HACMP/ES V4.2.2. Release Notes

The HACMP/ES Readme file is located under the /usr/sbin/cluster directory in the file READMEV.R.M.UPDATE, where V.R.M stands for the Version, Release and Modification level. For example, the HACMP/ES V4.2.2 Readme file is called README4.2.2.UPDATE. The Readme file is often updated when a PTF is installed, especially if the PTF, in addition to fixing some bugs, also adds some new functionality to HACMP/ES, for example, support for new hardware.

5.8.2 IBM RS/6000 Cluster Technology Readme Files

The IBM RS/6000 Cluster Technology (RSCT) software has its own Readme files. They are located under the /usr/sbin/rsct/README directory and are called rsct.basic.README and rsct.clients.README. As for HACMP/ES, we recommend that you read these files.

5.9 Data Necessary for IBM Support to Troubleshoot a Problem

In this section we explain what data the system administrator must provide to the IBM Support Personnel in order to diagnose an HACMP/ES V4.3 problem. The debug data and log files listed in this section are sufficient to solve the majority of the problems that can occur in a running cluster. However, the reader should understand that in case of very complex situations, IBM Support may ask for additional, more detailed, information.

The data to submit to IBM is the following:

- HACMP/ES V4.3 cluster snapshot
- Miscellaneous commands and files
- Description of the customer environment

We now explain how to collect this data.

5.9.1 HACMP/ES V4.3 Cluster Snapshot

A cluster snapshot allows the system administrator to save the HACMP/ES configuration in two ASCII files. The snapshot can be extremely useful in two situations. First, when the cluster configuration is corrupted, it can be quickly restored by just applying a previously saved snapshot. Second, it enables you to have an immediate idea of the critical AIX and HACMP/ES configuration parameters of a cluster.

We recommend that you save a cluster snapshot as soon as the HACMP/ES configuration is completed and has been tested. The following figure shows how to save a cluster snapshot:

```
 Add a Cluster Snapshot

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Cluster Snapshot Name                            [clustera]               /
  Custom Defined Snapshot Methods                  []                      +
* Cluster Snapshot Description                     [hacmp/es cluster]



F1=Help          F2=Refresh       F3=Cancel        F4=List
Esc+5=Reset      F6=Command       F7=Edit          F8=Image
F9=Shell         F10=Exit         Enter=Do
```

The field `Cluster Snapshot Name` is used to specify the name of the snapshot files. In this example, SMIT creates two files called `clustera` under the /usr/sbin/cluster/snapshots directory. One file has the extension .odm and contains all the cluster configuration parameters as they are written in the HACMP/ES GODM classes. The second file has the extension .info and contains the output of AIX and HACMP/ES commands like `lslpp, lsvg, netstat, ifconfig, no, cllsif`, and others.

The field `Custom Defined Snapshot Methods` allows the system administrator to specify the full path name of a command or shell script that is executed by HACMP/ES to collect additional information, for example, regarding the customer application.

The `Cluster Snapshot Description` field is used to write a comment about the snapshot being taken.

5.9.2 Miscellaneous Commands and Files

In addition to the commands executed automatically when a cluster snapshot is taken, we also recommend that you save the following data *on all cluster nodes.*

- /tmp/hacmp.out*

- /var/adm/cluster.log
- /usr/sbin/cluster/history/cluster*
- /tmp/clstrmgr.debug*
- errpt -a
- lscfg -v
- lslpp -l
- /var/ha/log/*

  Saves the log files of the Topology Services, Group Services and Event Management daemons.

- /var/ha/run/*

  Saves the configuration and eventual core files of Topology Services, Group Services and Event Management.

5.9.3 Description of the Customer Environment

The system administrator must also provide a detailed description of the customer environment. For example, he should explain the physical network layout (are there any bridges/routers/hubs in the cluster configuration?), the client systems accessing the cluster (what Operating System do they run? what systems are they?), the customer application (name and version), the external disks (machine type, model, cabling, RAID or mirroring, SCSI or SSA).

And anything else you think can help solve the problem!

Chapter 6. Configuration Examples

This chapter provides configuration examples, including:

- Global Network
- User-defined events
- Kerberos

6.1 Global Network

In this section we take a closer look at the steps involved in creating a global network, using the `claddnetwork` command. For more information on the `claddnetwork` command and the concept of a global network, refer to 4.2.2.5, “claddnetwork” on page 63.

First, let us document our cluster environment and the adapters we need to configure. You can see in Figure 57 on page 132 that we have two SP nodes that are in two entirely different SP systems. However, we wish to create a heartbeating network between them using the SP Ethernet.

Note that the setup we have created is for demonstration purposes only. To have a supported global network setup, all physical network types must be the same, for example, all Ethernet or all token ring. We unfortunately did not have the necessary hardware, so we set up a working, but at the time of writing, non-supported configuration.

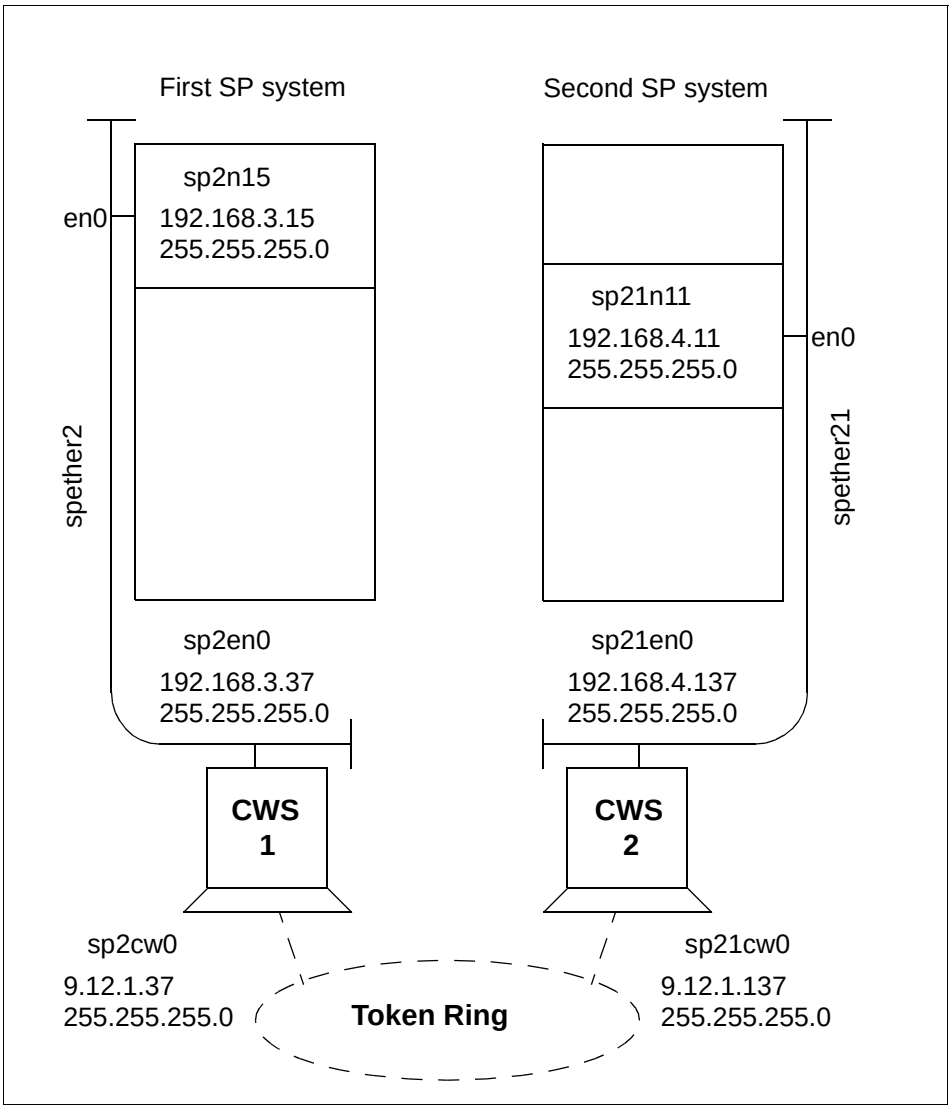


Figure 57. The Physical Network Setup

As with most of the HACMP setup process, some configuration and testing must be performed prior to telling HACMP how to use the setup. In our case, before we can set up a global network, we need to define to each of the nodes the routes they must take to reach each other. This involves the following steps:

1. Set ipforwarding to 1 on both Control Workstations (CWSs), as they are acting as the routers.

```
# no -o ipforwarding
ipforwarding = 1
```

2. Define, on each node, that the route to the other SP system's SP Ethernet is through their own CWS. The simplest way to achieve this is to use SMIT, with the fastpath *mkroute*. Here we can see an example of us setting the default route on sp21n11:

```
Add Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Destination TYPE                                     net                    +
* DESTINATION Address                                 [192.168.3]
  (dotted decimal or symbolic name)
* Default GATEWAY Address                             [192.168.4.137]
  (dotted decimal or symbolic name)
* METRIC (number of hops to destination gateway)      [1]                     #
  Network MASK (hexadecimal or dotted decimal)        [255.255.255.0]


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

This should update the routing table on sp21n11, as shown in the next example:

```
# netstat -rn
Routing tables
Destination      Gateway           Flags   Refs     Use  If   PMTU  Exp  Groups

Route Tree for Protocol Family 2 (Internet):
default          192.168.4.137     UG        0       0  en0     -   -
9.12.1/24        192.168.4.137     UG        0    4413  en0     -   -
80.7/16          80.7.6.11         U         2   73641  en1     -   -
80.9/16          80.9.9.1          U         0       4  en2     -   -
127/8            127.0.0.1         U         9   54112  lo0     -   -
192.168.3/24     192.168.4.137     UG        1   73394  en0     -   -
192.168.4/24     192.168.4.11      U         7   77681  en0     -   -
192.168.14/24    192.168.14.11     U         2  155530 css0     -   -

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0       0  lo0 16896   -
```

3. Define on each CWS that the route to the other SP system's SP Ethernet, is via the token ring network to the other CWS. Again, the simplest way to perform this is through SMIT. We can see this action taking place on CWS 2 in the following example:

```
Add a Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Destination TYPE                                 net                  +
* DESTINATION Address                             [192.168.3]
  (dotted decimal or symbolic name)
* GATEWAY Address                                 [9.12.1.37]
  (dotted decimal or symbolic name)
* METRIC (number of hops to destination gateway)  [1]                   #
  Network MASK (hexadecimal or dotted decimal)    [255.255.255.0]


F1=Help            F2=Refresh         F3=Cancel          F4=List
Esc+5=Reset        F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

Notice that we use the token ring IP address here, *not* the CWS's Ethernet address. This correctly defines that we communicate across the token ring network, which is shown in the routing table on CWS 2 as:

```
# netstat -rn
Routing tables
Destination      Gateway           Flags   Refs     Use  If   PMTU  Exp  Groups

Route tree for Protocol Family 2 (Internet):
default          9.12.1.30         UG       11  8202411  tr0     -   -
9.12.1/24        9.12.1.137        U        17  1298931  tr0     -   -
10.1/16          192.168.4.9       UG        2  1028236  en0     -   -
127/8            127.0.0.1         U         6     3660  lo0     -   -
192.168.3/24     9.12.1.37         UGD       5   125825  tr0     -   -
192.168.4/24     192.168.4.137     U        60  6682456  en0     -   -

Route tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0 16896   -
```

4. Update the /etc/hosts file on each node to include the opposite node for name resolution. Next we can see the /etc/hosts entry on sp21n11:

```
# grep 192.168.3.15 /etc/hosts
192.168.3.15    sp2n15                  # spethernet for other SP
```

5. Test that the communication paths are correct with the `traceroute` command, and that the hostname and IP address resolve to each other. In the following example on sp21n11 we see that name resolution and communication are working correctly for the setup we have just created:

```
# host sp2n15
sp2n15 is 192.168.3.15
# host 192.168.3.15
sp2n15 is 192.168.3.15
# ping -c 3 sp2n15
PING sp2n15: (192.168.3.15): 56 data bytes
64 bytes from 192.168.3.15: icmp_seq=0 ttl=253 time=4 ms
64 bytes from 192.168.3.15: icmp_seq=1 ttl=253 time=4 ms
64 bytes from 192.168.3.15: icmp_seq=2 ttl=253 time=4 ms

----sp2n15 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
# traceroute 192.168.3.15
trying to get source for 192.168.3.15
source should be 192.168.4.11
traceroute to 192.168.3.15 (192.168.3.15) from 192.168.4.11 (192.168.4.11), 30 h
ops max
outgoing MTU = 1500
 1  sp21en0 (192.168.4.137)  8 ms  3 ms  3 ms
 2  sp21en0 (192.168.4.137)  3 ms
fragmentation required, trying new MTU = 1492
 2  9.12.1.37 (9.12.1.37)  6 ms  6 ms  6 ms
 3  sp2n15 (192.168.3.15)  8 ms  8 ms  8 ms
#
```

If the tests produce the output expected, then we have correctly configured our routes and are now in a position to define the global network to HACMP/ES. In our example we see that we pass to our local CWS, then to the remote CWS, and finally onto the destination system. It is important to make sure that we are using the correct route, and not just assume that the IP packets are using the route we set up.

The first task, if we have not already done so, is to define our SP Ethernet to HACMP for each node under the Configure Adapters menu, located under Cluster Topology, in SMIT. Since each node is on a different network both logically and physically, we must assign a different network name for each SP Ethernet.

Once each SP Ethernet has been defined, we can tell HACMP to configure a global network to include both SP Ethernet networks, either through SMIT or directly from the command line using `claddnetwork`. Here is an example of us using the SMIT menu to add a local network to a global network on sp21n11, called globalnet:

```
 Change/Show a Global Network

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Local Network Name                                  spether21
  Global Network Name                                 [globalnet]             +




F1=Help            F2=Refresh         F3=Cancel          F4=List
Esc+5=Reset        F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

We repeat the action and add the other SP Ethernet to the global definition. The next stage, as always when making a change in HACMP, is to perform a `Synchronize Cluster Topology`.

Once completed, we should have a new heartbeating global network. If we look at Topology Services, we see that a new network has been defined:

```
# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          7340    active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 ethernet2_0    [ 0]    3    3 S 80.7.6.11       80.7.6.31
 ethernet2_0    [ 0]              0x85b793d7      0x85b79537
 HB Interval = 1 secs. Sensitivity = 4 missed beats
 globalnet_0    [ 1]    2    2 S 192.168.4.11    192.168.4.11
 globalnet_0    [ 1]              0x45b79390      0x45b7961f
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 16926) hagsd(  5574)
   Configuration Instance = 6
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
```

If we use the new command `cllsgnw`, we can list the association between our local and global networks. In the following example, we see that we now have both our local networks `spether2` and `spether21` defined to our global network `globalnet`:

```
# cllsgnw -a
Name       Global Name
ethernet2
spether2   globalnet
spether21  globalnet
```

For more details of the `cllsgnw` command, refer to 4.2.2.8, “cllsgnw” on page 65.

We can also see that the HACMPnetwork object class of the GODM has been updated with the new global name:

```
# odmget HACMPnetwork

HACMPnetwork:
        name = "ethernet2"
        attr = "public"
        network_id = 0
        globalname = ""

HACMPnetwork:
        name = "spether2"
        attr = "private"
        network_id = 1
        globalname = "globalnet"

HACMPnetwork:
        name = "spether21"
        attr = "private"
        network_id = 2
        globalname = "globalnet"
```

For more details of the HACMPnetwork GODM class, refer to “HACMPnetwork” on page 21.

6.2 User-Defined Events

This section gives a brief overview of the basics and a guidance on how to configure a user-defined event. The guidance is a step-by-step description based on the example we chose.

Before starting with any modification, you should first back up your system. We highly recommend making a copy of the */usr/sbin/cluster/events/rules.hacmprd* file.

Note: The rules.hacmprd file must be the same on all nodes in a cluster.

We used the application Netscape FastTrack Server (Web Server) for our example. The HACMP-related requirements/information for this application are:

- To have at least one file system on the shared disk
- One common IP address (HACMP service address)
- How to start and stop this application
- What happens if one of the processes gets killed

6.2.1 The Script Files

It was necessary to write some script files to imbed this application in HACMP/ES V4.3. Basically we had to modify the HACMP Application Start and Stop script files and the /usr/sbin/cluster/events/rules.hacmprd file. Additionally we create three different kinds of script files.

6.2.1.1 Basic HACMP Script Files

These are the HACMP Application Start and Stop script files. In our case it was necessary to modify these scripts. To get the event detection working properly we had to add a *test* file, which was required to determine whether HACMP/ES stopped the application or whether the application died unexpectedly. The scripts are listed in B.1.1“Start Script” on page 202 and B.1.2“Stop Script” on page 203.

6.2.1.2 User-Event Script Files

These are the necessary script files to define our user event. We had to create a Recovery Program file and three script files.

- The Recovery Program file webserv.rp (listed in B.2.1“The webserv.rp File” on page 203)
- The script files we used are:
  - webserv_local (listed in B.2.3“The webserv_local Script” on page 204)
  - webserv_remote (listed in B.2.2“The webserv_remote Script” on page 204)
  - webserv_complete (listed in B.2.4“The webserv_complete Script” on page 206)

6.2.2 The Recovery Program

The Recovery Program file, or <xxx>.rp file, is where you define the programs that will be executed by the HACMP Cluster Manager in case of an event. This file is read by the Cluster Manager, who uses this file to start the HACMP

event scripts (recovery commands) for a given event in the right order. You can specify on what kind of nodes the programs will run.

Note: For user-defined events, <xxx> can be any name.

The first keyword of each stanza in the recovery program file can be:

This indicates a comment line in the file.

barrier This indicates a synchronization point for all the specified commands before it. When a note hits the barrier statement in the recovery program, the Cluster Manager initiates the barrier protocol on this node. When all nodes have met the barrier in the recovery program and voted to approve the protocol, Group Services notifies all nodes that this protocol has completed. The next command of the recovery program is then executed.

event The cluster node where the event occurred executes this script (recovery command).

other All the nodes where the event did not occur execute this script (recovery command).

all All cluster nodes run this script (recovery command).

Note: The format of the lines starting with *event*, *other*, and *all* is:

relationship command_to_run expected_status NULL

relationship This is where one of the three-keywords mentioned above (event, other, all) is specified.

“command_to_run (recovery command)”

This is a double-quotes-delimited string. For user-defined events it is the full path name of the command to be executed. For the HACMP event scripts, it is the event name only.

expected_status This is the expected return code of the specified command. The actual return code is compared with the specified one. If the values do not match, the Cluster Manager detects an event failure. If you specify an X, the comparison is turned off.

NULL Null is a reserved field for a future release. The word NULL must appear at the end of each of these lines.

Note: There has to be at least one blank between the values.

- If you specify multiple recovery commands between two barrier commands, or before the first one, the recovery commands are executed in parallel, both on the node itself and on all the nodes.

6.2.3 The rules.hacmprd File

The rules.hacmprd file is located in the /usr/sbin/cluster/events directory. This file is used by the Cluster Manager to map the events to their respective Recovery Program. It exists only once on a cluster node and must have the same content on all nodes in the HACMP/ES cluster. The format of an entry is as follows:

1. Name
2. State (qualifier)
3. Recovery program path
4. Recovery type (reserved for future use)
5. Recovery level (reserved for future use)
6. Resource variable name (used for Event Manager events)
7. Instance vector (used for Event Manager events)
8. Predicate (used for Event Manager events)
9. Rearm predicate (used for Event Manager events)

Note:

- Each entry must have exactly these nine lines, in this sequence.
- A line can be empty.

6.2.4 Configuration Steps

This section describes the configuration steps for the application we used. The steps may vary for another application, but the basic structure is the same. We describe here only the additional steps you have to do to define a user event. We assume that the application is already installed and configured to the HACMP/ES V4.3 cluster.

1. Select a possible event condition.

   For more detailed information about possible event conditions, see *HACMP Enhanced Scalability User-Defined Events*, SG24-5327 or *RS/6000 SP Monitoring: Keep It Alive*, SG24-4873, or *IBM RS/6000 Cluster Technology for AIX: Event Management Programming Guide and Reference*, SA22-7354.

2. Test your event conditions.

   We recommend that you test the functionality of your event conditions using SP Event Perspective or the SP Problem Management functions by issuing the `pmandef` command.

3. Create the event script files.

   In our case we had to create three script files:

   - webserv_local

     This script file handles the restart of the application. If the restart does not work, a takeover is initiated by this script. For more details, see the script file itself in B.2.3“The webserv_local Script” on page 204.

   - webserv_remote

     This script file is used to log that this event happened on another node in the HACMP/ES cluster. For the script file itself, see B.2.2“The webserv_remote Script” on page 204.

   - webserv_complete

     This script file is used to log on all cluster nodes that the handling of the event finished. For the script file itself, see B.2.4“The webserv_complete Script” on page 206.

4. Test the event script files.

   We recommend that you test the script files outside of HACMP first.

5. Change the existing Start and Stop script files.

   In our case it was necessary to modify the existing application Start and Stop script files. To get the event detection working properly we had to add a test file, which was required to determine whether HACMP/ES stopped the application or whether the application died unexpectedly.

6. Test the modified Start and Stop script files.

7. Create the Recovery Program file (xxx.rp file).

   We named our Recovery Program file webserv.rp. We decided to have only one synchronization point and to execute the commands after the *event* and *other* key word in parallel. For more information about the content of this file, see B.2.1“The webserv.rp File” on page 203.

8. Save the existing rules.hacmprd file.

9. Change the rules.hacmprd file.

   We added the lines shown in Figure 58 on page 143 to the end of the rules.hacmprd file. For the resource variable, resource ID, event

expression and rearm expression we used the same values as in step 1 and step 2 on page 142. A listing of the complete rules.hacmprd file is available in B.2.5“The rules.hacmprd File” on page 206.

```
##### Beginning of Event Definition     WEBSERV Resource  ###########
#
UE_WEB_RESOURCE
0
/usr/local/cluster/events/webserv.rp
2
0
# 6) Resource variable only used for event manager events
IBM.PSSP.Prog.pcount
# 7) Instance vector, only used for event manager events
NodeNum=*;ProgName=ns-httpd;UserName=nobody
# 8) Predicate, only used for event manager events
X@0 == 0 && X@1 != 0
# 9) Rearm predicate, only used for event manager events
X@0 > 0
##### End of Event Definition      LPD Resource    ######
```

Figure 58. User-Defined Event Extensions to the rules.hacmprd File

10. Save the modified rules.hacmprd file.

    We recommend that you save the modified rules.hacmprd file in another directory. This is due to the fact that a PTF or an upgrade to a newer version may overwrite this file.

11. Stop HACMP on all nodes in this cluster.

    This is necessary because this file is read by the cluster manager only once on a node during the start of HACMP/ES.

12. Copy the files to all nodes in the cluster.

    Copy all the new and modified files to all nodes in this cluster. Make sure that all new script files are executable, and that they are in the same path location.

13. Start HACMP on one of the nodes.

14. Check the HACMP log file(s) to see that the start worked properly.

15. Repeat steps 13 to 14 until all nodes are up and running.

16. Test the new event.

6.3 Kerberos

Both HACMP/ES and HACMP for AIX require the file /.rhosts on every cluster node in order to perform functions like Cluster Synchronization, Cluster Verification, Dynamic Automatic Reconfiguration Events (DARE), and others. However, the /.rhosts file introduces a security hole by allowing users to execute the commands `/bin/rcp`, `/bin/rsh`, and `/bin/rlogin` without having to type the root password. Starting with HACMP/ES V4.2.1 and HACMP for AIX V4.2.1, it is now possible to configure HACMP to use Kerberos instead of the /.rhosts file on the RS/6000 SP.

Note: HACMP/ES requires a /.rhosts file on every cluster node. It must contain the Service and Boot adapters of all the nodes configured.

PSSP V3.1 currently implements MIT Kerberos Version 4. Kerberos provides an authenticated version of the `rcp` and `rsh` commands, and HACMP/ES V4.3 can be configured to use them, hence increasing overall system security. The authenticated `rsh` and `rcp` commands do not rely on the existence of the /.rhosts file.

For a detailed description of Kerberos, refer to the following documentation:

- *IBM Parallel System Support Programs for AIX Administration Guide*, SA22-7348.
- *RS/6000 Scalable POWERParallel Systems: PSSP Version 2 Technical Presentation*, SG24-4542.

In this section we explain how to configure Kerberos Version 4 under HACMP/ES V4.3. The RS/6000 SP has been installed with PSSP V3.1. There are two different procedures to configure Kerberos. One involves using the cl_setup_kerberos utility, while the other makes you execute all the commands manually from the command line. Our suggestion is to use the cl_setup_kerberos utility, which is less prone to human error. In fact, the manual procedure is quite long and complicated, but you can use this procedure for other than HACMP V4.3.

6.3.1 Cluster Configuration

In this section we use a 2-node cluster configuration, node sp21n13 and node sp21n15.

Node sp21n13 has the following network adapters:

- en0 (sp21n13, the SP Ethernet)
- en1 (n13_svc and n13_boot, the Service and Boot adapters)

- en2 (n13_stdby, the Standby adapter)
- css0 (sp21sw13, the SP Switch base address)
- css0 (sw13_svc and sw13_boot, the Service and Boot IP alias addresses)

Node sp21n15 has the following network adapters:

- en0 (sp21n15, the SP Ethernet)
- en1 (n15_svc and n15_boot, the Service and Boot adapters)
- en2 (n15_stdby, the Standby adapter)
- css0 (sp21sw15, the SP Switch base address)
- css0 (sw15_svc and sw15_boot, the Service and Boot IP alias addresses)

Figure 59 on page 145 summarizes all the adapters defined in our HACMP/ES cluster.

```
Adapter     Type       Network     Type     Attribute  Node      Interface
-------------------------------------------------------------------------

sw13_boot   boot       aliascss    hps      private    sp21n13    css0
sw13_svc    service    aliascss    hps      private    sp21n13    css0
sp21sw13    service    basecss     hps      private    sp21n13    css0
n13_boot    boot       ethernet1   ether    public     sp21n13    en1
n13_svc     service    ethernet1   ether    public     sp21n13    en1
n13_stdby   standby    ethernet1   ether    public     sp21n13    en2
sp21n13     service    spether     ether    private    sp21n13    en0
sw15_boot   boot       aliascss    hps      private    sp21n15    css0
sw15_svc    service    aliascss    hps      private    sp21n15    css0
sp21sw15    service    basecss     hps      private    sp21n15    css0
n15_boot    boot       ethernet1   ether    public     sp21n15    en1
n15_svc     service    ethernet1   ether    public     sp21n15    en1
n15_stdby   standby    ethernet1   ether    public     sp21n15    en2
sp21n15     service    spether     ether    private    sp21n15    en0
```

Figure 59. Adapters in Our AHCMP/ES Cluster

6.3.2 Configuring Kerberos Using the cl_setup_kerberos Utility

This section shows the procedure for configuring Kerberos by running a setup utility called cl_setup_kerberos.

To configure Kerberos:

1. Before running cl_setup_kerberos, make sure that HACMP/ES has been properly installed on both cluster nodes, sp21n13 and sp21n15. and Kerberos is installed and configured on each node, root.admin is

authenticated, and the /.k file is present. (This is the normal PSSP initial setup.)

2. On node sp21n13, configure the Cluster Topology in the usual manner. It is a requirement that the SP Ethernet adapter be part of the cluster configuration because this utility issues an authenticated rcmd to the cluster nodes through the en0.
3. At this point we execute cl_setup_kerberos on node sp21n13. This utility is located under /usr/sbin/cluster/sbin. The cl_setup_kerberos utility will prompt you to enter the password for every new Kerberos principal for all HACMP IP labels. This password can be the same as the Kerberos administration password, but does not have to be.
4. At this point change the Cluster Security mode from Standard to Enhanced.
5. Then issue a Cluster Topology Synchronization.
6. Delete the cl_krb_service file that was created as a result of step 3.
7. The final step is to remove the /.rhosts file from both cluster nodes.

Make sure you read 6.3.4, “Potential Problems when Using Kerberos” on page 158.

6.3.3 Configuring Kerberos Manually

In the following sections we see the step-by-step procedure for configuring Kerberos.

6.3.3.1 Defining the Adapters in the SDR

PSSP V3.1 automatically configures the rcmd service for all the adapters defined in the System Data Repository (SDR). The rcmd service enables the execution of the authenticated `rsh` and `rcp` commands for the network interfaces of these adapters.

When PSSP V3.1 was installed, we decided to define in the SDR only the SP Ethernet adapter (en0) and the SP Switch base address adapter (css0), as can be seen in Figure 60 on page 147, executing the `splstdata -a` command.

```
# splstdata -a | egrep "13|15"
                List LAN Database Information

node#  adapt           netaddr           netmask           hostname  type  rate
                          other_addrs
------------------------------------------------------------------------------
   13   css0     192.168.14.13      255.255.255.0   sp21sw13.msc.its    NA   NA
                          ""
   15   css0     192.168.14.15      255.255.255.0   sp21sw15.msc.its    NA   NA
                          ""
   13    en0      192.168.4.13      255.255.255.0   sp21n13.msc.itso   bnc   NA
                          ""
   15    en0      192.168.4.15      255.255.255.0   sp21n15.msc.itso   bnc   NA
                          ""
```

Figure 60. Initial Adapter Definition

The `ksrvutil list` command, executed on node sp21n13, reads the client srvtab file, /etc/krb-srvtab, and shows us the rcmd service for both the en0 adapter and the css0 adapter, as shown in Figure 61 on page 147.

```
# ksrvutil list
Version    Principal
   1     rcmd.sp21sw13@MSC.ITSO.IBM.COM
   1     rcmd.sp21n13@MSC.ITSO.IBM.COM
```

Figure 61. The Client srvtab on Node sp21n13

Our HACMP/ES cluster also includes the en1 and en2 adapters. The first step in configuring Kerberos is to define them in the SDR in order to have the rcmd service created. In Figure 62 on page 148 we see the SMIT menu used to define the en2 adapter of node sp21n13.

Note: The same operation must be performed for the en2 adapter of node sp21n15.

```
                    Additional Adapter Database Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                 [Entry Fields]
  Start Frame                                        [1]                      #
  Start Slot                                         [13]                     #
  Node Count                                         [1]                      #

  OR

  Node Group                                         []                       +

  OR

  Node List                                          []

* Adapter Name                                       [en2]
* Starting Node's IP Address or Hostname             [128.200.30.3]
* Netmask                                            [255.255.0.0]
  Additional IP Addresses                            []
  Ethernet Adapter Type                               bnc                     +
  Token Ring Data Rate                                                        +
  Skip IP Addresses for Unused Slots?                 no                      +
  Enable ARP for the css0 Adapter?                    yes                     +
  Use Switch Node Numbers for css0 IP Addresses?      yes                     +
[BOTTOM]

F1=Help            F2=Refresh         F3=Cancel          F4=List
F5=Reset           F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

Figure 62. Configure the en2 Adapter of Node sp21n13

Next, we configure the en1 adapter of node sp21n13 in the SDR, as shown in Figure 63 on page 149.

Note: The same operation must be performed for the en1 adapter of node sp21n15.

```
                    Additional Adapter Database Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
  Start Frame                                         [1]                        #
  Start Slot                                          [13]                       #
  Node Count                                          [1]                        #

  OR

  Node Group                                          []                         +

  OR

  Node List                                           []
* Adapter Name                                        [en1]
* Starting Node's IP Address or Hostname              [128.100.10.30]
* Netmask                                             [255.255.0.0]
  Additional IP Addresses                             [128.100.10.3]
  Ethernet Adapter Type                                                          +
  Token Ring Data Rate                                                           +
  Skip IP Addresses for Unused Slots?                  no                        +
  Enable ARP for the css0 Adapter?                     yes                       +
  Use Switch Node Numbers for css0 IP Addresses?       yes                       +
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 63. Configure the en1 Adapter on Node sp21n13

The en1 adapter must be configured with two IP addresses, Service and Boot. It is *important* to specify the Boot address in the field `Starting Node's IP Address or Hostname` because this is the IP address used to configure the en1 network interface when the cluster node is booted. The Service address must be specified in the field `Additional IP Addresses`, which is a new field introduced with PSSP V3.1. Prior to PSSP V3.1, it was not possible to use SMIT to assign multiple IP addresses to one single network interface in the SDR via the above SMIT screen. An alternative was to execute the following `SDRChangeAttrValues` command:

```
# SDRChangeAttrValues Adapter node_number==13 adapter_type==en1 \
  other_addrs=128.100.10.3
```

Note: SDRChangeAttrValues writes the 128.100.10.3 IP address in the other_addrs attribute of the Adapter SDR class. This attribute was introduced with PSSP V2.3.

We have now finished configuring all our Ethernet adapters, en0, en1 and en2. Next, we must configure the two SP Switch IP alias addresses. We use the SMIT menu Additional Adapter Database Information again. Figure 64 on page 150 shows how to define the Service and Boot IP alias addresses for the css0 adapter on node sp21n13.

Note: The same operation must be performed for the css0 adapter of node sp21n15.

```
                    Additional Adapter Database Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                 [Entry Fields]
  Start Frame                                        [1]                        #
  Start Slot                                         [13]                       #
  Node Count                                         [1]                        #

  OR

  Node Group                                         []                         +

  OR

  Node List                                          []
* Adapter Name                                       [css0]
* Starting Node's IP Address or Hostname             [192.168.14.13]
* Netmask                                            [255.255.255.0]
  Additional IP Addresses                            [140.40.4.13,140.40.4.3>
  Ethernet Adapter Type                                                         +
  Token Ring Data Rate                                                          +
  Skip IP Addresses for Unused Slots?                 no                        +
  Enable ARP for the css0 Adapter?                    yes                       +
  Use Switch Node Numbers for css0 IP Addresses?      no                        +
[BOTTOM]

F1=Help            F2=Refresh         F3=Cancel          F4=List
F5=Reset           F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

Figure 64. Define the Service and Boot IP Alias Addresses

6.3.3.2 Configuring the godm Service

After defining the rcmd service, we must now configure the *godm* service. The configuration of godm is performed on the command line by executing the `kadmin` command and then the `ank` (add new key) option.

Figure 65 on page 151 shows how to execute the `kadmin` commands.

Note: The `kadmin` command must be executed on the Control Workstation.

```
# kadmin
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin:  ank godm.sp21n13
Admin password:
Password for godm.sp21n13:
Verifying, please re-enter Password for godm.sp21n13:
godm.sp21n13 added to database.
admin:  ank godm.sp21n15
Admin password:
Password for godm.sp21n15:
Verifying, please re-enter Password for godm.sp21n15:
godm.sp21n15 added to database.
admin:  ank godm.sp21sw13
Admin password:
Password for godm.sp21sw13:
Verifying, please re-enter Password for godm.sp21sw13:
godm.sp21sw13 added to database.
admin:  ank godm.sp21sw15
Admin password:
Password for godm.sp21sw15:
Verifying, please re-enter Password for godm.sp21sw15:
godm.sp21sw15 added to database.
```

Figure 65. kadmin Command Example

The `ank` option must be used for all the IP labels (sp21n13, sp21sw13, n13_svc, n13_boot, n13_stdby, sw13_svc, sw13_boot, sp21n15, sp21sw15, n15_svc, n15_boot, n15_stdby, sw15_svc, sw15_boot) of both cluster nodes.

The next step is to extract a service key file from the Kerberos authentication database, again for all the IP labels of both cluster nodes. To perform this operation, we use the `ext_srvtab` command, as shown in Figure 66 on page 152.

Note: The `ext_srvtab` command must be executed on the Control Workstation.

```
# mkdir /tmp/kerby
# cd /tmp/kerby
# ext_srvtab -n sp21n13 sp21sw13 n13_svc n13_boot n13_stdby sw13_svc  \
               sw13_boot sp21n15 sp21sw15 n15_svc n15_boot n15_stdby  \
               sw15_svc sw15_boot
# ls -l
total 112
-rw-------   1 root     system        80 Jul 21 11:52 n13_boot-new-srvtab
-rw-------   1 root     system        82 Jul 21 11:52 n13_stdby-new-srvtab
-rw-------   1 root     system        78 Jul 21 11:52 n13_svc-new-srvtab
-rw-------   1 root     system        80 Jul 21 11:52 n15_boot-new-srvtab
-rw-------   1 root     system        82 Jul 21 11:52 n15_stdby-new-srvtab
-rw-------   1 root     system        78 Jul 21 11:52 n15_svc-new-srvtab
-rw-------   1 root     system        78 Jul 21 11:52 sp21n13-new-srvtab
-rw-------   1 root     system        78 Jul 21 11:52 sp21n15-new-srvtab
-rw-------   1 root     system        80 Jul 21 11:52 sp21sw13-new-srvtab
-rw-------   1 root     system        80 Jul 21 11:52 sp21sw15-new-srvtab
-rw-------   1 root     system        82 Jul 21 11:52 sw13_boot-new-srvtab
-rw-------   1 root     system        80 Jul 21 11:52 sw13_svc-new-srvtab
-rw-------   1 root     system        82 Jul 21 11:52 sw15_boot-new-srvtab
-rw-------   1 root     system        80 Jul 21 11:52 sw15_svc-new-srvtab

# ksrvutil list -f n13_svc-new-srvtab
Version    Principal
   1     godm.n13_svc@MSC.ITSO.IBM.COM
   1     rcmd.n13_svc@MSC.ITSO.IBM.COM
#
```

Figure 66. ext_srvtab Command Example

For each IP label specified on the command line, the `ext_srvtab` command creates a new service key file in the current working directory with a file name of iplabel-new-srvtab. Every single file contains the rcmd and godm service for that IP label.

6.3.3.3 Creating the Client srvtab File

All the iplabel-new-srvtab files created by the `ext_srvtab` command in the previous section must now be combined into one single file using the `cat` command, as shown in Figure 67 on page 152:

```
# cat n13_boot-new-srvt n13_stdby-new-srvtab n13_svc-new-srvtab \
> n15_boot-new-srvtab n15_stdby-new-srvtab n15_svc-new-srvtab \
> sp21n13-new-srvtab sp21n15-new-srvtab sp21sw13-new-srvtab \
> sp21sw15-new-srvtab sw13_boot-new-srvtab sw13_svc-new-srvtab \
> sw15_boot-new-srvtab sw15_svc-new-srvtab > ../big-srvtab
#
```

Figure 67. Create Single Client srvtab File

The file just created, big-srvtab, must now be transferred to all cluster nodes and renamed to /etc/krb-srvtab. The file /etc/krb-srvtab is often called the Client srvtab File.

Once transferred to all cluster nodes, we can have a look at the contents of /etc/krb-srvtab using the `ksrvutil` command. As can be seen in Figure 68 on page 153, it contains both the rcmd and godm services for all IP labels of both cluster nodes.

```
# ksrvutil list
Version    Principal
   1     godm.n13_boot@MSC.ITSO.IBM.COM
   1     rcmd.n13_boot@MSC.ITSO.IBM.COM
   1     godm.n13_stdby@MSC.ITSO.IBM.COM
   1     rcmd.n13_stdby@MSC.ITSO.IBM.COM
   1     godm.n13_svc@MSC.ITSO.IBM.COM
   1     rcmd.n13_svc@MSC.ITSO.IBM.COM
   1     rcmd.n15_boot@MSC.ITSO.IBM.COM
   1     godm.n15_boot@MSC.ITSO.IBM.COM
   1     godm.n15_stdby@MSC.ITSO.IBM.COM
   1     rcmd.n15_stdby@MSC.ITSO.IBM.COM
   1     rcmd.n15_svc@MSC.ITSO.IBM.COM
   1     godm.n15_svc@MSC.ITSO.IBM.COM
   1     rcmd.sp21n13@MSC.ITSO.IBM.COM
   1     godm.sp21n13@MSC.ITSO.IBM.COM
   1     godm.sp21n15@MSC.ITSO.IBM.COM
   1     rcmd.sp21n15@MSC.ITSO.IBM.COM
   1     godm.sp21sw13@MSC.ITSO.IBM.COM
   1     rcmd.sp21sw13@MSC.ITSO.IBM.COM
   1     godm.sp21sw15@MSC.ITSO.IBM.COM
   1     rcmd.sp21sw15@MSC.ITSO.IBM.COM
   1     godm.sw13_boot@MSC.ITSO.IBM.COM
   1     rcmd.sw13_boot@MSC.ITSO.IBM.COM
   1     rcmd.sw13_svc@MSC.ITSO.IBM.COM
   1     godm.sw13_svc@MSC.ITSO.IBM.COM
   1     godm.sw15_boot@MSC.ITSO.IBM.COM
   1     rcmd.sw15_boot@MSC.ITSO.IBM.COM
   1     godm.sw15_svc@MSC.ITSO.IBM.COM
   1     rcmd.sw15_svc@MSC.ITSO.IBM.COM
#
```

Figure 68. Show /etc/krb-srvtab via the ksrvutil Command

Another way to display the contents of the /etc/krb-srvtab file is the `klist` command with the `-srvtab` option, as seen in Figure 69 on page 154:

```
# klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
godm            n13_boot        MSC.ITSO.IBM.COM 1
rcmd            n13_boot        MSC.ITSO.IBM.COM 1
godm            n13_stdby       MSC.ITSO.IBM.COM 1
rcmd            n13_stdby       MSC.ITSO.IBM.COM 1
godm            n13_svc         MSC.ITSO.IBM.COM 1
rcmd            n13_svc         MSC.ITSO.IBM.COM 1
rcmd            n15_boot        MSC.ITSO.IBM.COM 1
godm            n15_boot        MSC.ITSO.IBM.COM 1
godm            n15_stdby       MSC.ITSO.IBM.COM 1
rcmd            n15_stdby       MSC.ITSO.IBM.COM 1
rcmd            n15_svc         MSC.ITSO.IBM.COM 1
godm            n15_svc         MSC.ITSO.IBM.COM 1
rcmd            sp21n13         MSC.ITSO.IBM.COM 1
godm            sp21n13         MSC.ITSO.IBM.COM 1
godm            sp21n15         MSC.ITSO.IBM.COM 1
rcmd            sp21n15         MSC.ITSO.IBM.COM 1
godm            sp21sw13        MSC.ITSO.IBM.COM 1
rcmd            sp21sw13        MSC.ITSO.IBM.COM 1
godm            sp21sw15        MSC.ITSO.IBM.COM 1
rcmd            sp21sw15        MSC.ITSO.IBM.COM 1
godm            sw13_boot       MSC.ITSO.IBM.COM 1
rcmd            sw13_boot       MSC.ITSO.IBM.COM 1
rcmd            sw13_svc        MSC.ITSO.IBM.COM 1
godm            sw13_svc        MSC.ITSO.IBM.COM 1
godm            sw15_boot       MSC.ITSO.IBM.COM 1
rcmd            sw15_boot       MSC.ITSO.IBM.COM 1
godm            sw15_svc        MSC.ITSO.IBM.COM 1
rcmd            sw15_svc        MSC.ITSO.IBM.COM 1
#
```

Figure 69. Show /etc/krb-srvtab via the klist Command

6.3.3.4 Updating the /.klogin File

The next step in the configuration consists of updating the /.klogin file on the Control Workstation. The root users /.klogin file contains a list of principals that are authorized to invoke processes as the root user with the authenticated `rsh` and `rcp` commands. The /.klogin file must be updated on the Control Workstation and then transferred to the HACMP/ES clusternodes. It must include all the principals shown by the `klist -srvtab` command.

Figure 70 on page 155 shows the /.klogin file after it was updated.

```
# cat /.klogin
root.admin@MSC.ITSO.IBM.COM
rcmd.sp21en0@MSC.ITSO.IBM.COM
rcmd.sp21n01@MSC.ITSO.IBM.COM
rcmd.sp21n05@MSC.ITSO.IBM.COM
rcmd.sp21n06@MSC.ITSO.IBM.COM
rcmd.sp21n07@MSC.ITSO.IBM.COM
rcmd.sp21n08@MSC.ITSO.IBM.COM
rcmd.sp21n09@MSC.ITSO.IBM.COM
rcmd.sp21n10@MSC.ITSO.IBM.COM
rcmd.sp21n11@MSC.ITSO.IBM.COM
rcmd.sp21n13@MSC.ITSO.IBM.COM
rcmd.sp21n15@MSC.ITSO.IBM.COM
godm.n13_boot@MSC.ITSO.IBM.COM
rcmd.n13_boot@MSC.ITSO.IBM.COM
godm.n13_stdby@MSC.ITSO.IBM.COM
rcmd.n13_stdby@MSC.ITSO.IBM.COM
godm.n13_svc@MSC.ITSO.IBM.COM
rcmd.n13_svc@MSC.ITSO.IBM.COM
rcmd.n15_boot@MSC.ITSO.IBM.COM
godm.n15_boot@MSC.ITSO.IBM.COM
godm.n15_stdby@MSC.ITSO.IBM.COM
rcmd.n15_stdby@MSC.ITSO.IBM.COM
rcmd.n15_svc@MSC.ITSO.IBM.COM
godm.n15_svc@MSC.ITSO.IBM.COM
rcmd.sp21n13@MSC.ITSO.IBM.COM
godm.sp21n13@MSC.ITSO.IBM.COM
godm.sp21n15@MSC.ITSO.IBM.COM
rcmd.sp21n15@MSC.ITSO.IBM.COM
godm.sp21sw13@MSC.ITSO.IBM.COM
rcmd.sp21sw13@MSC.ITSO.IBM.COM
godm.sp21sw15@MSC.ITSO.IBM.COM
rcmd.sp21sw15@MSC.ITSO.IBM.COM
godm.sw13_boot@MSC.ITSO.IBM.COM
rcmd.sw13_boot@MSC.ITSO.IBM.COM
rcmd.sw13_svc@MSC.ITSO.IBM.COM
godm.sw13_svc@MSC.ITSO.IBM.COM
godm.sw15_boot@MSC.ITSO.IBM.COM
rcmd.sw15_boot@MSC.ITSO.IBM.COM
godm.sw15_svc@MSC.ITSO.IBM.COM
rcmd.sw15_svc@MSC.ITSO.IBM.COM
#
```

Figure 70. The /.klogin File

6.3.3.5 Destroy and Reissue the Kerberos Tickets

The next step is to destroy the Kerberos tickets using the `kdestroy` command on both HACMP/ES cluster nodes, as shown in the following:

```
# kdestroy
Tickets destroyed.
#
```

Then stop and restart the Kerberos daemons on the Control Workstation, and destroy and reissue the Kerberos tickets using the `kinit` command, as follows:

```
# stopsrc -s kadmind
0513-044 The stop of the kadmind Subsystem was completed successfully.
# stopsrc -s kerberos
0513-044 The stop of the kerberos Subsystem was completed successfully.
# startsrc -s kerberos
0513-059 The kerberos Subsystem has been started. Subsystem PID is 12412.
# startsrc -s kadmind
0513-059 The kadmind Subsystem has been started. Subsystem PID is 34346.
#
# kdestroy
Tickets destroyed.
#
# kinit root.admin
Kerberos Initialization for "root.admin"
Password:
#
```

Next, reissue the Kerberos tickets on both HACMP/ES cluster nodes again using the `kinit` command:

```
# kinit root.admin
Kerberos Initialization for "root.admin"
Password:
#
```

Now we can execute the authenticated `rsh` between the HACMP/ES cluster nodes (sp21n13 and sp21n15) and the Control Workstation (sp21cw0), to make sure that the rcmd service is working properly:

```
# /usr/lpp/ssp/rcmd/bin/rsh sp21n15 date
Mon Jul 20 15:28:56 EDT 1998
#
# /usr/lpp/ssp/rcmd/bin/rsh sp21n13 date
Mon Jul 20 15:29:02 EDT 1998
#
# /usr/lpp/ssp/rcmd/bin/rsh sp21cw0 date
Mon Jul 20 15:29:23 EDT 1998
#
```

6.3.3.6 Change the Cluster Security Mode

In order for HACMP/ES to use Kerberos instead of the /.rhosts file, it is necessary to change the Cluster Security mode from the default value of Standard to Enhanced, as shown in Figure 71 on page 157.

```
                        Change / Show Cluster Security

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
* Cluster Security Mode                                 Enhanced               +
  WARNING: The /.rhosts file must be removed
  from ALL nodes in the cluster when the
  security mode is set to 'Enhanced'.
  Failure to remove this file makes it
  possible for the authentication server
  to become compromised. Once the server
  has been compromised, all authentication
  passwords must be changed.

  Changes to the cluster security mode
  setting alter the cluster topology
  configuration, and therefore need to be
  synchronized across cluster nodes. Since
  cluster security mode changes are seen as
  topology changes, they cannot be performed
  along with dynamic cluster resource
  reconfigurations.
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 71. Change Cluster Security

As explained in the Warning message of the SMIT menu, it is now safe to remove the /.rhosts file from the HACMP/ES cluster nodes and then synchronize the cluster topology.

6.3.3.7 Test HACMP/ES

The Kerberos configuration is now complete. To test that HACMP/ES works correctly, we suggest that you perform a Cluster Verification or a Topology DARE.

6.3.4 Potential Problems when Using Kerberos

Having HACMP/ES use Kerberos instead of the /.rhosts file is certainly a great advantage from the point of view of system security. However, there is a possibility that you can run into problems. A common task when administering an RS/6000 SP system is to customize the SP nodes. Node customization includes the execution of the `setup_server` command, which is responsible for creating the /etc/krb-srvtab files for all the RS/6000 SP nodes. Unfortunately, `setup_server` does not create the /etc/krb-srvtab files correctly, as required by HACMP/ES. After completing the Kerberos configuration, we have an /etc/krb-srvtab file as shown in Figure 72 on page 158 on the two cluster nodes sp21n13 and sp21n15:

```
# klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
godm            n13_boot        MSC.ITSO.IBM.COM 1
rcmd            n13_boot        MSC.ITSO.IBM.COM 1
godm            n13_stdby       MSC.ITSO.IBM.COM 1
rcmd            n13_stdby       MSC.ITSO.IBM.COM 1
godm            n13_svc         MSC.ITSO.IBM.COM 1
rcmd            n13_svc         MSC.ITSO.IBM.COM 1
rcmd            n15_boot        MSC.ITSO.IBM.COM 1
godm            n15_boot        MSC.ITSO.IBM.COM 1
godm            n15_stdby       MSC.ITSO.IBM.COM 1
rcmd            n15_stdby       MSC.ITSO.IBM.COM 1
rcmd            n15_svc         MSC.ITSO.IBM.COM 1
godm            n15_svc         MSC.ITSO.IBM.COM 1
rcmd            sp21n13         MSC.ITSO.IBM.COM 1
godm            sp21n13         MSC.ITSO.IBM.COM 1
godm            sp21n15         MSC.ITSO.IBM.COM 1
rcmd            sp21n15         MSC.ITSO.IBM.COM 1
godm            sp21sw13        MSC.ITSO.IBM.COM 1
rcmd            sp21sw13        MSC.ITSO.IBM.COM 1
godm            sp21sw15        MSC.ITSO.IBM.COM 1
rcmd            sp21sw15        MSC.ITSO.IBM.COM 1
godm            sw13_boot       MSC.ITSO.IBM.COM 1
rcmd            sw13_boot       MSC.ITSO.IBM.COM 1
rcmd            sw13_svc        MSC.ITSO.IBM.COM 1
godm            sw13_svc        MSC.ITSO.IBM.COM 1
godm            sw15_boot       MSC.ITSO.IBM.COM 1
rcmd            sw15_boot       MSC.ITSO.IBM.COM 1
godm            sw15_svc        MSC.ITSO.IBM.COM 1
rcmd            sw15_svc        MSC.ITSO.IBM.COM 1
#
```

Figure 72. /etc/krb-srvtab File on Both Nodes before Customization

As can be seen, we have listed both the rcmd and the godm service for all IP labels of both cluster nodes.

In Figure 73 on page 159, we see the contents of the /etc/krb-srvtab file on cluster node sp21n13 after it was customized.

```
# klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
godm            n13_svc         MSC.ITSO.IBM.COM 1
rcmd            n13_svc         MSC.ITSO.IBM.COM 1
godm            n13_boot        MSC.ITSO.IBM.COM 1
rcmd            n13_boot        MSC.ITSO.IBM.COM 1
godm            n13_stdby       MSC.ITSO.IBM.COM 1
rcmd            n13_stdby       MSC.ITSO.IBM.COM 1
godm            sw13_boot       MSC.ITSO.IBM.COM 1
rcmd            sw13_boot       MSC.ITSO.IBM.COM 1
rcmd            sw13_svc        MSC.ITSO.IBM.COM 1
godm            sw13_svc        MSC.ITSO.IBM.COM 1
godm            sp21sw13        MSC.ITSO.IBM.COM 1
rcmd            sp21sw13        MSC.ITSO.IBM.COM 1
rcmd            sp21n13         MSC.ITSO.IBM.COM 1
godm            sp21n13         MSC.ITSO.IBM.COM 1
#
```

Figure 73. /etc/krb-srvtab File on Node sp21n13 after Customization

It is clear that all the entries regarding the IP labels of cluster node sp21n15 are missing. The same problem occurs if we look at the /etc/krb-srvtab file on cluster node sp21n15. This time the entries of cluster node sp21n13 are missing, as shown in Figure 74 on page 159:

```
# klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
rcmd            n15_svc         MSC.ITSO.IBM.COM 1
godm            n15_svc         MSC.ITSO.IBM.COM 1
rcmd            n15_boot        MSC.ITSO.IBM.COM 1
godm            n15_boot        MSC.ITSO.IBM.COM 1
godm            n15_stdby       MSC.ITSO.IBM.COM 1
rcmd            n15_stdby       MSC.ITSO.IBM.COM 1
godm            sw15_boot       MSC.ITSO.IBM.COM 1
rcmd            sw15_boot       MSC.ITSO.IBM.COM 1
godm            sw15_svc        MSC.ITSO.IBM.COM 1
rcmd            sw15_svc        MSC.ITSO.IBM.COM 1
godm            sp21sw15        MSC.ITSO.IBM.COM 1
rcmd            sp21sw15        MSC.ITSO.IBM.COM 1
godm            sp21n15         MSC.ITSO.IBM.COM 1
rcmd            sp21n15         MSC.ITSO.IBM.COM 1
#
```

Figure 74. /etc/krb-srvtab File on Node sp21n15 after Customization

If we perform a Cluster Verification using the /etc/krb-srvtab files as they are created by the setup_server command, HACMP/ES does not validate the configuration, as shown in Figure 75 on page 160:

```
                                COMMAND STATUS

Command: failed        stdout: yes           stderr: no

before command completion, additional instructions may appear below.

[MORE...42]

Verifying Cluster Security
ERROR: Cannot find rcmd.sw15_svc@MSC.ITSO.IBM.COM in /etc/krb-srvtab on node sp2
1n13
ERROR: Cannot find godm.sw15_svc@MSC.ITSO.IBM.COM in /etc/krb-srvtab on node sp2
1n13
ERROR: Cannot find rcmd.sw15_boot@MSC.ITSO.IBM.COM in /etc/krb-srvtab on node sp
21n13
ERROR: Cannot find godm.sw15_boot@MSC.ITSO.IBM.COM in /etc/krb-srvtab on node sp
21n13
ERROR: Cannot find rcmd.sp21sw15@MSC.ITSO.IBM.COM in /etc/krb-srvtab on node sp2
1n13
ERROR: Cannot find godm.sp21sw15@MSC.ITSO.IBM.COM in /etc/krb-srvtab on node sp2
[MORE...116]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 75. Cluster Verification

The moral of the story is that the system administrator has to remember to restore a good, valid /etc/krb-srvtab file every time an HACMP/ES cluster node is customized.

There are three ways to restore a valid /etc/krb-srvtab file:

1. Execute the cl_setup_kerberos utility. Refer to 6.3.2, “Configuring Kerberos Using the cl_setup_kerberos Utility” on page 145.
2. Manually create a valid /etc/krb-srvtab. Refer to 6.3.3, “Configuring Kerberos Manually” on page 146.
3. Our suggestion is to maintain, on each cluster node, a second copy of a valid /etc/krb-srvtab file and name it, for example, /etc/krb-srvtab.ORIG. In case a node is customized, the /etc/krb-srvtab file is overwritten, but the system administrator will be able to quickly restore it by just renaming the /etc/krb-srvtab.ORIG file.

Chapter 7. Resource Management Considerations

This chapter discusses some topics about disk operations and TCP/IP considerations. We also provide unofficial techniques for configuring unsupported configurations. You may use these techniques at your own risk.

7.1 Shared Disk Operation

The Lazy update function was introduced in HACMP V4.2.0. This function allows HACMP to use the exportvg and importvg commands automatically when necessary to refresh the ODM on a node. It may use the -L flag for the importvg command, if it is available. With this function, the Logical Volume Manager (LVM) can maintain the running shared volume group (VG) without stopping HACMP. When an updated volume group is taken over, it takes a longer time for the exportvg and importvg commands.

AIX V4.3 added a number of enhancements to the LVM design. These make it easier to maintain shared and concurrent volume groups when they are online. HACMP V4.3 C-SPOC exploits these new LVM capabilities. If you use the HACMP V4.3 C-SPOC extension to make shared VG changes, the changes are propagated immediately to the other cluster nodes that have the VG as a resource. This removes the need for a Lazy update on the next takeover. You don't need Lazy update to update the ODM and other files in the related nodes. So, when the volume group takeover happens, which is updated by extended C-SPOC commands, HACMP releases the locks and only varies the volume group without using the exportvg and importvg commands. It reduces total down time.

7.1.1 New Flags for AIX Disk Operation Commands

The following two AIX commands, varyonvg and importvg, have new flags.

7.1.1.1 varyonvg

The varyonvg command has new flags to maintain shared volume groups. The -b flag breaks disk reservations on disks locked as a result of a normal varyonvg command. The -u flag varies on a volume group but leaves the disks that make up the volume group in an unlocked state. After a maintenance operation, you can lock the disks again by using the varyonvg command without the -b or the -u flag.

7.1.1.2 importvg

The importvg command has the new -L flag since AIX bos.rte.lvm.4.2.1.4. It allows Logical Volume Manager (LVM) to read the Volume Group Data Area

(VGDA) and Logical Volume Control Blocks (LVCBs) of a volume group that is not currently varied on, and updates ODM and other files appropriately without a varyon operation.

7.1.2 C-SPOC Operation

This section describes how C-SPOC uses the new flags to maintain shared volume groups.

7.1.2.1 The Basic Procedure

On the node that currently owns the volume group:

1. Make the necessary change to the volume group, for example, changing the file system mount point.
2. Update the saved time stamp:

   ```
   clvgdats /dev/hdiskX > /usr/sbin/cluster/etc/vg/SHAREDVG
   ```

   where hdiskX is the disk that belongs to the shared volume group, and SHAREDVG is the shared volume group name. This updates the saved time stamp so that, on HACMP restart, the Lazy update logic will not export and import (or importvg -L) the volume group in this node.
3. Remove the reserve on the shared volume group:

   ```
   varyonvg -b -u SHAREDVG
   ```

   This command removes the reserve on the shared volume group, leaving it varied on and accessible to applications.

On the other nodes in the resource group that currently do not own the volume group:

4. Update ODM and other files:

   ```
   importvg -L SHAREDVG hdiskY
   ```

   hdiskY is a disk in the volume group corresponding to the Physical Volume ID (PVID) that was passed by the initiating node, and SHAREDVG is the shared volume group name. This command updates the ODM and other files, such as /etc/filesystems, on the other nodes appropriately without a vary on operation. If a logical volume (LV) name or file system mount point name duplication happens, then the importvg command may fail.
5. Update the saved time stamp:

   ```
   clvgdats /dev/hdiskY > /usr/sbin/cluster/etc/vg/SHAREDVG
   ```

   where hdiskY is the disk that belongs to the shared volume group, and SHAREDVG is the shared volume group name. This command updates the

saved time stamp, so that, on failover, the Lazy update logic will not export and import (or `importvg -L`) the volume group in this node.

Finally, on the node that currently owns the volume group:

6. Restore the reserve on the initial node:

   `varyonvg SHAREDVG`

   where `SHAREDVG` is the shared volume group name. This restores the reserve of the disks.

7.1.2.2 Example

Following is an example of a C-SPOC operation. It changes the jfs mount point of a shared file system.

If you want to change the file system name also, not only the mount point, in the HACMP resource group, the following procedure is recommended:

- Before you change the file system, stop the application if necessary, remove the file system from the resource group, then execute the resource DARE.
- After you change the file system name, add the new file system and mount point to the resource group, execute the resource DARE, then start the application if necessary.
- If you do not remove the resource from resource group first, HACMP cannot remove (umount) the old file system because it does not exist in the AIX ODM. If you get this situation, umount the file system from the old mount point, then run `the clruncmd` command by using the `# smit cm_rec_aids -> Recover From Script Failure`. SMIT menu.

You can use `the cl_chfs` command or the SMIT menu. To start the SMIT menu, type `# smit cl_chjfs` on the command line.

```
      Change/Show Characteristics of a Shared File System in the Cluster

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Resource Group Name                                 rg13
  File system name                                    /fs13
  NEW mount point                                     [/fs13_2]
  SIZE of file system (in 512-byte blocks)            [40960]
  Mount GROUP                                         []
  Mount AUTOMATICALLY at system restart?              no                  +
  PERMISSIONS                                         read/write          +
  Mount OPTIONS                                       []                  +
  Start Disk Accounting?                              no                  +
  Fragment Size (bytes)                               4096
  Number of bytes per inode                           4096
  Compression algorithm                               no


F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Reset       Esc+6=Command     Esc+7=Edit        Esc+8=Image
Esc+9=Shell       Esc+0=Exit        Enter=Do
```

In this example, we are going to change the mount point from /fs13 to /fs13_2. Figure 76 shows the /tmp/cspoc.log log file after you change the file system mount point.

```
# cat /tmp/cspoc.log
07/21/98 11:58:01 [========= C_SPOC COMMAND LINE =========]
07/21/98 11:58:01 /usr/sbin/cluster/sbin/cl_chfs -cspoc -g rg13 -m /fs13_2 /fs13
07/21/98 11:58:04 sp21n13: success: /usr/sbin/cluster/cspoc/clresactive  -c clre
sactive:-V:4200::uname:-m
07/21/98 11:58:04 sp21n15: success: /usr/sbin/cluster/cspoc/clresactive  -c clre
sactive:-V:4200::uname:-m
07/21/98 11:58:07 sp21n13: success: clgetvg  -f /fs13
07/21/98 11:58:11 sp21n13: success: clresactive  -v datavg13
07/21/98 11:58:11 sp21n15: success: clresactive  -v datavg13
07/21/98 11:58:16 sp21n13: success: chfs  -m /fs13_2 /fs13
07/21/98 11:58:18 sp21n13: success: clupdatevgts  datavg13
07/21/98 11:58:19 sp21n13: success: lspv
07/21/98 11:58:22 sp21n13: success: varyonvg  -n -b -u datavg13
07/21/98 11:59:11 sp21n15: success: clupdatevg  datavg13 00000590d27569f0
07/21/98 11:59:15 sp21n13: success: varyonvg  -n datavg13
```

Figure 76. The /tmp/scpoc.log Log File

This log file shows you how C-SPOC uses AIX commands with flags to change the file system mount point. The `clupdatevgts` command, after the `chfs` command, updates the datavg13 time stamp on node sp21n13. It uses

the `clvgdats` command explained in “Update the saved time stamp:” in "The Basic Procedure” on page 162.

The next `varyonvg -b -u datavg13` command releases the locks on the disks of datavg13. Refer to “Remove the reserve on the shared volume group:” in "The Basic Procedure” on page 162.

In the next line, on node sp21n15, the `clupdatevg` command changes the PVID to the hdisk name. This command internally executes `importvg -L` to update ODM and other files (refer to “Update ODM and other files:” in "The Basic Procedure” on page 162). Then it updates the datavg13 time stamp on node sp21n15 by using the `clvgdats` command (refer to “Update the saved time stamp:” in "The Basic Procedure” on page 162).

Finally, on node sp21n13, the `varyonvg` command restores the reserve (refer to “Restore the reserve on the initial node:” in "The Basic Procedure” on page 162).

The following screen shows the status after you changed the file system mount point:

```
# dsh -w sp21n13,sp21n15 lsfs /fs13
sp21n13: lsfs: 0506-915 No record matching /fs13 was found in /etc/filesystems.
sp21n15: lsfs: 0506-915 No record matching /fs13 was found in /etc/filesystems.
# dsh -w sp21n13,sp21n15 lsfs /fs13_2
sp21n13: Name            Nodename   Mount Pt              VFS   Size    Options
    Auto Accounting
sp21n13: /dev/lv13       --         /fs13_2               jfs   40960   rw
    no   no
sp21n15: Name            Nodename   Mount Pt              VFS   Size    Options
    Auto Accounting
sp21n15: /dev/lv13       --         /fs13_2               jfs   --      rw
    no   no

# dsh -w sp21n13,sp21n15 odmget -q value=/fs13_2 CuAt
sp21n13:
sp21n13: CuAt:
sp21n13:        name = "lv13"
sp21n13:        attribute = "label"
sp21n13:        value = "/fs13_2"
sp21n13:        type = "R"
sp21n13:        generic = "DU"
sp21n13:        rep = "s"
sp21n13:        nls_index = 640
sp21n15:
sp21n15: CuAt:
sp21n15:        name = "lv13"
sp21n15:        attribute = "label"
sp21n15:        value = "/fs13_2"
sp21n15:        type = "R"
sp21n15:        generic = "DU"
sp21n15:        rep = "s"
sp21n15:        nls_index = 640

# dsh -w sp21n13 lsvg -l datavg13 | grep fs13
sp21n13: lv13               jfs        5     5     1    open/syncd    /fs13_2
# dsh -w sp21n13 lspv | grep datavg13
sp21n13: hdisk2         00000590d27569f0    datavg13
sp21n13: hdisk3         00000590d275717f    datavg13
# dsh -w sp21n13 /usr/sbin/cluster/utilities/clvgdats hdisk2
sp21n13: 35b4b94c04140200
# dsh -w sp21n13,sp21n15 cat /usr/sbin/cluster/etc/vg/datavg13
sp21n13: 35b4b94c04140200
sp21n15: 35b4b94c04140200
```

You can find that the /etc/filesystems, ODM CuAt, and volume group time stamp are updated on both nodes.

7.2 Single Network Adapter Configuration

If there is no adapter slot for an HACMP standby adapter available, you cannot configure the cascading IP Address Takeover (IPAT) configuration

officially. We strongly recommend that you have a standby adapter. But, if this is impossible, there are some unofficial configuration examples.

A Rotating configuration is the only officially supported configuration for a single network adapter environment. So this case is not covered in this section.

7.2.1 Basic Situation

In order to configure a single network adapter configuration, the previous HACMP releases required the /usr/sbin/cluster/etc/netmon.cf file, which includes IP addresses or IP labels that can reply to Internet Control Message Protocol (ICMP) Echo Message. HACMP/ES V4.3 uses the ICMP Echo Message for a network broadcast address if necessary. With this function, users do not need to use the netmon.cf file in many cases. You can test whether you need this file or not with the `ping` command.

```
# ifconfig en0
en0: flags=e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT>
        inet 80.7.6.71 netmask 0xffff0000 broadcast 80.7.255.255
# ping -c2 80.7.255.255
PING 80.7.255.255: (80.7.255.255): 56 data bytes
64 bytes from 80.7.6.71: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 80.7.6.30: icmp_seq=0 ttl=255 time=3 ms (DUP!)
64 bytes from 80.7.6.90: icmp_seq=0 ttl=255 time=4 ms (DUP!)
64 bytes from 80.7.6.10: icmp_seq=0 ttl=255 time=5 ms (DUP!)
64 bytes from 80.7.6.71: icmp_seq=1 ttl=255 time=0 ms

----80.7.255.255 PING Statistics----
2 packets transmitted, 2 packets received, +3 duplicates, 0% packet loss
round-trip min/avg/max = 0/2/5 ms
```

In this example, the `ping` command (ICMP Echo Message) gets replies from three IP addresses: 80.7.6.30, 80.7.6.90, and 80.7.6.10, other than the local network interface, 80.7.6.71. This means you do not need the notmon.cf file. DUP! message shows that the `ping` command gets many replies. This is normal in this case.

If a router or another network machine, is running, they reply to the ICMP Echo Message. So if you have these machines, the netmon.cf file is not required.

If the ICMP Echo Message uses the broadcast address, then recent AIX versions do not respond to it by default. You can check this attribute and change it with the `no` command.

```
# no -a | grep bcastping
                bcastping = 0
# no -o bcastping=1
# no -a | grep bcastping
                bcastping = 1
```

If the bcastping is 1, AIX will reply to the ICMP Echo Message of a broadcast address. If it is 0, AIX will not reply. The default value is 0. This value is not kept when the system reboots. So, if you want to set the value after reboot, you need to add the no command to the /etc/rc.net file.

In the case of an HACMP/ES V4.3 node, topsvcs sets the bcastping value to 1 when it starts.

7.2.2 Single Adapter IP Address Takeover Examples

Traditionally in HACMP, when you need to implement IP address takeover using a single network adapter, you need to use a rotating resource group. This type of HACMP configuration does not use a separate standby adapter to move a failed server's IP address to, so can accomplish the single adapter IP address takeover in a standard way. The behavior of a rotating resource group on reintegration of the originally failed server into the cluster, however, is for that server to assume the backup role, and not to immediately reacquire the resources (IP address included) that it had before it failed. This may be satisfactory, or it may be that there is a requirement for the reintegrating server to get its resources back immediately, as it would in a cascading resource group configuration.

The following sections introduce examples of possible configurations to accomplish IP address takeover using a single network adapter. Described are the standard rotating resource group method, as well as other methods using a combination of application servers, ifconfig aliases, and customized events.

For all of these examples, it should be remembered that if you are using a single network adapter, that adapter is a single point of failure (SPOF). Therefore there should be an error notification on the failure of that adapter that will do a recovery action, such as shutting down the node with the takeover option.

7.2.2.1 Using a Rotating Resource Group

A rotating resource group does not require a standby adapter to do IP address takeover. Therefore it is a good candidate for single adapter IP

address takeover. With a rotating resource group though, if you want the resources to be returned to the original node upon its reentry into the cluster, you must do this manually. This can either be done by shutting down all of the nodes, and restarting them in the order in which you want them to acquire resources, or more likely, in the following way:

1. Reintegrate the server node as a standby node.
2. On the backup node, do an HACMP shutdown with takeover option.
3. Restart HACMP on the backup node, so that it resumes the standby role.

You can also make an N:M takeover configuration using rotating resource groups. For example, you could configure a cluster with five primary servers and two backup servers. Again, with a rotating resource group configuration, the roles of nodes as primaries or backups are not fixed. They are based on the order in which nodes enter and leave the cluster. If it is important to you for a certain node to serve a certain resource group, you must manage that by having the nodes enter the cluster at the correct time. If you want to have a serving node return to a backup role, you can use the three-step procedure described above to accomplish that.

The following is a rotating resource group configuration example:

```
                Configure Resources for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Resource Group Name                            risc71rg
  Node Relationship                              rotating
  Participating Node Names                       risc71 sp21n11
  Service IP label                              [svcip]               +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Application Servers                           [APPL1]               +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure 77. Resource Configuration (1)

In this example, there is one *rotating* resource group with one service IP label.

7.2.2.2 Using Application Server and Ifconfig Alias

In this configuration, HACMP does not takeover the IP address by itself. The application server script can control the alias address, which is independent from HACMP. In this configuration, HACMP does not control the alias address.

This configuration requires only a service adapter. You do not need both a boot and standby adapters.

The following is an adapter topology example:

```
Adapter      Type     Network    Net Type  Attribute  Node     IP Address
risc71_svc   service  ethernet2  ether     public     risc71   80.7.6.71
n11_svc      service  ethernet2  ether     public     sp21n11  80.7.6.11

netmask:255.255.0.0, risc71_alias:80.7.6.77
```

In this example, there are two nodes, risc71 and sp21n11. They have only one Ethernet adapter each. Each adapter has its own service address - risc71_svc has 80.7.6.71 and n11_svc has 80.7.6.11.

The following is a resource configuration example:

```
                Configure Resources for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Resource Group Name                            risc71rg
  Node Relationship                              cascading
  Participating Node Names                       risc71 sp21n11
  Service IP label                              []                  +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Application Servers                            [APPL1]             +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

In this example, there is one *cascading* resource group with no service IP labels.

The following is an example of application server scripts:

- Start script: /usr/sbin/cluster/local/START_SVR1

```
#!/bin/ksh
set -x
##############################################################################
# acquire alias address
##############################################################################
ALIAS_ADDR="risc71_alias"
NETWORK_NAME=ethernet2
#NETMASK=255.255.0.0

SVC_ADDR=‘/usr/sbin/cluster/utilities/cllsif -Si $LOCALNODENAME| \
awk ’($3 == "’$NETWORK_NAME’") && ($2 == "service") {print $7}’‘
IF=‘/usr/sbin/cluster/utilities/clgetif -a $SVC_ADDR‘
NETMASK=‘/usr/sbin/cluster/utilities/clgetif -n $SVC_ADDR‘

/usr/sbin/ifconfig $IF inet $ALIAS_ADDR netmask $NETMASK alias up

##############################################################################
# main (application start)
##############################################################################



exit 0
```

- Stop script: /usr/sbin/cluster/local/STOP_SVR1

```
#!/bin/ksh
set -x
##############################################################################
# main (application stop)
##############################################################################



##############################################################################
# release alias address
##############################################################################
ALIAS_ADDR="risc71_alias"
NETWORK_NAME=ethernet2

SVC_ADDR=‘/usr/sbin/cluster/utilities/cllsif -Si $LOCALNODENAME| \
awk ’($3 == "’$NETWORK_NAME’") && ($2 == "service") {print $7}’‘
IF=‘/usr/sbin/cluster/utilities/clgetif -a $SVC_ADDR‘

/usr/sbin/ifconfig $IF inet $ALIAS_ADDR delete

exit 0
```

The alias address will be added by the application server start script, and removed by the stop script. These scripts use the `ifconfig` command to control alias addresses.

You need to specify the takeover alias IP label as ALIAS_ADDR, and the network name as NETWORK_NAME. These scripts use the network name to decide the network interface name and netmask. If you want to use a netmask that is different from the service adapter netmask, you mast specify it in the start script.

The following is the IP address takeover scenario for this configuration:

1. Start HACMP on the server node, risc71. Then the application server start script, START_SVR1, uses the `ifconfig alias` command. It adds an alias address, risc71_alias, to the server node service address, risc71_svc.
2. Start HACMP on the backup node, sp21n11. Then the node waits for the server node failure.
3. If the server node fails, the backup node gets the cascading resource group, risc71rg. The resource group starts the application server start script, START_SVR1, which adds the shared service address, risc71_alias, to the backup node service address, n11_svc.
4. When the failed node comes back, the backup node releases the cascading resource group, risc71rg. The resource group removes the alias address by using the STOP_SVR1 script. Then the server node gets the resource group, risc71rg, which adds an alias address, risc71_alias, to the server node service address, risc71_svc, again.

   These operations are done by HACMP automatically.
5. To stop the HACMP cluster, stop HACMP with the graceful option on both nodes. Then HACMP releases all resource groups on both nodes, that is, it removes alias addresses by using STOP_SVR1 script.

7.2.2.3 Using Application Server and IPAT Events

This is another configuration sample. The topology configuration is something like a rotating configuration without a standby adapter. However, it uses a cascading resource group and shell script to control the shared service IP address. This configuration is simple and the takeover IP address is observed and controlled by HACMP, but the acquire_service_addr recovery script is not triggered directly by a change on the clstrmgr state.

The following is the adapter topology example:

```
Adapter      Type     Network    Net Type  Attribute  Node     IP Address
risc71_boot  boot     ethernet2  ether     public     risc71   80.7.6.20
risc71_svc   service  ethernet2  ether     public              80.7.6.71
n11_boot     boot     ethernet2  ether     public     sp21n11  80.7.6.10
risc71_svc   service  ethernet2  ether     public              80.7.6.71

netmask:255.255.0.0
```

In this topology example, there are two nodes, risc71 and sp21n11. They have only one Ethernet adapter each. Each adapter has its own boot address - risc71_boot has 80.7.6.20 and n11_boot has 80.7.6.10. There is one shared service address, risc71_svc.

The following is the resource configuration example:

```
                Configure Resources for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Resource Group Name                            risc71rg
  Node Relationship                              cascading
  Participating Node Names                       risc71 sp21n11
  Service IP label                              []                +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Application Servers                           [APPL2]           +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

In this example, there is one *cascading* resource group with no service IP label. The IP label is controlled by the application server scripts. Of course, you can add any other resources such as file systems, which are not related to the IP label.

In this configuration, if you verify the HACMP resources, you will get the following warning messages, because a service IP address is not configured as a resource. We are going to control the service IP address, risc71_svc, with application server scripts, so this message is expected.

```
Verifying Configured Resources...
WARNING: The service IP label (risc71_svc) on node risc71 is not configured
to be part of a resource group. Therefore it will not be acquired and used as
a service address by any node.
WARNING: The service IP label (risc71_svc) on node sp2n15 is not configured
to be part of a resource group. Therefore it will not be acquired and used as
a service address by any node.
```

The following is an example of application server scripts:

- Start script: /usr/sbin/cluster/local/START_SVR2

```
#!/bin/ksh
set -x
##############################################################################
# acquire service address
##############################################################################
SERVICE_LABEL="risc71_svc"
/usr/sbin/cluster/events/cmd/clcallev acquire_service_addr "$SERVICE_LABEL"

##############################################################################
# main (application start)
##############################################################################



exit 0
```

This script acquires the shared service address before the application starts.

- Stop script: /usr/sbin/cluster/local/STOP_SVR2

```
#!/bin/ksh
set -x
##############################################################################
# main (application stop)
##############################################################################



##############################################################################
# release service address
##############################################################################
SERVICE_LABEL="risc71_svc"
/usr/sbin/cluster/events/cmd/clcallev release_service_addr "$SERVICE_LABEL"

exit 0
```

This script releases the shared service address after the application stops.

Application server scripts use acquire_service_addr and release_service_addr recovery scripts by calling the `clcallev` command. These scripts handle the service IP address and notify the Topology Services subsystem.

You must specify the shared service IP label name as SERVICE_LABEL.

The following is the IP address takeover scenario for this configuration:

1. Start HACMP at server node, risc71. Then the application server start script, START_SVR2, calls the acquire_service_addr recovery script. The recovery script changes the server node boot address, risc71_boot, to the shared service address, risc71_svc.
2. Start HACMP on the backup node, sp21n11. The node waits for a server node failure.
3. If the server node fails, the backup node acquires the cascading resource group, risc71rg, which starts the application server start script, START_SVR2. The start script changes the backup node boot address, n11_boot, to the shared service address, risc71_svc.
4. When the failed node comes back, the backup node releases the cascading resource group, risc71rg, which resource group releases the shared service address, risc71_svc, by using the STOP_SVR2 stop script. Then the server node acquires the resource group, risc71rg, the resource group which acquires the shared service address, risc71_svc.

   These operations are done by HACMP automatically.
5. To stop the HACMP cluster, stop HACMP with the graceful option on both nodes. Then HACMP releases all resource groups on both nodes; that is, it releases the shared service address by using the STOP_SVR2 stop script.

7.2.2.4 Using Event Customize and IPAT Events

The configuration in "Using Application Server and IPAT Events” on page 172 can work, but the acquire_service_addr recovery script is not related to the with clstrmgr states such as RP_RUNNING and BARRIER. See Figure 6 on page 34 for further details. This section describes the other method that is related to the with the HACMP Cluster Manager state.

The topology configuration is the same configuration as in the previous example. It is like a rotating configuration without a standby adapter. However, it uses a cascading resource group and an event customization (pre, post) shell script to control the shared service IP address.

The following is the adapter topology example:

```
Adapter      Type     Network    Net Type  Attribute  Node     IP Address
risc71_boot  boot     ethernet2  ether     public     risc71   80.7.6.20
risc71_svc   service  ethernet2  ether     public              80.7.6.71
n11_boot     boot     ethernet2  ether     public     sp21n11  80.7.6.10
risc71_svc   service  ethernet2  ether     public              80.7.6.71

netmask:255.255.0.0
```

In this topology example, there are two nodes, risc71 and sp21n11. They have only one Ethernet adapter each. Each adapter has its own boot address - risc71_boot has 80.7.6.20 and n11_boot has 80.7.6.10. There is one shared service address, risc71_svc.

The following is the resource configuration example:

```
                Configure Resources for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Resource Group Name                            risc71rg
  Node Relationship                              cascading
  Participating Node Names                       risc71 sp21n11
  Service IP label                              []                  +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

In this example, there is one *cascading* resource group with no service IP label. The IP label is controlled by the event customize shell script. Of course, you can add application servers and any other resources such as file systems, which are not related to the IP label.

In this configuration, if you verify HACMP resources, you will get the following warning messages, because a service IP address is not configured. We are going to control the service IP address, risc71_svc, with the event customization shell script, so this message is expected.

```
Verifying Configured Resources...
WARNING: The service IP label (risc71_svc) on node risc71 is not configured
to be part of a resource group. Therefore it will not be acquired and used as
a service address by any node.
WARNING: The service IP label (risc71_svc) on node sp2n15 is not configured
to be part of a resource group. Therefore it will not be acquired and used as
a service address by any node.
```

The following is the Add a Custom Cluster Event example:

```
                        Add a Custom Cluster Event

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Cluster Event Name                                 [SINGLE_ADAPTER_IPAT]
* Cluster Event Description                          [single adapter ipat sample]
* Cluster Event Script Filename                      [/usr/sbin/cluster/local/SI>]
```

The Cluster Event Script Filename is `/usr/sbin/cluster/local/SINGLE_ADAPTER_IPAT` for this example.

The following are the event customize examples:

```
                        Change/Show Cluster Events

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]

  Event Name                                          node_up_local
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Pre-event Command                                  [SINGLE_ADAPTER_IPAT]    +
  Post-event Command                                 []                       +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
                        Change/Show Cluster Events

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]

  Event Name                                          node_down_remote
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Pre-event Command                                  [SINGLE_ADAPTER_IPAT]    +
  Post-event Command                                 []                       +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
                        Change/Show Cluster Events

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]

  Event Name                                     node_down_local
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Pre-event Command                             []                      +
  Post-event Command                            [SINGLE_ADAPTER_IPAT]   +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
                        Change/Show Cluster Events

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]

  Event Name                                     node_up_remote
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Pre-event Command                             []                      +
  Post-event Command                            [SINGLE_ADAPTER_IPAT]   +
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The following is the example of the event customization (pre, post) shell script:

- /usr/sbin/cluster/local/SINGLE_ADAPTER_IPAT

```
#!/bin/ksh
set -x
##############################################################################
# node_up_local, node_down_remote pre-event (acquire_service_addr)
# node_down_local, node_up_remote post-event (release_service_addr)
##############################################################################
GROUP="risc71rg"
SERVICE_LABEL="risc71_svc"

EVENTNAME=$1
if [ $EVENTNAME = "node_down_local" -o $EVENTNAME = "node_up_remote" ]; then
        RC=$2; shift
fi
TO_GROUP=$2
GREASE=$3

if [ $GROUP != $TO_GROUP ]; then
        exit 0
fi

odmget -q "group=$GROUP" HACMPgroup | grep nodes | awk '{print $3 " " $4}' | \
sed 's/"//g'  | read SERVER BACKUP

if [[   ( $LOCALNODENAME = $SERVER && $EVENTNAME = "node_up_local"   ) || \
        ( $LOCALNODENAME = $BACKUP && $EVENTNAME = "node_down_remote" && \
                                        "$GREASE" = "" ) ]]; then

        /usr/sbin/cluster/events/cmd/clcallev acquire_service_addr "$SERVICE_LABEL"

elif [[ ( $LOCALNODENAME = $SERVER && $EVENTNAME = "node_down_local" ) || \
        ( $LOCALNODENAME = $BACKUP && $EVENTNAME = "node_down_local" ) || \
        ( $LOCALNODENAME = $BACKUP && $EVENTNAME = "node_up_remote"  ) ]]; then

        /usr/sbin/cluster/events/cmd/clcallev release_service_addr "$SERVICE_LABEL"

fi

exit 0
```

This script controls the shared service IP address.

You need to specify the cascading resource group name by GROUP and the shared IP label name by SERVICE_LABEL. LOCALNODENAME is set by HACMP.

The outline of this script is as follows:

1. If the local node is the server node and the event is node_up_local, then acquire the service address.
2. If the local node is the backup node and the event is node_down_remote (without the graceful option), then acquire the service address.
3. If the event is node_down_local, then release the service address.

4. If the local node is backup node and the event is node_up_remote, then release the service address.

The following is the IP address takeover scenario for this configuration:

1. Start HACMP on the server node, risc71. The node_up_local pre event script, SINGLE_ADAPTER_IPAT, calls the acquire_service_addr recovery script. The recovery script changes the server node boot address, risc71_boot, to the shared service address, risc71_svc.
2. Start HACMP on the backup node, sp21n11. The node waits for a server node failure.
3. If the server node fails, on the backup node, the node_down_remote pre event script, SINGLE_ADAPTER_IPAT, calls the acquire_service_addr recovery script. The recovery script changes the backup node boot address, n11_boot, to the shared service address, risc71_svc.
4. When the failed server node comes back, on the backup node, the node_up_remote post event script, SINGLE_ADAPTER_IPAT, calls the release_service_addr recovery script. The recovery script releases the shared service address, risc71_svc. Then, at the server node, the node_up_local pre event script acquires the shared service address.
5. When the HACMP cluster stops, the node_down_local post event script, SINGLE_ADAPTER_IPAT, calls the release_service_addr recovery script. The recovery script releases the service address, risc71_svc. The node_down_remote graceful script does not do a takeover.

7.3 Cascading by Using Standby and Aliasing

In this section we describe the solution for a problem we encountered. This problem is not unique to HACMP/ES, it can happen on all HACMP versions. The solution we are going to describe here is only a possible workaround and may not work in all cases.

7.3.1 Situation

We used a three-node cascading relationship for all three resource groups in Cluster B (the hardware layout is shown in A.2‘Hardware Configuration” on page 190), but we had only one standby adapter. Therefore, if two nodes went down, the takeover for the second failing node failed because there was no longer a standby adapter available. The following sections describe the function, the advantages and disadvantages of our enhancement.

7.3.2 Our Workaround

The workaround we chose is based on the solution described in the redbook *HACMP Enhanced Scalability*, SG24-2081 (Chapter 14, "Cascading by Using One Network Adapter"). The original scripts where designed by Simon Marchese (IBM UK). It was necessary to change the acquire_takeover_addr and release_takeover_addr scripts only. We could use the cl_alias_IP_address and cl_unalias_IP_address scripts unchanged. If you are interested in more about HACMP IP Address Aliasing code, you can have the original from IBM intranet at `http://max4.sbank.uk.ibm.com`.

We enhanced the functionality of these scripts to make it fit our environment and requirements. The scripts now check if there are standby adapters defined or if the defined ones are free to use. If no standby adapter is defined or the defined one is not available, then the aliasing functionality is used. The following sections give you more information about the advantages and disadvantages of our solution. The files we used are listed in B.3‘Modified HACMP Scripts” on page 210

7.3.2.1 Technical Description

IP Address Takeover (IPAT) is one of the major functions of HACMP. The technique used to support IPAT in HACMP is IP address swapping. When a service address needs to be taken by a node, either through node failure or maintenance, that address is swapped onto a spare network adapter, known as a *standby interface*, whose own address is first discarded. By using an alternative technique known as IP address aliasing, the requirement for network adapters can be reduced to one adapter per node by avoiding discarding addresses on takeover.

In this case, both techniques are brought together. The IP address swapping technique is used as long as there is a standby interface available. If no standby interface is available, the IP address aliasing technique is used. The alias address will always be assigned to the primary service address of the backup node. If both techniques are in use and the standby becomes free due to a rejoin of a failed node, the aliased address stays where it is, and there will be no IP address swapping.

7.3.2.2 Advantages

This solution has the following advantages:

- The enhancement reduces the minimum number of network adapters required to support a cascading resource group to one per node.
- In configurations of more than two nodes, multiple standby adapters are not required if multiple takeovers need to be supported. Takeover can be

caused by node failure and graceful shutdown (with takeover) for maintenance.

- This enhancement will especially help in large HACMP/ES V4.3 configurations with multinode configurations, especially as the nodes can be limited in the available number of adapter slots.

7.3.2.3 Disadvantages

This solution has the following disadvantages:

- By utilizing IP address aliasing, we are relying on a single network adapter to support multiple IP addresses. Currently, a single network adapter can only support one hardware address (or MAC address, as it is also known). That means that IP address aliasing cannot support Hardware Address Takeover (HWAT).
- We now may have only one network adapter configured on a given network. In this case, HACMP cannot determine whether a heartbeat failure over this network is due to failure of the network itself or of the adapter. This is only a problem in HACMP clusters where there are only two nodes active. This may be because there are only two nodes in the cluster or because the other nodes are not currently active, either through failure or graceful shutdown for maintenance.
- The system is more likely to suffer a performance bottleneck at the network adapter because we are now supporting multiple IP addresses on a single network adapter.

7.3.2.4 System Requirements

The enhancement has been partially tested with HACMP/ES V4.3. However, the usual HACMP implementation testing should be performed.

Note: We did not have enough time to do all the necessary tests. Therefore, the scripts may contain some bugs, but for all the tests we did they worked well.

In order to test whether your cluster configuration will support IP address aliasing, the enhancement may be simulated by using the `ifconfig` and `netstat` commands. Perform the test on an HACMP cluster node that has been taken out of the HACMP cluster for maintenance, or on a non-HACMP clustered RS/6000.

Make sure that the correct address is removed from the `netstat` command output by the `ifconfig` commands. If the results are as expected, the enhancement is likely to work, since that is pretty much what it does in the

code. If the results are not as expected, there is a problem and the enhancement may not work successfully.

For an example of what you should see when testing your system, refer to the output listed in Figure 78 on page 184.

```
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          100     0      100     0     0
lo0   1536  127         localhost           100     0      100     0     0
en0   1500  <Link>                            0     0        0     0     0
en0   1500  192.9.200   192.9.200.1           0     0        0     0     0
# ifconfig en0 detach
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          100     0      100     0     0
lo0   1536  127         localhost           100     0      100     0     0
# ifconfig en0 192.9.200.1 up
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          101     0      101     0     0
lo0   1536  127         localhost           101     0      101     0     0
en0   1500  <Link>                            0     0        0     0     0
en0   1500  192.9.200   192.9.200.1           0     0        0     0     0
# ifconfig en0 alias 192.9.200.2 up
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          101     0      101     0     0
lo0   1536  127         localhost           101     0      101     0     0
en0   1500  <Link>                            0     0        0     0     0
en0   1500  192.9.200   192.9.200.1           0     0        0     0     0
en0   1500  192.9.200   192.9.200.2           0     0        0     0     0
# ifconfig en0 delete 192.9.200.2
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          101     0      101     0     0
lo0   1536  127         localhost           101     0      101     0     0
en0   1500  <Link>                            0     0        0     0     0
en0   1500  192.9.200   192.9.200.1           0     0        0     0     0
# ifconfig en0 alias 192.9.200.2 up
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          101     0      101     0     0
lo0   1536  127         localhost           101     0      101     0     0
en0   1500  <Link>                            0     0        0     0     0
en0   1500  192.9.200   192.9.200.1           0     0        0     0     0
en0   1500  192.9.200   192.9.200.2           0     0        0     0     0
# ifconfig en0 delete 192.9.200.1
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                          101     0      101     0     0
lo0   1536  127         localhost           101     0      101     0     0
en0   1500  <Link>                            0     0        0     0     0
en0   1500  192.9.200   192.9.200.2           0     0        0     0     0
#
```

Figure 78. Testing IP Aliasing

7.3.2.5 Support

The enhancement is supplied "as is." No commitment to support the enhancement is implied. Customers should not contact their local support organization for help with this enhancement unless a prior agreement has been made.

7.3.2.6 Installation

There are two ways to install this enhancement. For your convenience, the enhancement is packaged as a tar format archive, stored with absolute path names from /usr/local/cluster/events down. This one is as ip_alias_fullp.tar. For all of you who do not like to have it in this path, we stored the same files with relative path names from ./ down. This one is ip_alias_relp.tar. These files are normally located in the /usr/local/cluster/sample directory if you follow the procedures described in B‘Script Files Used In This Book” on page 201. For more information about the installation process, see "Using the Modified Event Scripts” on page 186.

Overwrite existing files

Installation should be performed while the cluster is down. The enhancement consists of enhanced versions of the acquire_takeover_addr and release_takeover_addr events and two new utilities, cl_alias_IP_address and cl_unalias_IP_address. These may be copied over the existing versions (which we do not recommend), because any subsequent fix to HACMP may replace the new events, so a check should be made before reintegrating an upgraded node into an active cluster. HACMP usually renames any replaced events as <eventname>.ORIG when installing a fix, so the enhanced events will not be lost, but their function will be lost and they may not be compatible with other functions contained in the fix.

Note: If using this method, do not forget to save the original files.

Using Alternative Path

An alternative method is to install the new events elsewhere, for example, into a new directory such as /usr/local/cluster/events. The HACMP events can then be changed through SMIT to point to the new event scripts. As mentioned in the previous paragraph, any subsequent fix to HACMP may overwrite the path name in the ODM of the new events, so a check should be made before reintegrating an upgraded node into an active cluster.

7.3.2.7 Configuration

Depending on the method of installation, you have two different configuration procedures:

- By using the overwrite method (see "Overwrite existing files” on page 185), there are no additional configuration steps required. What you have to change is the path information for the AliasPGM_Path variable in the acquire_takeover_addr and release_takeover_addr event scripts.
- By using the alternative path method ("Using Alternative Path” on page 185), you have to modify the path information for the acquire_takeover_addr and release_takeover_addr events. This change is performed through SMIT, using the standard HACMP SMIT panels.

The enhancement takes effect in two cases:

- When no standby adapters are configured on a given node for an HACMP network. Therefore, only service and boot adapter labels should be configured to enable the enhancement.
- When at least one standby adapter is configured on a given node for an HACMP network, and this adapter is already used by another service address due to a node failure or graceful shutdown (with takeover) for maintenance.

7.3.2.8 Using the Modified Event Scripts

As previously mentioned, there are two ways to install the modified scripts: You can overwrite the current scripts or copy them to a local directory. To keep the number of necessary changes as low as possible, even if HACMP is going to be updated, we recommend that you use a local directory for modifications like this one.

Before you can follow one of the following sections to implement the enhancement, you have to get the script files as mentioned in B‘Script Files Used In This Book” on page 201.

Using the ip_alias_fullp.tar File

By using the ip_alias_fullp.tar file, you may do the installation as follows:

1. Logon as user *root*.
2. The following command will automatically place the files into the /usr/local/cluster/events and /usr/local/cluster/events/utils directories:

   ```
   tar -xf /usr/local/sample/ip_alias_fullp.tar
   ```

Note: The /usr/local/cluster/events directory will be automatically created if it does not exist.

3. Now you need to modify the cluster events for acquire_takeover_addr and release_takeover_addr, either by using the command:

   ```
   /usr/sbin/cluster/utilities/clchevent
   ```

or via SMIT:

```
# smit hacmp
=> Cluster Configuration
===> Cluster Resources
=====> Cluster Events
=======> Change/Show Cluster Events
```

Modify the path for the event command so that /usr/local/cluster/events is used. For an example of how the screen looks for the acquire_takeover_addr event see, Figure 79 on page 187.

```
 Change/Show Cluster Events

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]

  Event Name                                    acquire_takeover_addr

  Description                                   Script run to configure a standby a>

* Event Command                                [/usr/local/cluster/events/acquire_t>

  Notify Command                                []
  Pre-event Command                             []                                  +
  Post-event Command                            []                                  +
  Recovery Command                              []
* Recovery Counter                              [0]                                  #



F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 79. HACMP/ES V4.3 Change/Show Cluster Events

Using the ip_alias_relp.tar File

If you do not want to have the files in the /usr/local/cluster/events directory, perhaps you would like to overwrite the original ones. To do this, you have to use the ip_alias_relp.tar file. You may do the installation as follows:

1. Logon as user *root*.
2. Change your actual directory to the one where you like to have the code installed.
3. The following command will place the files into the local directory and in the ./utils directory:

   ```
   tar -xf /usr/local/sample/ip_alias_relp.tar
   ```

Note: The ./utils directory will be automatically created if it does not exist.

4. Depending on the directory you choose, you have to continue with different steps:
   - If you used /usr/sbin/cluster/events or /usr/es/sbin/cluster/events as your local directory, then continue with step 5.
   - In all other cases, you need to modify the cluster events for acquire_takeover_addr and release_takeover_addr, either by using the command:

     ```
     /usr/sbin/cluster/utilities/clchevent
     ```

     or via SMIT:

     ```
     # smit hacmp
     => Cluster Configuration
     ===> Cluster Resources
     =====> Cluster Events
     =======> Change/Show Cluster Events
     ```

     Modify the path for the event command so that the directory you chose is used. An example of this screen is shown in Figure 79 on page 187.
5. Now change the AliasPGM_Path variable in the acquire_takeover_addr and release_takeover_addr event scripts. The AliasPGM_Path variable has to be set to the full path location of the cl_alias_IP_address and cl_unalias_IP_address scripts.

7.3.2.9 Removing or Deactivating this Enhancement

This section gives you a brief overview of how to deactivate or remove this enhancement. Depending on the installation method you choose, you have to do different steps to remove or deactivate this solution:

- Removing when the Overwrite Method was used

The only thing you have to do here is to copy the saved original files back.

- Removing when the Alternative Path Method was used

  Here you can choose between deactivating and removing:
  - To deactivate, you have to use the SMIT screen Change/Show Cluster Events to change the path information back to the original one.
  - To delete, do the deactivation first and then remove the file.

Appendix A. Our Environment

This appendix shows the environment we used to get the results documented in this book. Some of the examples may use slightly different environments. In these cases, we describe the differences.

A.1 Hardware and Software

Hardware

- RS/6000 SP wide node
- RS/6000 model 530

Software

- IBM Parallel System Support Programs for AIX Version 3 Release 1
- IBM HACMP Enhanced Scalability Version 4 Release 3
- IBM AIX Version 4 Release 3

Note

We did not use the GA versions of these products.

A.2 Hardware Configuration

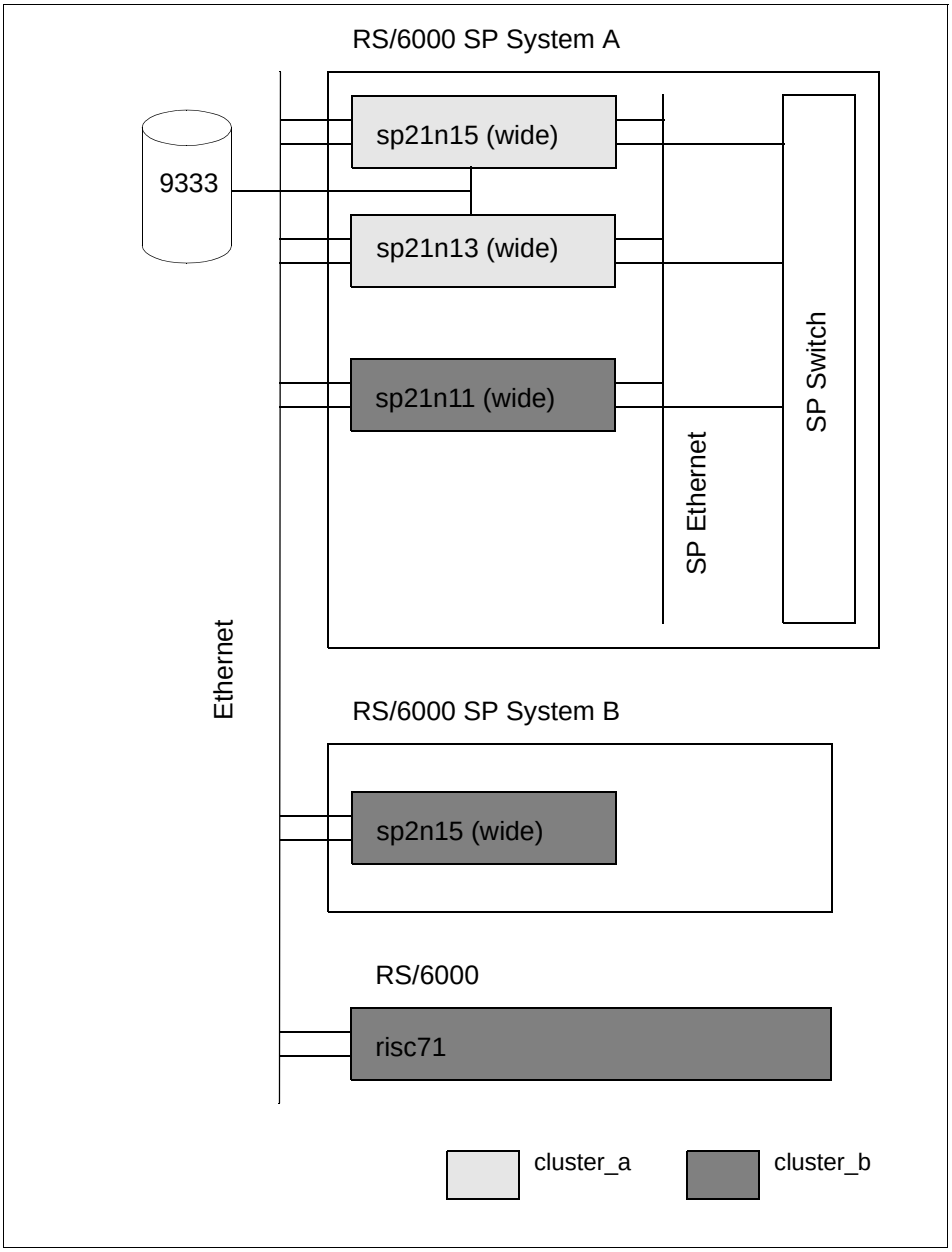


Figure 80. Hardware Configuration

A.3 Cluster cluster_a

The cluster cluster_a uses the following nodes:

- sp21n13 (SP node)
- sp21n15 (SP node)

A.3.1 TCP/IP Networks

Cluster ID 1

Cluster Name cluster_a

Table 4. cluster_a TCP/IP Networks

Network Name	Network Type	Network Attribute	Netmask
ethernet1	ether	public	255.255.0.0
spether	ether	public	255.255.255.0
basecss	hps	private	255.255.255.0
aliascss	hps	private	255.255.255.0

A.3.2 TCP/IP Network Adapters

Node Name sp21n13

Table 5. cluster_a TCP/IP Network Adapters for sp21n13

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Network Attribute
en1	n13_svc	svc	128.100.10.3	ethernet1	public
en1	n13_boot	boot	128.100.10.30	ethernet1	public
en2	n13_stdby	stdby	128.200.30.3	ethernet1	public
en0	sp21n13	svc	192.168.4.13	spether	private
css0	so21sw13	svc	192.168.14.13	basecss	private
css0	sw13_svc	svc	140.40.4.13	aliascss	private
css0	sw13_boot	boot	140.40.4.33	aliascss	private

Node Name sp21n15

Table 6. cluster_a TCP/IP Network Adapters for sp21n15

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Network Attribute
en1	n15_svc	svc	128.100.10.1	ethernet1	public
en1	n15_boot	boot	128.100.10.10	ethernet1	public
en2	n15_stdby	stdby	128.200.20.2	ethernet1	public
en0	sp21n15	svc	192.168.4.15	spether	private
css0	sp21sw15	svc	192.168.14.15	basecss	private
css0	sw15_svc	svc	140.40.4.15	aliascss	private
css0	sw15_boot	boot	140.40.4.55	aliascss	private

A.3.3 Shared Logical Volumes

Disk Device 9333

Table 7. cluster_a Shared Logical Volumes

Node Name	Physical Volume	Volume Group	File System	Logical Volume	Major Number
sp21n13	hdisk2, hdisk3	datavg13	/fs13	lv13	50
sp21n15	hdisk4	datavg15	/fs15	lv15	60

A.3.4 Cluster Resource Groups

A Mutual Takeover configuration is used in cluster_a. Two Cascading Resource Groups have been configured. One is rg13 and contains the resources shown in Figure 81 on page 193:

```
# /usr/sbin/cluster/utilities/clshowres -g rg13

Resource Group Name                             rg13
Node Relationship                               cascading
Participating Node Name(s)                      sp21n13 sp21n15
Service IP Label                                n13_svc sw13_svc
HTY Service IP Label
Filesystems                                     /fs13
Filesystems Consistency Check                   fsck
Filesystems Recovery Method                     sequential
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
AIX Connections Services
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
SSA Disk Fencing                                false
Filesystems mounted before IP configured        false


Run Time Parameters:

Node Name                                       sp21n13
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       sp21n15
Debug Level                                     high
Host uses NIS or Name Server                    false
```

Figure 81. Cluster Resource Group rg13

The second Resource Group is rg15 and contains the resources shown in Figure 82 on page 194:

```
# /usr/sbin/cluster/utilities/clshowres -g rg15

Resource Group Name                             rg15
Node Relationship                               cascading
Participating Node Name(s)                      sp21n15 sp21n13
Service IP Label                                n15_svc sw15_svc
HTY Service IP Label
Filesystems                                     /fs15
Filesystems Consistency Check                   fsck
Filesystems Recovery Method                     sequential
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
AIX Connections Services
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
SSA Disk Fencing                                false
Filesystems mounted before IP configured        false


Run Time Parameters:

Node Name                                       sp21n15
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       sp21n13
Debug Level                                     high
Host uses NIS or Name Server                    false
```

Figure 82. Resource Cluster Group rg15

A.4 Cluster cluster_b

The cluster cluster_b uses the following nodes:

- sp21n11 (RS/6000 SP wide node)
- sp2n15 (RS/6000 SP wide node)
- risc71 (RS/6000)

A.4.1 TCP/IP Networks

Cluster ID 2

Cluster Name cluster_b

Table 8. cluster_b TCP/IP Networks

Network Name	Network Type	Network Attribute	Netmask
ethernet2	ether	public	255.255.0.0

A.4.2 TCP/IP Network Adapters

Node Name sp21n11

Table 9. cluster_b TCP/IP Network Adapters for sp21n11

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Global Network Name	Network Attribute
en1	n11_svc	svc	80.7.6.11	ethernet2		public
en1	n11_boot	boot	80.7.6.10	ethernet2		public
en2	n11_stdby	stdby	80.9.9.1	ethernet2		public

Node Name sp2n15

Table 10. cluster_b TCP/IP Network Adapters for sp2n15

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Global Network Name	Network Attribute
en1	n152_svc	svc	80.7.6.31	ethernet2		public
en1	n152_boot	boot	80.7.6.30	ethernet2		public
en2	n152_stdby	stdby	80.9.9.3	ethernet2		public

Node Name risc71

Table 11. cluster_b TCP/IP Network Adapters for risc71

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Global Network Name	Network Attribute
en0	risc71_svc	svc	80.7.6.71	ethernet2		public

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Global Network Name	Network Attribute
en0	risc71_boot	boot	80.7.6.20	ehternet2		public
en1	risc71_stdby	stdby	80.9.9.2	ethernet2		public

A.4.3 Cluster Resource Groups

We had three Cascading Resource Groups configured. The relationship was mutual takeover, with two possible backup nodes. The names of the resource groups are sp21n11rg, risc71rg and sp2n15rg.

Figure 83 on page 197 shows the configuration of the resource group sp21n11rg.

```
# /usr/sbin/cluster/utilities/clshowres -g sp21n11rg

Resource Group Name                             sp21n11rg
Node Relationship                               cascading
Participating Node Name(s)                      sp21n11 risc71 sp2n15
Service IP Label                                n11_svc
HTY Service IP Label
Filesystems                                     /fs11
Filesystems Consistency Check                   fsck
Filesystems Recovery Method                     sequential
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
AIX Connections Services
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
SSA Disk Fencing                                false
Filesystems mounted before IP configured        false
Run Time Parameters:

Node Name                                       sp21n11
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       risc71
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       sp2n15
Debug Level                                     high
Host uses NIS or Name Server                    false

#
```

Figure 83. Resource Group sp21n11rg

Figure 84 on page 198 shows the configuration of resource group *risc71rg.*

```
# /usr/sbin/cluster/utilities/clshowres -g risc71rg

Resource Group Name                             risc71rg
Node Relationship                               cascading
Participating Node Name(s)                      risc71 sp21n11 sp2n15
Service IP Label                                risc71_svc
HTY Service IP Label
Filesystems                                     /fs71
Filesystems Consistency Check                   fsck
Filesystems Recovery Method                     sequential
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
AIX Connections Services
Application Servers                             env
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
SSA Disk Fencing                                false
Filesystems mounted before IP configured        false
Run Time Parameters:

Node Name                                       risc71
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       sp21n11
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       sp2n15
Debug Level                                     high
Host uses NIS or Name Server                    false

#
```

Figure 84. Resource Group risc71rg

Figure 85 on page 199 shows the configuration of resource group *sp2n15rg*.

```
# /usr/sbin/cluster/utilities/clshowres -g sp2n15rg

Resource Group Name                             sp2n15rg
Node Relationship                               cascading
Participating Node Name(s)                      sp2n15 sp21n11 risc71
Service IP Label                                n152_svc
HTY Service IP Label
Filesystems
Filesystems Consistency Check                   fsck
Filesystems Recovery Method                     sequential
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
AIX Connections Services
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
SSA Disk Fencing                                false
Filesystems mounted before IP configured        false
Run Time Parameters:

Node Name                                       sp2n15
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       sp21n11
Debug Level                                     high
Host uses NIS or Name Server                    false

Node Name                                       risc71
Debug Level                                     high
Host uses NIS or Name Server                    false

#
```

Figure 85. Resource Group sp2n15rg

Appendix B. Script Files Used In This Book

The examples in this appendix are available at an FTP site and via the World Wide Web.

Note

These examples have been tested on pre-release versions of HACMP/ES V4.3, IBM RS/6000 Cluster Technology Version 1.1 and IBM Parallel System Support Programs for AIX V3.1. They may not be suitable for use in a particular environment, and are not supported by IBM in any way. IBM is not responsible for your use of these examples.

1. FTP Site

The files are available for anonymous FTP from www.redbooks.ibm.com. To retrieve the files using FTP, you must have access to the Internet. You can use the following procedure (Figure 86 on page 201):

```
# cd /tmp
# ftp www.redbooks.ibm.com
ftp> bin
ftp> cd /redbooks/SG245328
ftp> get disc5328.tar.Z
ftp> quit
#
# uncompress disc5328.tar.Z
# mkdir -p /usr/local/cluster
# cd /usr/local/cluster
# tar -xf /tmp/disc5328.tar
# rm /tmp/disc5328.tar
#
```

Figure 86. Installing Examples by Using FTP

2. WWW Site

The examples can also be downloaded using the World Wide Web. The URL `www.redbooks.ibm.com` provides details on the procedure. You can use one of the following examples:

`ftp://www.redbooks.ibm.com/redbooks/SG245328`

or

`http://www.redbooks.ibm.com`

and then select *Additional Materials*.

B.1 Application Start and Stop Scripts

This section contains the start and stop scripts we used for our application (Netscape FastTrack Server).

B.1.1 Start Script

```
#!/usr/bin/ksh
#
# This Program (webserv_start) starts the WEB server function
#
# (C) COPYRIGHT International Business Machines Corp. 1998
#

PGM=ns-httpd
PGMPath=/usr/local/www-home/bin/httpd
PGMOpt="-d /usr/local/www-home/httpd-sp21n07/config"
PGMDIR=/usr/local/cluster/bin
TMPDIR=/usr/local/cluster/tmp

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

# creating test directory for User Defined Event

if [ ! -d /usr/local/cluster/tmp ]
then
   mkdir -p /usr/local/cluster/tmp
fi

print "$(date) Starting \"$PGM\" "

# remove file which may be left from a failed User Event
rm -f $TMPDIR/$PGM.failure > /dev/null 2>&1

# Start the application
$PGMPath/$PGM $PGMOpt

sleep 2
ChkPGM=$(ps -e | grep $PGM | wc -l)

if [ $ChkPGM != 0 ]
then
   # Program is up and running creating test file for User Defined Event
   touch /usr/local/cluster/tmp/$PGM.on
   print "$(date) The $PGM server function is up and running"
else
   # Program is NOT up and running ! Try to restart
   $PGMPath/$PGM $PGMOpt
   sleep 2
   ChkPGM=$(ps -e | grep $PGM | wc -l)
   if [ $ChkPGM != 0 ]
   then
      touch /usr/local/cluster/tmp/$PGM.on
      print "$(date) The $PGM server function is up and running"
   else
      print "$(date) The Start off \"$PGM program failed"
      exit 4
   fi
```

```
fi

exit 0
```

B.1.2 Stop Script

```
#!/usr/bin/ksh
#
# This Program (webserv_stop) stops the WEB server function
#
# (C) COPYRIGHT International Business Machines Corp. 1998
#

PGM=ns-httpd

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

print "$(date) Stopping \"$PGM\" program"

# Selecting the right process number to stop the application
StopID=$(ps -ef |grep httpd | awk '{print $2 " " $3}' | grep " 1" | awk '{print
$1}')
kill $StopID > /dev/null 2>&1
rm -f /usr/local/cluster/tmp/$PGM.on > /dev/null 2>&1

# Check if application is stopped if not try it again
ChkPGM=$(ps -e | grep $PGM | wc -l)

if [ $ChkPGM != 0 ]
then
   StopID=$(ps -ef |grep httpd | awk '{print $2 " " $3}' | grep " 1" | awk '{print $1}')
   kill $StopID > /dev/null 2>&1
   ChkPGM=$(ps -e | grep $PGM | wc -l)
   if [ $ChkPGM != 0 ]
   then
      print "$(date) Stopping off \"$PGM\" program failed"
      exit 4
   fi
else
   print "$(date) Program \"$PGM\" successfully stopped"
fi

exit 0
```

B.2 Files for User-Defined Events

This section contains the files we needed to get the User Event for our application (Netscape FastTrack Server) working.

B.2.1 The webserv.rp File

```
#
# This file contains the HACMP ES recovery program for
# the user defined "webserv" event
#
# (C) COPYRIGHT International Business Machines Corp. 1998
```

```
#
# format:
# relationship     command to run   expected status NULL
#
other "/usr/local/cluster/bin/webserv_remote" 0 NULL
event "/usr/local/cluster/bin/webserv_local" 0 NULL
#
barrier
#
all "/usr/local/cluster/bin/webserv_complete" X NULL
```

B.2.2 The webserv_remote Script

```
#!/usr/bin/ksh
#
# This Program (webserv_remote) prints a message to the logfile
# on the remote Nodes
#
# (C) COPYRIGHT International Business Machines Corp. 1998
#

UtilPath=/usr/sbin/cluster/utilities

PGM=ns-httpd

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

HA_NodeName=$($UtilPath/clhandle -h $EVLOCATION | awk '{print $2}')

print "$TIMESTAMP $EVNAME detected a failure of the \"$PGM\" program"
print "The failure occurred on the  remote SP-Node $HA_NodeName"
print ""

exit 0
```

B.2.3 The webserv_local Script

```
#!/usr/bin/ksh
#
# This Program (webserv_local) prints a message to the logfile
# and restarts the failed application
#
# (C) COPYRIGHT International Business Machines Corp. 1998
#

set -x
UtilPath=/usr/sbin/cluster/utilities

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

PGM=ns-httpd
PGMPath=/usr/local/www-home/bin/httpd
PGMOpt="-d /usr/local/www-home/httpd-sp21n07/config"
PGMDIR=/usr/local/cluster/bin
TMPDIR=/usr/local/cluster/tmp
```

```
############################################################
# Start_Takeover
############################################################
function Start_Takeover
{

sleep 5
/usr/sbin/cluster/utilities/clstop -y -N -gr
exit 0

}

############################################################
# NoReExecute
############################################################
function NoReExecute
{

sleep 20
rm -f $TMPDIR/$PGM.failure
exit 0

}

############################################################
# MAIN
############################################################

HA_NodeName=$($UtilPath/clhandle -h $EVLOCATION | awk '{print $2}')

print "$TIMESTAMP $EVNAME detected a failure of the \"$PGM\" program"
print "The failure occurred on the local SP-Node $HA_NodeName"
print ""

if [  -f $TMPDIR/$PGM.on ]
then
   if [  -f $TMPDIR/$PGM.failure ]
   then
      print "$(date) Recovery of \"$PGM\" program failure"
      print "terminated due to too quick occurrence of failure"
      Start_Takeover &
      print "Takeover initiated !!"
      exit 0
   fi
   print "$(date) Going to restart the \"$PGM\" program"
   $PGMPath/$PGM $PGMOpt
   /usr/bin/touch $TMPDIR/$PGM.failure
   NoReExecute &
   sleep 1
   ChkPGM=$(ps -e | grep $PGM | wc -l)
   if [ $ChkPGM != 0 ]
   then
      print "$(date) The failed \"$PGM\"  program successfully restarted"
   else
      print "$(date) The restart of the \"$PGM\" program failed !!"
      wait
      /usr/bin/touch $TMPDIR/$PGM.failure
      Start_Takeover &
      print "Takeover initiated !!"
   fi
else
   print "$(date) Recovery of \"$PGM\" program failure"
   print "terminated due to a stop of HACMP"
```

```
fi

exit 0
```

B.2.4 The webserv_complete Script

```
#!/usr/bin/ksh
#
# This Program (webserv_complete) prints a message to the logfile
# on all Nodes
#
# (C) COPYRIGHT International Business Machines Corp. 1998
#

UtilPath=/usr/sbin/cluster/utilities

PGM=ns-httpd

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

HA_NodeName=$($UtilPath/clhandle -h $EVLOCATION | awk ’{print $2}’)

print "$(date) Recovery program for the \"$PGM\" application failure handling completed
on $HA_NodeName"

exit 0
```

B.2.5 The rules.hacmprd File

```
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 43haes430 src/43haes/usr/sbin/cluster/events/rules.hacmprd 1.1
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1996,1997
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# ##############################################################################
#
# "@(#)29   1.1   src/43haes/usr/sbin/cluster/events/rules.hacmprd, hacmp.pe, 43
haes430 11/7/96 13:41:37"
#
#   COMPONENT_NAME: EVENTS
#
#   FUNCTIONS: none
#
#   ORIGINS: 27
#
#
#   (C) COPYRIGHT International Business Machines Corp. 1990,1994
#   All Rights Reserved
#   Licensed Materials - Property of IBM
#   US Government Users Restricted Rights - Use, duplication or
```

```
#   disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# ###################################################################
#
# This file contains the HACMP/PE recovery program to event mapping
#
# format:  (1) name
#          (2) state (qualifier)
#          (3) recovery program path
#          (4) recovery type  (Reserved for future use)
#          (5) recovery level (Reserved for future use)
#          (6) resource variable name (Used for Event Manager events)
#          (7) instance vector (Used for Event Manager events)
#          (8) predicate (Used for Event Manager events)
#          (9) rearm predicate (Used for Event Manager events)
#
##### Beginning of Event Definition  Node Up ###############
#
TE_JOIN_NODE
0
/usr/sbin/cluster/events/node_up.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition       Node Up  ######
#
#
##### Beginning of Event Definition     Node Down  ##########
#
TE_FAIL_NODE
0
/usr/sbin/cluster/events/node_down.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Node Down    ######
#
#
##### Beginning of Event Definition     Network Up  ##########
#
TE_JOIN_NETWORK
0
/usr/sbin/cluster/events/network_up.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events
```

```
# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Network Up    ######
#
#
##### Beginning of Event Definition     Network Down  ##########
#
TE_FAIL_NETWORK
0
/usr/sbin/cluster/events/network_down.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Network Down    ######
#
#
##### Beginning of Event Definition     Swap Adapter  ##########
#
TE_SWAP_ADAPTER
0
/usr/sbin/cluster/events/swap_adapter.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Swap Adapter    ######
#
#
##### Beginning of Event Definition     Join Stabdby  ##########
#
TE_JOIN_STANDBY
0
/usr/sbin/cluster/events/join_standby.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Join Standby    ######
#
#
##### Beginning of Event Definition     Fail Standby  ##########
#
```

```
TE_FAIL_STANDBY
0
/usr/sbin/cluster/events/fail_standby.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Fail Standby    ######
#
#
##### Beginning of Event Definition     DARE Topology  ##########
#
TE_DARE_TOPOLOGY
0
/usr/sbin/cluster/events/reconfig_topology.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      DARE Topology    ######
#
#
##### Beginning of Event Definition     DARE Resource  ##########
#
TE_DARE_RESOURCE
0
/usr/sbin/cluster/events/reconfig_resource.rp
2
0
# 6) Resource variable only used for event managemer events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      DARE Resource    ######
#
#
##### Beginning of Event Definition     WEBSERV Resource  ##########
#
UE_WEB_RESOURCE
0
/usr/local/cluster/events/webserv.rp
2
0
# 6) Resource variable only used for event manager events
IBM.PSSP.Prog.pcount
# 7) Instance vector, only used for event manager events
NodeNum=*;ProgName=ns-httpd;UserName=nobody
```

```
# 8) Predicate, only used for event manager events
X@0 == 0 && X@1 != 0
# 9) Rearm predicate, only used for event manager events
X@0 > 0
##### End of Event Definition      LPD Resource    ######
#
```

B.3 Modified HACMP Scripts

This section contains the additional and modified script we used for our work around described in 7.3, “Cascading by Using Standby and Aliasing” on page 180.

B.3.1 The HACMP Script acquire_takeover_addr

```
#!/bin/ksh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 43haes430 src/43haes/usr/sbin/cluster/events/acquire_takeover_addr.sh 1.16
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1998
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# @(#)66        1.18  src/43haes/usr/sbin/cluster/events/acquire_takeover_addr.sh,
hacmp.events, 43haes430, 9816A_43ha430 4/24/98 08:55:52
#
#   COMPONENT_NAME: EVENTS
#
#   FUNCTIONS: name_to_addr
#
#   ORIGINS: 27
#
#
#   (C) COPYRIGHT International Business Machines Corp. 1990,1994
#   All Rights Reserved
#   Licensed Materials - Property of IBM
#   US Government Users Restricted Rights - Use, duplication or
#   disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
################################################################################
#                                                                              #
#       Name:           acquire_takeover_addr                                  #
#                                                                              #
#       Description:    This script is called when a remote node               #
#                       leaves the cluster.                                    #
#                       The script first checks to see if a                    #
#                       configured standby address exists and                  #
#                       is considered ’up’ by clstrmgr, then does              #
#                       a standby_address -> takeover_address swap.            #
#                                                                              #
```

```
#                                                                              #
#                       For an SP-switch, the script aliases the               #
#                       takeover address on the same adapter as the            #
#                       local service address.                                 #
#       Called by:      node_down_remote, node_up_local                        #
#                                                                              #
#       Calls to:       cl_swap_IP_address                                     #
#                       cl_alias_IP_address                                    #
#                                                                              #
#       Arguments:      takeover_address...                                    #
#                                                                              #
#       Returns:        0       success                                        #
#                       1       failure                                        #
#                       2       bad argument                                   #
#                                                                              #
################################################################################

PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster/events/utils:/us
r/sbin/cluster/utilities
export PATH

PROGNAME=$0
TELINIT=false
DELAY=5
STATUS=0
AliasPGM_Path=/usr/local/cluster/events/utils

if [ ! -n "$EMULATE" ]
then
        EMULATE="REAL"
fi



if [ $# -eq 0 ]
then
    cl_echo  1029 "Usage: $PROGNAME takeover_address...\n"  $PROGNAME
    exit 2
fi

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

# Routine to turn NIS on.
turn_on_DNS_NIS() {
if [ "$NAME_SERVER" = "true" ]
then
    if [ "$EMULATE" = "EMUL" ]
    then
        cl_echo 3020 "NOTICE >>>> The following command was not executed <<<<\n"
        echo "/usr/sbin/cluster/events/utils/cl_nm_nis_on\n"
    else
        /usr/sbin/cluster/events/utils/cl_nm_nis_on
    fi
    if [ $? -ne 0 ]
    then
        STATUS=$?
    fi
fi
}
```

```
#############################################################################
# Name: addback_route
#
#       When two or more standbys are on the same subnet, only one of the
#       standbys is in the routing table as the route. If this standby is
#       used to takeover the remote address, the route also gets destroyed
#       in routing table. This routine is used to restore the route for
#       the remaining standbys on the subnet.
#
# Arguments: standby_IP_address
#
# Returns: None
#
#############################################################################
addback_route () {

NETWORK=‘/usr/sbin/cluster/utilities/cllsif -cSn $1 | cut -d’:’ -f3 | uniq‘

standby_list=‘/usr/sbin/cluster/utilities/cllsif -cS | grep "standby" | cut -d’:’ -f7‘

for standby in $standby_list
do
    #
    # Make sure the standby is not the same one
    #
    if [ "$standby" = "$1" ]
    then
        continue
    fi

    #
    # Make sure two standbys are on the same network
    #
    network=‘/usr/sbin/cluster/utilities/cllsif -cSn $standby | cut -d’:’ -f3 | uniq‘
    if [ "$network" != "$NETWORK" ]
    then
        continue
    fi

    #
    # Make sure the standby is defined on local node
    #
    /usr/sbin/cluster/utilities/clgetif -n $standby >/dev/null 2>&1
    if [ $? != 0 ]
    then
        continue
    fi

    NETMASK=‘/usr/sbin/cluster/utilities/clgetif -n $standby‘
    INTERFACE=‘/usr/sbin/cluster/utilities/clgetif -a $standby‘

    #
    # Make sure the standby is up on local node
    #
    addr=i"$standby"_"$LOCALNODENAME"
    addr=‘/bin/echo $addr | /bin/sed -e s/[./]/x/g‘
    VAR=\$"$addr"
    set +u
    VAL="‘eval echo $VAR‘"
    set -u

    if [ "$VAL" != "UP" ]
    then
```

```
        continue
    fi

    #
    # Do ifconfig to add the route in. Will be a no-op if already in
    #
    if [ "$EMULATE" = "EMUL" ]
    then
        cl_echo 3020 "NOTICE >>>> The following command was not executed <<<< \n"
        echo "ifconfig $INTERFACE $standby netmask $NETMASK up \n"
    else
        ifconfig $INTERFACE $standby netmask $NETMASK up
    fi

done
}

#
# This routine maps a label_address to the internet_dot_address.
# Thus, given a label_address, we do not require the name server
# to get its corresponding dot address.
#
name_to_addr () {
    /bin/echo `/usr/sbin/cluster/utilities/cllsif -cSn $1 | cut -d: -f7 | uniq`
    exit $?
}

#####################################################################
# Start of modified part 1 (aliasing on other networks)
#####################################################################
use_aliasing ()
{
              save="placeholderjunk"
              SERVS=`/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
                | grep :service: | grep :$NETWORK: | /bin/cut -d':' -f1`
              for service in $SERVS
                do
                   # Check if netmon thinks service/boot is up or down
                   service_dot_addr=`name_to_addr $service`
                   addr=i"$service_dot_addr"_"$LOCALNODENAME"
                   addr=`/bin/echo $addr | /bin/sed -e s/[./]/x/g`
                   VAR=\$"$addr"
                   set +u
                   SERVICE_STATE="`eval echo $VAR`"
                   set -u
                  boot=`/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME | grep
:boot: | grep :$NETWORK: | /bin/cut -d':' -f1`
                  boot_dot_addr=`name_to_addr $boot`
                   addr=i"$boot_dot_addr"_"$LOCALNODENAME"
                   addr=`/bin/echo $addr | /bin/sed -e s/[./]/x/g`
                   VAR=\$"$addr"
                   set +u
                   BOOT_STATE="`eval echo $VAR`"
                   set -u

                   if [ "$SERVICE_STATE" = "UP" -o "$BOOT_STATE" = "UP" ]
                   then
                     INTERFACE=`/usr/sbin/cluster/utilities/clgetif -a $service_dot_addr`
                      NETMASK=`/usr/sbin/cluster/utilities/clgetif -n $service_dot_addr`
                      if [ "$EMULATE" = "EMUL" ]
                      then
                        cl_echo 3020 "NOTICE >>>> The following command was not executed
<<<<\n"
```

```
                    echo "$AliasPGM_Path/cl_alias_IP_address $INTERFACE $addr_dot_addr
$NETMASK\n"
                    else
                       $AliasPGM_Path/cl_alias_IP_address $INTERFACE $addr_dot_addr
$NETMASK
                    fi
                    STATUS=$?
                  fi
                if [ $STATUS -eq 0 ]
                then
                   break
                fi
              done # for service in $SERVS
}
#####################################################################
# End of modified part 1 (aliasing on other networks)
#####################################################################

#############################################################################
#
#  main routine
#
#############################################################################
# Turn NIS off.
if [ "$NAME_SERVER" = "true" ]
then
    if [ "$EMULATE" = "EMUL" ]
    then
        cl_echo 3020 "NOTICE >>>> The following command was not executed <<<< \n"
        echo "/usr/sbin/cluster/events/utils/cl_nm_nis_off\n"
    else
        /usr/sbin/cluster/events/utils/cl_nm_nis_off
    fi
    if [ $? -ne 0 ]
    then
        exit 1
    fi
fi

set -u

BOOT_ADDR=""
SERVICE_ADDR=""

for addr in $*
do

    #
    # Determine if address is already configured.  If not, try to
    # acquire it.
    #
    clgetif -a $addr 2>/dev/null
    if [ $? -ne 0 ]
    then

        #
        # Get dot address of takeover_address, network and configured standby
        # addresses for later use.
        #
        STATUS=1
        addr_dot_addr=‘name_to_addr $addr‘
        NETWORK=‘/usr/sbin/cluster/utilities/cllsif -cSn $addr_dot_addr | /bin/cut -d’:’
-f3 | uniq‘
```

```
        # Get the service address associated with this network
          SERVICE_ADDR=‘/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
                       | grep $NETWORK | grep service | /bin/cut -d’:’ -f7 | uniq‘

        # Determine he interface associated with the service address
        INTERFACE=‘/usr/sbin/cluster/utilities/clgetif -a $SERVICE_ADDR‘

        if [ -z "$INTERFACE" ]
        then
            # Get the boot address associated with this network
            BOOT_ADDR=‘/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
                        | grep $NETWORK | grep boot | /bin/cut -d’:’ -f7 | uniq‘

            # Determine the interface associated with the boot address
            INTERFACE=‘/usr/sbin/cluster/utilities/clgetif -a $BOOT_ADDR‘
        fi

        # Unable to determine local boot/service interface.  We should never be here
        if [ -z "$INTERFACE" ]
        then
            MSG=‘dspmsg scripts.cat 342 "Unable to determine local boot/service
interface.\n"‘
            STATUS=1
        fi

        # Determine if this is an SP switch interface.  If so,
        # execute appropriate script.
        if [ $INTERFACE = "css0" ]
        then

            # Determine the netmask
            for interface in $SERVICE_ADDR $BOOT_ADDR
            do
                SP_SWITCH_NETMASK=‘/usr/sbin/cluster/utilities/clgetif -n $interface‘
                if [ -n "$SP_SWITCH_NETMASK" ]
                then
                    break
                fi
            done

            if [ -n "$SP_SWITCH_NETMASK" ]
            then
                if [ "$EMULATE" = "EMUL" ]
                then
                       cl_echo 3020 "NOTICE >>>> The following command was not executed
<<<< \n"
                        cl_echo "/usr/sbin/cluster/events/utils/cl_swap_IP_address
cascading acquire $INTERFACE $addr $addr $SP_SWITCH_NETMASK"
                else
                       /usr/sbin/cluster/events/utils/cl_swap_IP_address \
                               cascading acquire $INTERFACE $addr  \
                               $addr $SP_SWITCH_NETMASK
                fi
                STATUS=$?
            else
                STATUS=1
            fi

            save="placeholderjunk"

        else
```

```
            STDBYS=‘/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME | grep
:standby: \
            | cut -d’:’ -f1,3 | grep -w $NETWORK | /bin/cut -d ’:’ -f1 ‘

#####################################################################
# Start of modified part 2 (aliasing on other networks)
#####################################################################
          print "STDBYS=’$STDBYS’"
          if [ -z "$STDBYS" ]
          then
              # run aliasing
              use_aliasing
            else # [ -n $STDBYS ]
                StbyAvailable=false
                # as normal
#####################################################################
# End of modified part 2 (aliasing on other networks)
#####################################################################

            for standby in $STDBYS
            do
                #
                # Get dot address of standby_label and its associated interface
                # for later use.
                #
                standby_dot_addr=‘name_to_addr $standby‘
                INTERFACE=‘/usr/sbin/cluster/utilities/clgetif -a $standby_dot_addr‘

                if [ -n "$INTERFACE" ]
                then
#####################################################################
# modified part 3
                    StbyAvailable=true
#####################################################################
                    #
                    # If standby address in the local node is ’up’,
                    # swap the standby_address to the takeover_address
                    # (cl_swap_IP_address).
                    #
                    NETMASK=‘/usr/sbin/cluster/utilities/clgetif -n $standby_dot_addr‘
                    save="i$standby_dot_addr"_"$LOCALNODENAME"
                    save=‘/bin/echo $save | /bin/sed -e s/[./]/x/g‘
                    VAR=\$"$save"
                    set +u
                    VAL="‘eval echo $VAR‘"
                    set -u
                    if [ -z "$VAL" -o "$VAL" = "UP" ]
                    then
                        if [ "$EMULATE" = "EMUL" ]
                        then
                                cl_echo 3020 "NOTICE >>>> The following command was not
executed <<<<  \n"
                                echo "/usr/sbin/cluster/events/utils/cl_swap_IP_address
cascading acquire $INTERFACE $addr_dot_addr $standby_dot_addr $NETMASK\n"
                                STATUS=$?
                                echo "addback_route $standby_dot_addr\n"
                                break
                        else
                                /usr/sbin/cluster/events/utils/cl_swap_IP_address \
                                cascading acquire $INTERFACE $addr_dot_addr
$standby_dot_addr $NETMASK
                                STATUS=$?
                                addback_route $standby_dot_addr
```

```
                                break
                            fi
                        fi
                    fi
                done
#####################################################################
# Start of modified part 4 (aliasing on other networks)
#####################################################################

                if [ "$StbyAvailable" = "false" ]
                then
                    use_aliasing
                fi
# fi from modified part 2
              fi # if [ -n "$STDBYS" ]
#####################################################################
# End of modified part 4 (aliasing on other networks)
#####################################################################
            fi

            if [ $STATUS -ne 0 ]
            then
               MSG=‘dspmsg scripts.cat 340 "IP Address Takeover of $addr_dot_addr failed.\n"
$addr_dot_addr‘
                /bin/echo $MSG >/dev/console

                # Turn Name Service back on
                turn_on_DNS_NIS
                exit 1
            else
                #
                # Mark this standby adapter ’DOWN’, so it will not be
                # used again in the next iteration
                #
                export $save=DOWN
                TELINIT=true
            fi
        fi
    done

    # Turn on Name Service
    turn_on_DNS_NIS

    #
    # Start tcp/ip servers and network daemons via ’telinit a’.
    #
    if [ "$TELINIT" = "true" ]
    then
        #
        # Set hostname to first public service address
        #
    #       FIRST_SERVS=‘/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
    #                   | grep :service: \
    #                   | grep :public: | cut -d’:’ -f1‘
    #       FIRST_SERV=‘echo $FIRST_SERVS |cut -d’ ’ -f1‘
    #       if [ -n "$FIRST_SERV" ]
    #       then
    #               hostname $FIRST_SERV
    #       fi


        if [ ! -f /usr/sbin/cluster/.telinit ]
        then
```

```
        #
        # In /etc/inittab, there is an entry to touch /usr/sbin/cluster/.telinit
        # after tcp/ip is functionally up.
        #
        #
        #  Save NFS exports list from /etc/xtab.  telinit will cause rc.nfs to
        #  be run which blows away the entries
        #
        ##
        if [ "$EMULATE" = "EMUL" ]
        then
                cl_echo 3020 "NOTICE >>>> The following command was not executed <<<< \n"
        echo " cp /etc/xtab /tmp/xtab"
        else

        cp /etc/xtab /tmp/xtab
        if [ $? -ne 0 ]
        then
            cl_echo 1051 "Could not save xtab file. Please export hacmp defined
filesystems\n"
        fi

        telinit a

        while [ ! -f /usr/sbin/cluster/.telinit ]
        do
            sleep $DELAY
        done

        cp /tmp/xtab /etc/xtab
        if [ $? -ne 0 ]
        then
            cl_echo 1052 "Could not restore xtab file. Please export hacmp defined
filesystems\n"
        fi
    fi # if emulate

    fi
fi

exit $STATUS
```

B.3.2 The HACMP Script release_takeover_addr

```
#!/bin/sh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 43haes430 src/43haes/usr/sbin/cluster/events/release_takeover_addr.sh 1.4.1.5
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1998
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# @(#)91      1.4.1.6  src/43haes/usr/sbin/cluster/events/release_takeover_addr.sh,
hacmp.events, 43haes430, 9816A_43ha430 4/24/98 08:56:07
#
```

```
#   COMPONENT_NAME: EVENTS
#
#   FUNCTIONS: none
#
#   ORIGINS: 27
#
#
#   (C) COPYRIGHT International Business Machines Corp. 1990,1994
#   All Rights Reserved
#   Licensed Materials - Property of IBM
#   US Government Users Restricted Rights - Use, duplication or
#   disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#############################################################################
#                                                                           #
#       Name:           release_takeover_addr                               #
#                                                                           #
#       Description:    This script is called if the local node has         #
#                       the remote node's service address on its            #
#                       standby adapter, and either the remote node         #
#                       re-joins the cluster or the local node              #
#                       leaves the cluster gracefully.                      #
#                                                                           #
#       Called by:      node_down_local, node_up_remote                     #
#                                                                           #
#       Calls to:       cl_swap_IP_address                                  #
#                                                                           #
#       Arguments:      takeover-address...                                 #
#                                                                           #
#       Returns:        0       success                                     #
#                       1       failure                                     #
#                       2       bad argument                                #
#                                                                           #
#############################################################################

PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster/events/utils:/us
r/sbin/cluster/utilities
export PATH

PROGNAME=$0
STATUS=0
AliasPGM_Path=/usr/local/cluster/events/utils

if [ ! -n "$EMULATE" ]
then
        EMULATE="REAL"
fi


if [ $# -eq 0 ]
then
        cl_echo 1029 "Usage: $PROGNAME takeover-address..\n" $PROGNAME
        exit 2
fi

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

# Routine to turn NIS on.
turn_on_DNS_NIS () {
if [ "$NAME_SERVER" = "true" ]
```

```
then
        if [ "$EMULATE" = "EMUL" ]
        then
                cl_echo 3020 "NOTICE >>>> The following command was not executed <<<< \n"
                echo "/usr/sbin/cluster/events/utils/cl_nm_nis_on \n"
        else
                /usr/sbin/cluster/events/utils/cl_nm_nis_on
        fi
        if [ $? -ne 0 ]
        then
                STATUS=$?
        fi
fi
}

#
# This routine maps a label_address to the internet_dot_address.
# Thus, given a label_address, we do not require the name server
# to get its corresponding dot address.
#
name_to_addr () {
        /bin/echo '/usr/sbin/cluster/utilities/cllsif -cSn $1 | cut -d: -f7 | uniq'
        exit $?
}

###############################################################################
#
#  main routine
#
###############################################################################

# Turn NIS off.
if [ "$NAME_SERVER" = "true" ]
then
        if [ "$EMULATE" = "EMUL" ]
        then
                cl_echo 3020 "NOTICE >>>> The following command was not executed <<<< \n"
                echo "/usr/sbin/cluster/events/utils/cl_nm_nis_off\n"
        else
                /usr/sbin/cluster/events/utils/cl_nm_nis_off
        fi

        if [ $? -ne 0 ]
        then
                exit 1
        fi
fi

set -u

for addr in $*
do
    #
    # Determine if address is already unconfigured.  If not, try to
    # release it. If hostname or interface not found, clgetif will
    # print error. If hostname not found fail event.
    #
    clgetif -a $addr
    return_code=$?
    if [ $return_code -ne 0 ]
    then
        # Only fail the event if hostname not found (return_code = 1)
        # If interface not found then already unconfigured so drop through
```

```
        if [ $return_code -eq 1 ]
        then
            exit 1
        fi
    else
        STBY_IP_ADDR=""
        addr_dot_addr=‘name_to_addr $addr‘

        #
        # Get the standby interface to which the remote service address is mapped.
        #
        STBY_INTERFACE=‘/usr/sbin/cluster/utilities/clgetif -a $addr_dot_addr‘

        if [ "$STBY_INTERFACE" = "" ]
        then
            cl_echo 318 "No service address $addr was taken by this node." $addr
            continue
        fi

        #
        # Get the netmask and network name for later use.
        #
        NETMASK=‘/usr/sbin/cluster/utilities/clgetif -n $addr_dot_addr‘
       NETWORK=‘/usr/sbin/cluster/utilities/cllsif -cSn $addr_dot_addr | cut -d’:’ -f3 |
uniq‘

        #
        # Get this node’s original standby address from the configuration.
        #
        STBYS=‘/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME|grep :standby: \
            | cut -d’:’ -f3,7 | grep -w $NETWORK | cut -d’:’ -f2‘

#####################################################################
# Start of modified part 1 (aliasing on other networks)
#####################################################################
        Local_IP_Addr=$(/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME|grep
:service: | grep -w $NETWORK | cut -d: -f7)
        Local_SVC_Interface=$(/usr/sbin/cluster/utilities/clgetif -a $Local_IP_Addr)
        if [ "$Local_SVC_Interface" = "$STBY_INTERFACE" ]
        then
           STBY_IP_ADDR=""
        else
        for s in $STBYS
        do
            if [ "‘/usr/sbin/cluster/utilities/clgetif -a $s‘" = "" ]
            then
                #
                # This standby is not configured, it is the missing standby.
                # Record it and exit loop.
                #
                STBY_IP_ADDR="$s"
                break
            fi
        done
        fi
#####################################################################
# End of modified part 1 (aliasing on other networks)
#####################################################################

        if [ -n "$STBY_IP_ADDR" ]
        then
            #
            # Reconfigure the standby.
```

```
            #

            if [ "$EMULATE" = "EMUL" ]
            then
               cl_echo 3020 "NOTICE >>>> The following command was not executed <<<<  \n"
                echo "/usr/sbin/cluster/events/utils/cl_swap_IP_address \
                        cascading release $STBY_INTERFACE $STBY_IP_ADDR $addr_dot_addr
$NETMASK \n"
            else
                /usr/sbin/cluster/events/utils/cl_swap_IP_address \
                      cascading release $STBY_INTERFACE $STBY_IP_ADDR $addr_dot_addr
$NETMASK
            fi
            if [ $? != 0 ]
            then
                STATUS=1
            fi
        else
            # Determine if the interface belongs to an SP switch.  If so,
            # call the SP Switch-related script.
            if [ -n "$STBY_INTERFACE" -a $STBY_INTERFACE = "css0" ]
            then
                if [ "$EMULATE" = "EMUL" ]
                then
                       cl_echo 3020 "NOTICE >>>> The following command was not executed
<<<< \n"
                        echo "/usr/sbin/cluster/events/utils/cl_swap_IP_address \
                        cascading release css0 $addr_dot_addr $addr $NETMASK \n"
                else
                        /usr/sbin/cluster/events/utils/cl_swap_IP_address \
                        cascading release css0 $addr_dot_addr $addr $NETMASK
                fi

                if [ $? != 0 ]
                then
                    STATUS=1
                fi
            else
#####################################################################
# Start of modified part 2 (aliasing on other networks)
#####################################################################

              if [ "$EMULATE" = "EMUL" ]
              then
                 cl_echo 3020 "NOTICE >>>> The following command was not executed <<<<\n"
                 echo "$AliasPGM_Path/cl_unalias_IP_address $STBY_INTERFACE $addr
$NETMASK\n"
              else
                 $AliasPGM_Path/cl_unalias_IP_address $STBY_INTERFACE $addr $NETMASK
              fi
              if [ $? != 0 ]
              then
                 cl_log 319 "No missing standby found for service address $addr." $addr
                 STATUS=1
              fi
#####################################################################
# End of modified part 2 (aliasing on other networks)
#####################################################################
            fi
        fi
    fi
done
```

```
turn_on_DNS_NIS
exit $STATUS
```

B.3.3 The Script cl_alias_IP_address

```
#!/bin/sh
PROGNAME="$0"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

DELETE_ROUTES=/usr/sbin/cluster/.delete_routes
ADD_ROUTES=/usr/sbin/cluster/.add_routes
ROUTE_ADD=0

#############################################################################
# Name: flush_arp
#
#Flushes entire arp cache
#
# Returns: None.
#############################################################################
flush_arp () {
for addr in '/etc/arp -a | /bin/sed -e 's/^.*(\([0-9].*[0-9]\)).*$/\1/' -e /incomplete/d'
do
/etc/arp -d $addr >/dev/null 2>&1
done
return 0
}

#############################################################################
# Name: add_routes
#
#       Echos route add commands nessary to restore routing table after
#adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#############################################################################
add_routes() {
/bin/echo "#!/bin/sh -x"
/bin/echo "PATH=$PATH"

for interface in "$@"
do
netstat -rn | fgrep $interface | fgrep " UG " | \
awk '{print "route add -net " $1" "$2}'

netstat -rn | fgrep $interface | fgrep -v " UG " | \
                        fgrep -v " U " | awk '{print "route add " $1" "$2}'
done

        /bin/echo "exit 0"
return 0
}

#############################################################################
# Name: delete_routes
#
#       Echos route delete commands nessary to clear routing table
```

```
#       before adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
###############################################################################
delete_routes () {
        /bin/echo "#!/bin/sh -x"
/bin/echo "PATH=$PATH"

for interface in "$@"
do
        netstat -rn | fgrep $interface | fgrep -v "H" | \
awk '{print "route delete -net " $1" "$2}'
done

        for interface in "$@"
        do
                netstat -rn | fgrep $interface | fgrep "H" | \
                        awk '{print "route delete " $1" "$2}'
        done

        /bin/echo "exit 0"
return 0
}

###############################################################################
#
# Main entry point
#
###############################################################################
cl_echo 33 "Starting execution of $0 with parameters $*" $0 "$*"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

set -u

# this form for single interface
if [ $# -eq 3 ]
then
IF=$1
ADDR=$2
NETMASK=$3

        # Get routes bound to adapter and create script file
        # to re-add routes later.
add_routes $IF | tee $ADD_ROUTES
#add_routes $IF > $ADD_ROUTES
chmod +x $ADD_ROUTES

# Prevent 'no routes to dest' errors by adding default
# route to loopback. The packets will get dropped but
# TCP will endure
route add default 127.0.0.1 >/dev/null 2>&1
ROUTE_ADD=$?

# down old interfaces
cl_echo 60 "$PROGNAME: Configuring adapter $IF at IP address $ADDR" $PROGNAME $IF $ADDR
#ifconfig $IF down
```

```
# Must delete routes because of ifconfig down.
        delete_routes $IF | tee $DELETE_ROUTES
       #delete_routes $IF > $DELETE_ROUTES
        chmod +x $DELETE_ROUTES
        $DELETE_ROUTES

#set the specified interface to specified address
ifconfig $IF alias $ADDR netmask $NETMASK up
if [ $? -ne 0 ]
       then
ifconfig $IF alias $ADDR netmask $NETMASK up
if [ $? -ne 0 ]
       then
              cl_log 59 "$PROGNAME: Failed ifconfig $IF inet $ADDR netmask $NETMASK up."
$PROGNAME $IF $ADDR $NETMASK
              exit 1
fi
fi

# flush arp table
flush_arp

# Add back pre-existing routes
$ADD_ROUTES

# Delete default route only if we succeed before
if [ $ROUTE_ADD -eq 0 ]
then
route delete default 127.0.0.1
fi

else
# else bad arg count
cl_echo 62 "usage: $PROGNAME interface address netmask" $PROGNAME
cl_echo 63 "    or $PROGNAME interface1 address1 interface2 address2 netmask" $PROGNAME
exit 2
fi

cl_echo 32 "Completed execution of $0 with parameters $*.  Exit status = $?" $0 "$*" $?

exit 0
```

B.3.4 The Script cl_unalias_IP_address

```
#!/bin/sh -x
#
# Returns:      0 - success
#               1 - ifconfig failure
#               2 - bad number of arguments
#               3 - Hardware swap failure
#
# Environment:  VERBOSE_LOGGING,PATH
################################################################################
NEW_ADDRESSES=
PATH=$PATH:/usr/sbin/cluster/events/utils

PROGNAME="$0"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi
```

```
DELETE_ROUTES=/usr/sbin/cluster/.delete_routes
ADD_ROUTES=/usr/sbin/cluster/.add_routes
ROUTE_ADD=0

#############################################################################
# Name: flush_arp
#
#Flushes entire arp cache
#
# Returns: None.
#############################################################################
flush_arp () {
for addr in '/etc/arp -a | /bin/sed -e 's/^.*(\([0-9].*[0-9]\)).*$/\1/' -e /incomplete/d'
do
/etc/arp -d $addr >/dev/null 2>&1
done
return 0
}

#############################################################################
# Name: add_routes
#
#       Echos route add commands nessary to restore routing table after
#adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#############################################################################
add_routes() {
/bin/echo "#!/bin/sh -x"
/bin/echo "PATH=$PATH"

for interface in "$@"
do
netstat -rn | fgrep $interface | fgrep " UG " | \
awk '{print "route add -net " $1" "$2}'

netstat -rn | fgrep $interface | fgrep -v " UG " | \
                        fgrep -v " U " | awk '{print "route add " $1" "$2}'
done

        /bin/echo "exit 0"
return 0
}

#############################################################################
# Name: delete_routes
#
#       Echos route delete commands nessary to clear routing table
#       before adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#############################################################################
delete_routes () {
        /bin/echo "#!/bin/sh -x"
/bin/echo "PATH=$PATH"

for interface in "$@"
do
```

```
        netstat -rn | fgrep $interface | fgrep -v "H" | \
awk '{print "route delete -net " $1" "$2}'
done

        for interface in "$@"
        do
                netstat -rn | fgrep $interface | fgrep "H" | \
                        awk '{print "route delete " $1" "$2}'
        done

        /bin/echo "exit 0"
return 0
}
#
# This routine maps a label_address to the internet_dot_address.
# Thus, given a label_address, we do not require the name server
# to get its corresponding dot address.
#
name_to_addr () {
    /bin/echo `/usr/sbin/cluster/utilities/cllsif -cSn $1 | cut -d: -f7 | uniq`
    exit $?
}
################################################################################
#
# Main entry point
#
################################################################################
cl_echo 33 "Starting execution of $0 with parameters $*" $0 "$*"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

set -u

# this form for single interface
if [ $# -eq 3 ]
then
IF=$1
NAME=$2
NETMASK=$3

ADDR=`name_to_addr $NAME`

        # Get routes bound to adapter and create script file
        # to re-add routes later.
add_routes $IF > $ADD_ROUTES
chmod +x $ADD_ROUTES

# Prevent 'no routes to dest' errors by adding default
# route to loopback. The packets will get dropped but
# TCP will endure
route add default 127.0.0.1 >/dev/null 2>&1
ROUTE_ADD=$?

# down old interfaces
cl_echo 60 "$PROGNAME: Configuring adapter $IF at IP address $ADDR" $PROGNAME $IF $ADDR
#ifconfig $IF down

# Must delete routes because of ifconfig down.
        delete_routes $IF > $DELETE_ROUTES
        chmod +x $DELETE_ROUTES
```

```
        $DELETE_ROUTES


ifconfig $IF $ADDR delete

# flush arp table
flush_arp

# Add back pre-existing routes
$ADD_ROUTES

# Replace automated route - but use alias in case > 1 addresses
ADDRESS=‘ifconfig $IF | awk ’FNR==2 {print $2}’‘
NETMASK=‘/usr/sbin/cluster/utilities/clgetif -n $ADDRESS‘

ifconfig $IF alias $ADDRESS netmask $NETMASK up

# Delete default route only if we succeed before
if [ $ROUTE_ADD -eq 0 ]
then
route delete default 127.0.0.1
fi

else
# else bad arg count
cl_echo 62 "usage: $PROGNAME interface address netmask" $PROGNAME
exit 2
fi

cl_echo 32 "Completed execution of $0 with parameters $*.  Exit status = $?" $0 "$*" $?

exit 0
```

Appendix C. Special Notices

This publication is intended to help HACMP Enhanced Scalability and RS/6000 SP specialists who want to know about RS/6000 Cluster Technology. The information in this publication is not intended as the specification of any programming interfaces that are provided by HACMP Enhanced Scalability and RS/6000 Cluster Technology. See the PUBLICATIONS section of the IBM Programming Announcement for HACMP Enhanced Scalability and RS/6000 Cluster Technology for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate

them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	HACMP
IBM ®	RS/6000
SP	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 233.

- *RS/6000 SP High Availability Infrastructure,* SG24-4838
- *RS/6000 SP Monitoring: Keep It Alive*, SG24-4873
- *RS/6000 SP: PSSP 2.2 Survival Guide*, SG24-4928
- *HACMP Enhanced Scalability*, SG24-2081
- *HACMP Enhanced Scalability User-Defined Events*, SG24-5327

D.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

D.3 Other Publications

These publications are also relevant as further information sources:

- *AIX Version 3.2 and 4 Performance Tuning Guide*, SC23-2365

- *HACMP for AIX, Version 4.3: Concepts and Facilities*, SC23-4276
- *HACMP for AIX, Version 4.3: Planning Guide*, SC23-4277
- *HACMP for AIX, Version 4.3: Installation Guide*, SC23-4278
- *HACMP for AIX, Version 4.3: Administration Guide*, SC23-4279
- *HACMP for AIX, Version 4.3: Troubleshooting Guide*, SC23-4280
- *HACMP for AIX, Version 4.3: Programming Locking Applications*, SC23-4281
- *HACMP for AIX, Version 4.3: Programming Client Applications*, SC23-4282
- *HACMP for AIX, Version 4.3: Master Index and Glossary*, SC23-4285
- *HACMP for AIX, Version 4.3: Enhanced Scalability Installation and Administration Guide*, SC23-4284
- *IBM RS/6000 Cluster Technology for AIX: Event Management Programming Guide and Reference*, SA22-7354
- *IBM RS/6000 Cluster Technology for AIX: Group Services Programming Guide and Reference*, SA22-7355
- *IBM Parallel System Support Programs for AIX Administration Guide*, SA22-7348

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** – to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** – send orders to:

	IBMMAIL	**Internet**
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 408 256 5422 (Outside USA)** – ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

ACK	Acknowledgment
ADSM	ADSTAR Distributed Storage Manager
AIX	Advanced Interactive Executive
API	Application Program Interface
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
C-SPOC	Cluster Single Point of Control facility
Clinfo	Client Information Program
clsmuxpd	Cluster Smux Peer Daemon
CNN	Cluster Node Number
CP	Crown Prince
CPU	Central Processing Unit
CWS	Control Workstation
DARE	Dynamic Automatic Reconfiguration Event
DMS	Deadman Switch
DNS	Domain Name Server
EM	Event Management
EMAPI	Event Management API
EMCDB	Event Management Configuration Database
FDDI	Fiber Distributed Data Interface
FSM	Finite State Machine
GID	Group Identifier
GL	Group Leader
GODM	Global Object Data Manager
GPFS	General Parallel File System
GS	Group Services
GSAPI	Group Services Application Programming Interface
HACMP	High Availability Cluster Multi-Processing
HACMP/ES	HACMP Enhanced Scalability
HAI	High Availability Infrastructure
HANFS	High Availability Network File System
HB	Heartbeat
HiPS	High Performance Switch
HWAT	Hardware Address Takeover
HWM	High Water Mark
IBM	International Business Machines Corporation
ICMP	Internet Control Message Protocol
IP	Interface Protocol
IPAT	IP Address Takeover
ITSO	International Technical Support Organization
JFS	Journaled File System
KA	Keep Alive
LAN	Local Area Network
LPP	Licensed Program Product
LVCB	Logical Volume Control Block
LVM	Logical Volume Manager

LWM	Low Water Mark
MAC	Medium Access Control
MIB	Management Information Base
MIT	Massachusetts Institute of Technology
NFS	Network File System
NIM	Network Interface Module
NIS	Network Information System
NS	Name Server
ODM	Object Data Manager
PBS	Phoenix Broadcast System
Perl	Practical extraction & reporting language
PID	Process Identifier
PING	Packet Internet Groper
PSSP	Parallel System Support Programs
PTF	Program Temporary FIX
PTPE	Performance Toolbox Parallel Extension
PTX/6000	Performance Toolbox/6000
RAID	Redundant Array of Independent Disks
RMAPI	Resource Monitor API
RSCT	RS/6000 Cluster Technology
SBS	Structured Byte String
SCSI	Small Computer System Interface
SDR	System Data Repository
SMIT	System Management Interface Tool
SMP	Symmetric Multi-Processors
SMUXD	SNMP Multiplexor Daemon
SNMP	Simple Network Management Protocol
SP	RS/6000 SP
SPMI	System Performance Measurement Interface
SPS	SP Switch
SRC	System Resource Controller
SSA	Serial Storage Architecture
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TMSCSI	Target Mode SCSI
TMSSA	Target Mode SSA
TS	Topology Services
UDP	User Datagram Protocol
UID	User Identifier
VG	Volume Group
VGDA	Volume Group Data Area
VSD	Virtual Shared Disks
VSM	Visual System Management

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

K

L

T

U

V

Z

ITSO Redbook Evaluation

HACMP Enhanced Scalability Handbook
SG24-5328-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ **Customer** _ **Business Partner** _ **Independent Software Vendor** _ **IBM employee**
_ **None of the above**

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

SG24-5328-00
Printed in the U.S.A.